

SMFUTIL
THE SMF DATA MOVER
VERSION 8

SYSTEM PROGRAMMERS GUIDE

©COPYRIGHT 1985-2006
WOVEN SOFTWARE
PINEHURST, TEXAS

OCTOBER 25, 2006

WS-SU8000-3

**MARKETED IN THE UNITED STATES AND CANADA BY
ADVANCED SOFTWARE PRODUCTS GROUP, INC.
3185 SOUTH HORSESHOE DRIVE
NAPLES, FLORIDA 34104
MARKETING AND CONTRACT INFORMATION
PHONE (239) 649-1548
FAX (239) 649-3291**

**COPYRIGHT 1985-2005
WOVEN SOFTWARE
3201 GOODSON LOOP
PINEHURST, TEXAS 77362**

**Permission is granted for current "Licensed Customers" of the SMFUTIL
product to make limited duplications of this manual for use within their
own facilities so long as such duplication and usage falls within the scope
and terms of the product license.**

All other rights reserved.

**TECHNICAL SUPPORT
PHONE (281) 252-9499
FAX (281) 252-9560**

SUMMARY OF AMENDMENTS - VERSION 6.0

The following items are new or revised over version 5.0 and 5.1 of SMFUTIL.

1. IDRC SUPPORT IN DYNAMIC ALLOCATION.

The DDA block structure has been enhanced to support IDRC compression via a specification in the DCB parameter keyword.

2. ADDITIONAL GDG PROCESSING SUPPORT.

Two new keywords, GDGCKPTI and CKPTGDG (DDA Block) now offer simplified processing of GDG data sets.

3. IMPROVED PERFORMANCE.

Enhanced I/O overlap and buffer management should result in reduced CPU and SRB usage and substantial reductions in clock execution time.

4. IMPROVED ALLOCATION ERROR REPORTING.

An interface to the Allocation Failure Message generator (IEFDB476 for MVS/XA and MVS/ESA and IKJEFF18 for MVS/370) has been implemented to assist the user in determining the cause of SVC 99 errors during testing. In addition, SMS error codes are now reported.

5. FULL SMS SUPPORT IN DYNAMIC ALLOCATION DDA BLOCKS.

The DDA command structure has been enhanced to support the specification of SMS parameters "LIKE", "MGMTCLAS", "DATACLAS", and "STORCLAS" keywords for allocation of data sets targeted for SMS control. SMS must be active for the keywords to be used.

6. OUTPUT DATA SET DATE/TIME RANGE LIMITING.

A new keyword, ODTRANGE, has been added to allow an output data set to be limited to a specific date/time range. This allows multiple data sets to be created in the same execution and yet have different date/time span characteristics.

7. AUTOMATED DUMPING AND CLEARING OF ALL SMF DATA SETS CONTAINING DATA IN A SINGLE STEP.

A new keyword, DUMPCLEARALL, has been added to allow fully automate the SMF DUMPING and CLEARING process. DUMPCLEARALL automatically determines which SMF data sets contain data, switches SMF data set if required, allocates the SMF data sets to be dumped in the correct order (oldest first), dumps the data and clears the SMF data sets just dumped. Data set "MOD" restart protection is fully supported and automatic recovery is built in.

8. ENHANCED COPYBOTH AND COPYVBTH CAPABILITY.

The keywords COPYBOTH and COPYVBTH have been enhanced to accept a parmlist of one or more records types to be copied to both the user defined target (via TYPETODD for example) and to the standard output data set(s) (i.e. SYSUT2). This allows some record types to be selectively retained on the master output file without having to keep all types when subsets are being created.

9. ENHANCED IEFU29 EXIT.

The IEFU29 exit distributed with the product has been enhanced. The exit can now check for the presence of JESx prior to executing a start command. Two versions of the exit are present. Member IEFU29S acts as always and IEFU29 issues a start command with no SMFDSN keyword present. This is for shops that wish to implement the DUMPCLEARALL method of dumping and clearing the SMF data sets all at one time.

10. ENHANCED ARCHIVE MAINTENANCE AND REPORTING UTILITIES.

The SMFUAMNT and SMFUARPT utility programs have been significantly enhanced. SMFUAMNT now supports deleting data sets at the names and/or volume level via a DELETEDSN and DELETEVOL keywords. In addition, entire sets may now be deleted or renamed. SMFUARPT includes new report selection and ordering capability with full filtering and masking on multiple fields.

11. NOTIFICATION FACILITY CAPABILITY.

A new keyword, NOTIFY, has been added to allow the notification of the system operator and/or selected TSO users of the completion condition of an SMFUTIL execution. Optionally, the notification can be restricted to notification only on run failure.

12. PROCESS IDENTIFICATION VIA CHECKPOINT RECORD.

Two new keywords, PROCESS and PROCSYS, have been added to allow an SMFUTIL process to be uniquely identified in the associated checkpoint data set. This helps to prevent the incorrect usage of a checkpoint data set in more than one process.

13. IMPROVED VERIFICATION OF ARCHIVED DATA SET STATUS.

During searches of the SMFUTIL archive, SMFUTIL now verifies that the data set from the archive record is still cataloged and that the volser indicated is still part of the catalog set. This catalog validation can be disabled by specifying NOLOCATE at execution time.

14. AUTOMATIC CATALOG VOLUME OVERRIDE.

During searches of the SMFUTIL archive, if the selected archived data set is on DASD, SMFUTIL checks the volume in the system catalog against the volume in the SMFUTIL archive database. If different, SMFUTIL overrides with the catalog volume and unit specification. This is to allow archived SMF data sets on disk volumes to be freely moved about via HSM or similar products and still be available for SMFUTIL archive retrieval.

15. IMPROVED CICSSPLIT CAPABILITY.

The CICSSPLIT keyword now allows the data selected to be limited by subtype, class, and CICS version. This allows very specific CICS data subsets to be created. In addition, full masking of the 'applid' is now supported.

16. ENHANCED VALUE SPLIT PROCESSING

During data value splitting process (such as CICSSPLIT), a new keyword (VSINGLE) allows the site to specify if a record is to be copied only to the first matching target or to multiple targets if more than one match is found. VSINGLE capability allows specific targets to be set up to receive required data and a more generic (masked) target to receive all other data with each record going to one and only one user specified DD.

17. IMPROVED RECORD PRINT FUNCTION.

The PRINT function has been enhanced to better recognize and report CICS record (type 110) data content. Multiple CICS versions in the same data stream are now supported.

18. MVS/ESA RELEASE 5 SUPPORT.

SMFUTIL 6.0 supports the IBM SYSPLEX system architecture including the extended SMF MAN data set naming convention.

19. DUMMY DDA DATA SET SUPPORT.

The DDASTART/DDAEND block structure syntax now supports a "DUMMY" parameter to allow a NULLFILE data set to be defined for dynamic allocation.

20. DETAILED OUTPUT STATISTICS NOW AVAILABLE

A new keyword, ODETAIL, has been added to allow the user to request detailed statistics of individual record types for the output data sets.

21. ARCHIVE RECORD EXPIRATION DATES NOW SUPPORTED

A new keyword in the DDA block structure, AEXPIRE, has been added to allow the user to specify the length of time the related archive record is to be retained in the archive.

22. NEW SUFFIX FORMAT SUPPORT

SMFUTIL now supports a WEEK format data set name suffix. The SUFFIX keyword of the DDA block structure has been enhanced to support this and a new keyword, GENDSNW, has been added to support adding the suffix to JCL DD's SYSUT2 and SYSUT2D just as GENDSND and GENDSNM do. In addition, the DDA block SUFFIX command also now supports a date and time format suffix for creating unique data sets each time a dump is run.

23. CACONVRT UTILITY ENHANCEMENTS

The CACONVRT utility program that converts CA/JARS-SMF SCDS databases to SMFUTIL format has been enhanced to automatically convert all configurations in a single pass or a selected configuration only. In addition, CACONVRT'ed records are now automatically flagged as non-cataloged data sets. This allows the user to omit the NOLOCATE keyword from SMFUTIL extract runs that may consider older CA/JARS-SMF created data sets for input. During the extract, these data sets can now be mixed with new SMFUTIL created data set without bypassing the safety of performing a LOCATE to ensure the SMFUTIL data sets are still available via the system catalog.

24. AUTOMATED CATALOGED DATA SET INPUT.

A new keyword, CATINPUT, offers simplified processing of an entire catalog index level of data sets for input. While designed to be normally used with date/time suffixed data sets, it also includes support for GDG indexes.

25. CA/JARS SCDS DATA ERROR OVERRIDE

A new keyword, INDTVOFF, enables CA/JARS-SMF conversions sites to be able to bypass time errors in the SCDS records and obtain all data from CA/JARS-SMF created data sets.

SUMMARY OF AMENDMENTS - VERSION 6.1

The following items are new or revised over version 6.0 of SMFUTIL.

1. ARCHIVE DATABASE MAINTENANCE SUPPORT VIA ISPF.

A new ISPF based ARCHIVE maintenance utility is included in the SMFUTIL base package. This facility allows examination and modification of SMFUTIL Archive data in real time.

2. TEXTSCAN CAPABILITY.

A new keyword, TEXTSCAN, has been added to allow the user to search entire record for specific text strings. Masking is supported.

3. EXPANDED KEYWORD PARSER.

The syntactical processing of SMFUTIL control keywords has been totally rewritten and now supports keywords longer than 8 characters. Several previous keywords that were truncated to fit the 8 character limit now support the longer, more natural version of the word. For example, the keyword "YESTRDAY" may now be specified as the more natural "YESTERDAY". The previous, truncated version of the keyword affected will continue to be supported.

4. INPUT I/O ERROR LIMIT CAPABILITY.

A new keyword, IOELIMIT, has been added to allow the user or site to specify the number of consecutive I/O error an input device may generate before the execution is terminated. This avoids flooding the operator console with IOS001 messages in the event that an input device is generating constant errors with no progress being made in processing the media.

5. YEARFIRST OPTION ON GENERATED DATA SET NAME SUFFIXES.

A new keyword, YEARFIRST, allows the year to be at the front of the generated suffix for that catalog listings will appear in chronological order. The MONTH suffix syntax will be '.Yccyymm' and the week suffix syntax will be '.YccyyW##'. Note that the DAY and TIME suffix types already have the year first structure so are unaffected by YEARFIRST. To affect only MONTH suffix types, the keyword YEARFIRSTM may be specified. To affect only WEEK suffix types, the keyword YEARFIRSTW may be specified.

6. IMPROVED DATESCAN FUNCTION.

DATESCAN has been reworked to report boundaries of the missing data section in a more friendly fashion. Each boundary is now listed using the day of the week and the Gregorian date and time of the start and end of the section. In addition, an optional keyword of "WEEKDAYS" not allows the DATESCAN function to be limited to checking Monday through Friday for missing data.

7. SUPPORT FOR SUBTYPE SPECIFICATIONS ON INCLUDE/EXCLUDE

The INCLUDE and EXCLUDE keywords now support record subtype specifications for specific record types.

8. NEW USER PERPETUAL CALENDAR SUPPORT

In addition to the previous MONTHCAL user defined calendar, SMFUTIL now supports two new perpetual calendar structures, WEEKPCAL and MONTHPCAL. These structures break a calendar year into slices and assign a suffix value to each slice. The WEEKPCAL is used when week split and/or week suffixing is requested and the MONTHPCAL is used when month split and/or month suffixing is requested.

9. INTERNAL CONTROL OF GDG DATA SETS

The user may now optionally allow SMFUTIL to control the allocation of GDG data set name data sets such that the base GDG name is not ENQ'ed upon. This prevents SMFUTIL dump jobs that add data to one level of the GDG from interfering with other SMFUTIL jobs retrieving data from a different relative level of the same GDG base.

10. AUTOMATIC BACKUP OF ARCHIVE DATABASE AFTER UPDATES APPLIED.

The user may now optionally allow SMFUTIL to automatically back up the ARCHIVE base cluster after updates have been applied. This ensures that critical archive data is protected. This backup can be limited to once per week, once per day or any time an update has been applied to the database.

11. NATURAL ABBREVIATIONS SUPPORTED

Many of the keywords, especially for dynamic allocation, now have abbreviations that will help reduce syntax errors. For example, DDNAME may now be specified as DDN and DATA SET NAME as DSN.

12. ARCHIVE UPDATE PROTECTION VIA SMF RECORDS

SMFUTIL now has the optional ability to protect the update process of the SMFUTIL SMF data ARCHIVE. When an output data set that is being archived is closed, a before update copy of it's archive record is written to SMF via an installation specified SMF record type. When the update is successfully completed, an after image record is also written to indicate the update was completed.

13. ARCHIVE DATA FORWARD RECOVERY CAPABILITY

If the archive update protection facility is enabled, SMFUTIL now has the ability to forward recover ARCHIVE data via a special purpose execution containing the ARCHIVERECOVER keyword. This enables forward recovery of an archive database base cluster. This can be used to bring the database up to date after a restore of a damaged DASD volume that contains the base cluster.

14. EXECUTION ACCESS CONTROL

Easier control of who has authorization to execute SMFUTIL can be obtained by coding a macro called AUTHUSER in the SMFUDFLT module. As distributed, the AUTHUSER macro is set to a mask of '-' which allows any USERID to execute SMFUTIL. Masking in the defined Userid's is supported.

15. CRITICAL KEYWORD ACCESS CONTROL

Access to certain SMFUTIL keywords can be restricted via two macro structures in the SMFUDFLT module. AUTHWORD defines a critical SMFUTIL keyword (such as CLEAR or DUMPCLEARALL), and WORDUSER defines a list of one or more Userid's that can access the keyword. Masking in the defined Userid's is supported.

16. MULTIPLE DATABASE SUPPORT

For installations that have more than one ARCHIVE database, SMFUTIL now supports an improved RESERVE/DEQ protocol that protects updates using the cluster name as the RNAME component of the RESERVE/DEQ macro parameter list. This allows multiple database to be updated simultaneously. The old reserve methodology is still supported for compatibility to prior releases during the implementation of this release. The RSRVOLD parameter of SMFUDFLT should be left as the default of "YES" until this release has been moved to production usage on all systems and all prior versions of SMFUTIL are removed.

17. IMPROVED INTERRUPTED RUN RESTART CAPABILITY

A process restart after an interrupted run (with a SYSCKPT data set present) has been improved to automatically scratch and/or un-catalog any new data set created in the previous, interrupted, run, prior to the run restart commencing. This prevents dynamic allocation failures, duplicate data sets on multiple VTOC's, and/or improper catalog entries from being created.

18. IMPROVED CLEAR PROCESSING

A new function keyword, **SHORTCLEAR**, instructs **SMFUTIL** to utilize a “smart” clearing process when clear is invoked. This process uses an update in place method of writing the clearing records instead of a sequential overwrite. In addition, during **DUMPCLEARALL** execution, **SHORTCLEAR** only clears the actual existing data in the **MANx** data set, not the entire data set. This greatly speeds clear processing when clearing a **MANx** data set that is only partially full. It will also reduce the wall clock execution time for clearing full **MANx** data sets at the expense of increased CPU and EXCP usage. **SHORTCLEAR** defaults to **YES** in the **SMFUDFLT** module as shipped. It may be disabled at execution time via the **NOSHORTCLEAR** keyword.

19. SYMBOLIC DATE STRING OVERLAY SUPPORT IN DATA SET NAME FIELDS

Dynamically allocated output data set name fields may now contain specific symbolic date field strings that will be overlaid with the corresponding date information from the first record placed to the data set. This capability allows date structures to be placed anywhere in the data set name rather just at the end as suffixing does. In addition, multiple insertions per data set name are allowed.

SUMMARY OF AMENDMENTS - VERSION 7.0

The following items are new or revised over version 6.1 of SMFUTIL.

1. SUPPORT FOR HIGH-SPEED CARTRIDGE POSITIONING FOR DATA RETRIEVAL

SMFUTIL now supports high-speed tape positioning for data retrieval on 3480, 3490, and 3590 type devices. Automatic positioning to the nearest day is provided when data is retrieved via the SMFUTIL archive database. Manual positioning is supported via a new DDA block keyword, BIDSTART.

2. HIGH-SPEED CARTRIDGE POSITIONING FOR DATA SET EXTENTION

SMFUTIL now supports high-speed tape positioning for data set extension (MOD processing) on cartridge type devices. This support requires the use of the SYSCKPT data set to maintain the block position at last close on a MOD data set.

3. BAD SNAP OUTPUT LIMIT SUPPORTED

SMFUTIL now supports a limitation being placed on the number of records that are snapped to the BADSNAP DD statement. This allows the DD statement to be placed in large runs without the fear that large amounts of output snaps being produced.

4. BYTES MOVED STATISTICS REPORT ADDED

A new optional report showing the amount and percentages of bytes moved has been added. This report is requested via the BYTEDETAIL keyword or can be enabled by default via the "BDETAIL" parameter in the SMFUDFLT module during installation..

5. DUPLICATE RECORD DELETION EXIT POINT DEFINED

A new exit point can be used by the installation to gain control when a duplicate record is to be discarded. The DDEXIT keyword defines the user exit load module name to SMFUTIL. The exit may examine the record and decide if it is to be discarded or retained.

6. DISABLING ARCHIVE DATABASE USAGE DURING EXECUTION NOW SUPPORTED

A new keyword, NOARCHDB, now allows the user to disable the usage of an archive database defined in the SMFUDFLT defaults module during installation.

7. DISABLING SYSUT1 USAGE

A new keyword, NOSYSUT1, instructs SMFUTIL to ignore the specific DDNAME of SYSUT1 and only use input DD names that have at least 1 character appended to SYSUT1. This allows SMFUTIL to be executed in TSO sessions where SYSUT1 is allocated to a temporary work data set at logon.

8. YEAR 2000 COMPLIANCE

SMFUTIL is now fully Year 2000 compliant. As part of the required changes, all keyword parameters that are used to specify a year value now require that the year specification include the century value. Where the century value was optional it is now required as part of the syntax of the keyword parameter string.

9. IMPROVED CICSTYPE FUNCTIONALITY

The CICSTYPE selection filter has been enhanced to provide a report on the applid, CICS version and record count selected for the output data that matched the specified criteria.

10. CICSSPLIT ENHANCED

The VALUE SPLIT report produced by the CICSSPLIT keyword has been enhanced to show the CICS version that produced the selected data.

11. DCA CHECKPOINT DATA SET OVERRIDES

The DUMPCLEARALL Checkpoint data set allocation specifications may now be specified at execution time.

12. INPUT DETAIL STATISTICS REPORT ADDED

A new report, INPUT DETAIL, has been added. This report provides complete statistics, to the record type level, for each input data set. This report may be requested at execution time via the INPUTDETAIL keyword, or set to be produced all the time by setting the IDetail keyword in the SMFUDFLT module to "YES".

13. OUTPUT DETAIL STATISTICS REPORT MAY BE PRODUCED BY DEFAULT

The OUTPUTDETAIL report may now be produced by default by setting the ODETAIL parameter in the SMFUDFLT module to "YES".

14. SYSTEM DETERMINED BLOCKSIZE CAN BE ENABLED

A new keyword, SDB, instructs SMFUTIL that an omitted block size or a value of '0' is not to be overridden with internally chosen block size values. The operating system will supply an appropriate value.

15. IMPROVED INPUT DATA SET REPORTING

The INPUT DATA SET STATISTICS report has been enhanced to show the individual dsnames and volumes of all data sets concatenated to a single DDNAME.

16. ARCHIVE SELECTION REPORTING IMPLEMENTED

During an execution that selects input data sets via the SMFUTIL ARCHIVE database, each ARCHIVE record selected will be listed and the date/time range available will be shown along with the starting Block ID value, if available.

17. DYNAMIC LICENSE CODE IMPLEMENTED

The product license code may now be installed into the product as always or specified dynamically at execution time.

18. IMPROVED SORTED CONDITION CHECKING

The SORTTOLERANCE keyword has been altered to allow a "0" (zero) value to be specified to allow true sorted condition to be tested. In addition, a units value keyword has been added to allow the tolerance to be specified in minutes, seconds or hundredths of a second. The sorted/unsorted messages have been enhanced to show the sort tolerance that was in use. If an out of sort condition is detected, an additional message will be produced showing the record type of the first record discovered to be out of sort.

19. IMPROVED ODTRANGE CAPABILITY

The ODTRANGE keyword has been improved to allow multiple range specifications for the same DDNAME. This allows a single DDNAME to be used for multiple selected time ranges.

20. OUTPUT TIME RANGE CABILITY

A new keyword, OTIMERANGE, allows time ranges to be assigned to specific DD names to limit the data placed to them. This capability enables the user to create full and limited output data sets in a single pass of the input data.

21. DATA DISTRIBUTION REPORTS IMPLEMENTED

SMFUTIL now has the ability to produce two new reports called INPUTDATADISTRIBUTION (abbreviated as IDIST) and OUTPUTDATADISTRIBUTION (abbreviated as ODIST). These reports show the distribution of record types over the 24 hour day period by hour. Averaging can be specified by specific record type or all record types.

22. SUPPORT FOR IFASMFDP STYLE EXITS

SMFUTIL now supports three new keywords, USER1, USER2 and USER3. These are used to designate user supplied exit routine module names. These exits are called in accordance with the IBM defined IFASMFDP exit points of the same name. This allows SMFUTIL to directly drive locally and vendor developed SMF exits intended to be driven by IFASMFDP, such as the RACF 2.2 exits IRRADU00 and IRRADU86.

23. OUTPUT LINE COUNT PER PAGE NOW USER DEFINABLE

A new SMFUDFLT parameter, LINESPPG, and a new execution keyword LINESPERPAGE, now allow the user to control the way SMFUTIL paginates its output reports. A setting of '999' suppresses line counting and causes all reports to be produced without page breaks in the middle.

24. SYSTEM SYMBOLS NOW SUPPORTED

SMFUTIL now supports the use of system symbolic terms in the control statement set. If the SYMBOLS is set to "YES" in the SMFUDFLT defaults module and the symbolic substitution utility ASASYMBM is available in the system, SMFUTIL will call the utility for each input control statement that is read from the SYSIN data set. If any substitutions are performed, the modified control statement will be reprinted with the modified text. Calling the system symbolic utility may be disabled at execution time by specifying NOSYMBOLS as the first control card of the data set. This keyword should be on a line by itself.

25. CATALOG INPUT DATA SET LIMITS

The CATINPUT and GDGINPUT keywords have been enhanced to allow a limit of 1 to 255 to be specified for each data set prefix or GDG index level. This allows the user to predetermine the maximum number of input data sets to be processed during any given execution.

26. MULTIPLE SETS MAY BE SPECIFIED FOR ARCHIVE DATA RETRIEVAL

The SETNAME keyword and ARCHIVE search processing has been enhanced to allow multiple sets to be specified for SMF data retrieval via the archive.

27. DYNAMIC DEFAULTS OVERRIDE IMPLEMENTED

The defaults specified in the SMFUDFLT at installation time may be overridden at execution time via a PARMLIB member. This member is read prior to the processing of control statement specifications.

28. ALL CALENDARS NOW DYNAMICALLY LOADED.

Specifying a user calendar in the defaults table module SMFUDFLT via the MONTHCAL, MNTHPCAL or WEEKPCAL keywords no longer requires including the module into the link step. The module will be dynamically loaded at execution time.

SUMMARY OF AMENDMENTS - VERSION 8.0

The following items are new or revised over version 7.0 of SMFUTIL.

1. ASYNCHRONOUS CLEARING OF MULTIPLE MANX DATASETS

By specifying MULTICLR=YES in the SMFUDFLT module or the MULTICLEAR keyword at execution time, the DUMPCLEARALL function of SMFUTIL can now clear multiple MANx dataset simultaneously.

2. BUFFERING ABOVE THE LINE

All BSAM I/O buffers have been moved above the line for storage addressing constraint relief. This feature allows more output datasets to be created in a single pass of the input data.

3. SHORTCLEAR THRESHOLD SUPPORT

A “percentage full” threshold has been added to instruct SMFUTIL to automatically switch from a requested SHORTCLEAR to the more efficient (and faster) full clearing method when the MANx dataset is full or nearly full.

4. SAFE CLEAR OPTIONS INCORPORATED

During CLEAR processing SMFUTIL will now write a special indicator record as the first record of the dataset prior to completing the clear. This indicator record prevents SMFUTIL from placing the MANx dataset back into service as “AVAILABLE” in the event of a system outage and subsequent IPL during the CLEAR process. The dataset will remain “DUMP REQUIRED” and an SMFUTIL restart will be performed and the CLEAR completed. In addition, if SUCLEAR=YES is specified in the defaults module or SUCLEAR specified at execution time, the same special indicator record will be used to clear the entire dataset and only the first record will be overwritten with an IBM clear indicator record. This option prevents SMF from “MOD’ing on” to a MANx dataset during the IPL process. During IPL, only datasets that are actually CLEAR will look clear to SMF initialization. All others will look full. This prevents SMF from creating out of order data by reusing a previously partially used dataset before it can be dumped and cleared.

NOTE: If SUCLEAR is specified (or defaulted) the MAN1LEAVE option may be removed (NOMAN1LEAVE at execution time or MAN1LEAV=NO in SMFUDFLT defaults module) as SMF will no longer be able to reuse the primary MANx dataset until SMFUTIL has successfully cleared it and notified SMF of it’s availability.

5. LARGE BLOCK INTERFACE (LBI) SUPPORT

Support has been added for IBM’s Large Block Interface (LBI). Though the record length is still limited to 32,767, any device that supports the LBI specification can have block sizes larger than 32K. Current limits are approximately 64K for 3480 and 3490 devices and 256K for 3590 devices.

6. GENCENT PARAMETER REMOVED

For Year 2000 compatibility, the GENCENT defaults parameter has been removed. All suffix values containing year values will now contain both the century and the year. Sites still using GENCENT=NO in the SMFUDFLT module or specifying NOGENCENT at execution time must convert their data set name structures to include the century value.

7. SEPARATE SYNAD SUPPRESSION CONTROL FOR INPUT AND OUTPUT DATA SETS

SMFUTIL now supports suppression of the SYNAD exits for input or output data sets for debugging purposes.

8. TIME AND DATE/TIME LIMITATION FILTERS FOR INPUT DATA SETS.

SMFUTIL now supports limiting the input date/time or time values accepted from specific input DD names. The ITIMERANGE allows a user to set one or more specific time value ranges that are to be kept. The IDTRANGE allows a user to select one or more specific periods of time from an input source. Each input dataset may have one or more unique limits defined.

9. IMPROVED TYPETODD RECORD SELECTION

The TYPETODD keyword parameter has been enhanced to allow a list of record types to be specified for a single DD name.

10. ENHANCED SUFFIX CAPABILITY

Several new data set suffix definitions have been added including a YEAR suffix type and an EXTENDED TIME suffix.

11. USER CALENDAR GENERATION UTILITY INCLUDED

A new utility, SMFUGENC, designed to automate the generation of User Calendars, is now included in the base distribution.

12. SMF ID SYMBOLIC INCLUSION/EXCLUSION SUPPORTED

The SYSID and XSYSID keywords now support the &DID and &SID symbolic as the first system id specified. This allows a single system to be selected (the system providing input data or the system performing the execution) without knowing the system id ahead of time.

13. DTRANGE SYNTAX ENHANCED

The DTRANGE keyword now accepts new parameter syntax that makes it easier to specify the desired date/time range.

14. PROCESS LOGGING CAPABILITY ADDED

A new processing event recap log may be produced by the inclusion of a SYSLOG DD statement.

15. DDSIDKEEP/DDSIDSKIP KEYWORDS ADDED

Two new keyword, DDSIDKEEP and DDSIDSKIP, allows the user to specify specific SMF System ID's that are to acceptable or rejected for output to specific DD names.

16. SKIPHIDT KEYWORD ADDED

A new keyword, SKIPHIDT, allows the user to reject records with invalid future date values.

17. FIXED FORMAT REPORT SUPPORTED

A new keyword, REPORTFIXED, fixes all column widths in the SMFUTIL SYSPRINT report. This allows the user to develop a post processing program that can reliably read the report and summarize it.

18. IMPROVED DATESCAN REPORTING

The report produced by the DATESCAN function has been improved to show the record type and source DDNAME of the two records bounding any gap detected.

19. ABILITY TO SYNCHRONIZE (SINGLE THREAD) JOBS

A new execution parameter field, "SYNC" allows the user to single thread the execution of specific SMFUTIL functions between multiple jobs.

20. ABILITY TO DYNAMICALLY ALLOCATE SYSPRINT OUTPUT DATASET

If the SYSPRINT output DD statement is not available in the JCL, SMFUTIL will now dynamically allocate an output dataset for SYSPRINT. The user may direct this allocation to a spool dataset or to a sequential dataset on disk through various parameter overrides.

21. ARCHIVE BACKUP CLEANUP CAN NOW BE AUTOMATED

A new defaults value, ARCBNUMB, can be specified to indicate the maximum number of ARCHIVE backup datasets that SMFUTIL is to retain. SMFUTIL will automatically scratch and un-catalog older datasets above this count.

22. FINAL RECORD CHECKPOINT DATA VALUE FOR EACH OUTPUT DDA BLOCK

Each dynamically allocated output dataset may now have a unique checkpoint record to assist in detecting split boundaries that fall between executions. The keyword CKPTLAST can specify a unique checkpoint record name to be used for a dynamically allocated output dataset. This checkpoint record will be used in lieu of the LASTREC1 general purpose record.

23. MQSPLIT KEYWORD IMPLEMENTED

A new keyword, MQSPLIT, has been implemented to allow the user to split MQ Series record types 115 and 116 onto discrete output datasets based upon their subsystem name values.

24. CHECKPOINT PDS BROWSER ADDED

A new ISPF utility, SMFCKPUI, has been added to allow the user to examine the checkpoint records present in the Checkpoint PDS.

25. IMPROVED ARCHIVE REPORT UTILITY

The Archive Report Utility, SMFUARPT, has been enhanced to support a new keyword, USABLE, that causes the reports to be restricted to Archive Records that are usable to SMFUTIL for retrieving data. Defunct records will be ignored.

26. LIST ALL RECORD TYPES REGARDLESS OF COUNTS

A new keyword, LISTALL, causes SMFUTIL to list all record types from 0 to 255 on the "Detailed Statistics Report" even if the input count is zero for the record type.

27. AUTOMATIC CREATION OF CHECKPOINT PDS DATASETS

If it does not already exist and if the DACKPT parameter in SMFUDFLT is 'YES', SMFUTIL will now automatically create the CKPTDB dataset the first time it is specified. Parameter values in the SMFUDFLT module can be used to control the UNIT and VOLSER specification for this allocation.

28. SMF ID CONSISTENCY CHECKING

When a checkpoint dataset is present, SMFUTIL can now optionally check for an improper SMF ID in the input data. It does this by comparing the SMF ID of the input data with the ID retained in the \$PROCESS checkpoint record. If a mismatch is found, SMFUTIL will issue a warning message and set a minimum return code of 4 or optionally abort the execution.

29. CATALOG INPUT DATA SET LIMITS EXPANDED

The CATINPUT and GDGINPUT keywords have been enhanced to allow a specification to limit input to only a certain number of the latest datasets for an index. A 'latest' value of 1 to 255 may be specified for each data set prefix or GDG index level. When combined with the 'limit' value specification, this allows the user selective access to certain portions of a catalog list of datasets without processing the entire list.

30. CKPTGDG AND GDGCKPTI PROCESSING IMPROVED FOR INDEX WRAP

The CKPTGDG and GDGCKPTI keyword pair have been improved to automatically handle the situation where the GDG index wraps from 9999 to 0001. Data sets in this situation will be handled correctly and input in the correct order (i.e. order created). In addition, a BIDH sub-parameter keyword has been added for the user to be able to specify that, for ARCHIVEINPUT runs, datasets under the specified index are to have hourly BID values.

31. FORCED CREATION OF EMPTY DATASETS NOW MAY BE SPECIFIED AT THE DDA LEVEL FOR EACH DATASET

A keyword, DDACREATE, allow a DDA type dataset to be force allocated even if it has no output data. Alternately, a new keyword, DDANOCREATE, allows the suppression of a forced allocation for a specific dataset even though CREATEU0 or CREATS0 was specified.

32. DEFER/ALLOCATE SPECIFICATION ELIMINATED ON THE DDAEND KEYWORD

The positional keywords of DEFER or ALLOCATE have been removed from the syntax of the DDAEND keyword. Unused output datasets will never be allocated unless the CREATEU0, CREATES0 or DDACREATE keywords are specified as appropriate. The old syntax will still be honored but the specification will be ignored. For DDA data set names containing date sensitive suffixing or overlays, a new default parameter of CRECDATE will control what date is used for creating suffixes and overlays for empty datasets. A specification of 'YES' (the default as shipped) will cause the date from the first input record to be used. A specification of 'NO' will cause the executing systems current date to be used.

33. DDACOPY0RC KEYWORD SPECIFIES RETURN CODE FOR EMPTY DATASET

A new keyword, DDACOPY0RC, can now be used to indicate the return code to be issued when an output dataset receives no data. Previously the COPY0OK keyword could indicate a RC=0 for all output datasets in an entire run. Valid values are 0, 4, 8 or 12.

34. ARCHIVE BACKUP DSNAME CAN NOW BE TIED TO THE ARCHIVE CLUSTER NAME

If the ARCBDSN backup data set name prefix is specified as blanks, the ARCHIVE cluster name with qualifier '.BACKUP' appended to it will be used instead. It may be defined as a GDG index or it will be automatically suffixed with a DATE suffix for DAILY or specific DAY backups or with a DATE/TIME suffix if the backup frequency is ALWAYS. This has the added benefit of associating the ARCHIVE cluster with its backup datasets at the data set name prefix level. This is useful when multiple ARCHIVE data base structures are used.

35. ARCFORCE COMMAND ALLOWS ARCHIVE UPDATES FROM INTERRUPTED RUNS

A new keyword, ARCFORCE, allows SMFUTIL to force the update of the ARCHIVE database for ARCHIVE records that exist when a run is aborted for some reason.

36. BID VALUES NOW BE BY DAY OR HOUR

The Block ID (BID's) retained in the SMFUTIL ARCHIVE will automatically adjust to daily or hourly values depending on the output dataset split type. Daily datasets will have each BID slot represent the location of the associated hour on the tape. Weekly or Monthly datasets will have each BID slot represent the location of each day on the tape. For input data archiving or output datasets not being split, a BIDBYHOUR keyword is available within the DDA block to force hourly BID values.

37. ALTUNITS KEYWORD ALLOWS UNIT NAME SUBSTITUTIONS

The new ALTUNITS keyword allows a unit name to be substituted for a UCB device type to prevent emulated devices from being allocated. These devices typically cannot be used to mount real device media. An example of this is a 4mm SCSI DAT drive being used to emulate a 3490E tape drive. Even though the UCBTYPE device type strings are identical, SMFUTIL ARCHIVE SMF data on actual 3490E tapes cannot be mounted on the 4mm DAT drive. To prevent allocation from trying to use an emulated device as a real one a unit name for UCB device type substitution is required.

TABLE OF CONTENTS

1. INTRODUCTION	1
2. PROGRAM EXECUTION JCL REQUIREMENTS	3
2.1 EXECUTION LOGGING.....	5
3. DATA HANDLING METHODOLOGY	7
3.1 INPUT DATA HANDLING	7
3.2 OUTPUT DATA TARGETS.....	9
4. IMPLEMENTATION	11
4.1 INITIAL TESTING OF SMFUTIL	11
4.2 PRODUCTION PHASE IN – DATA MOVING.....	11
4.3 PRODUCTION PHASE IN – DATA DUMPING & CLEARING	13
5. INSTALLATION	15
5.1 PRODUCT DISTRIBUTION TAPE	15
5.2 UNLOADING THE PRODUCT TAPE.....	15
5.3 INSTALLING THE PRODUCT.....	16
5.4 ENSURING RESTART CAPABILITY	19
5.5 INSTALLING THE IEFU29 EXIT (OPTIONAL).....	19
6. PROGRAM EXECUTION CONTROL	21
6.1 INTRODUCTION TO CONTROL KEYWORD SPECIFICATIONS	21
6.2 CONTROL KEYWORDS - NORMAL	22
6.2.1 ACCOUNT	23
6.2.2 ACTIVEOK	23
6.2.3 ALL.....	24
6.2.4 ALLOW2	24
6.2.5 ALLOW3	24
6.2.6 ALTUNITS	25
6.2.7 ARCBACKUP	26
6.2.8 ARCFORCE	27
6.2.9 ARCHBIAS	27
6.2.10 ARCHINPT.....	28
6.2.11 ARCHIVDB.....	28
6.2.12 ARCHIVE.....	29
6.2.13 ARCHIVEACTIVE	30
6.2.14 ARCHIVERECOVER	30
6.2.15 ARCHSID	30
6.2.16 ARCHSMF	31
6.2.17 ARCHSMFRECNUM	31
6.2.18 ARCHSRCH.....	31
6.2.19 ARCHIVEUSER.....	32
6.2.20 ASMFIDLN	32
6.2.21 BADSNAPLIMIT.....	32
6.2.22 BGUSER.....	33
6.2.23 BIDBYHOUR.....	33
6.2.24 BIDINDEX	34
6.2.25 BIDPOINT.....	34
6.2.26 BYTEDETAIL.....	34
6.2.27 CATINPUT.....	35
6.2.28 CHKISORT.....	36
6.2.29 CHKOSORT	36
6.2.30 CICSSPLIT	37

6.2.31	CICSSUBC	38
6.2.32	CICSTYPE	39
6.2.33	CKPTDB	39
6.2.34	CKPTINPT	40
6.2.35	CLEAR	41
6.2.36	COPY00K	41
6.2.37	COPYBOTH	42
6.2.38	COPYVALUEBOTH	43
6.2.39	CREATES0	44
6.2.40	CREATEU0	44
6.2.41	CURRENTMONTH	45
6.2.42	CURRENTWEEK	45
6.2.43	DARETRY	46
6.2.44	DATASET	46
6.2.45	DATE	47
6.2.46	DATESCAN	47
6.2.47	DB2SPLIT	48
6.2.48	DCACHECKPOINT	49
6.2.49	DCBABENDEXIT	49
6.2.50	DCBRESET	50
6.2.51	DDEXIT	50
6.2.52	DDSIDKEEP	51
6.2.53	DDSIDSKIP	51
6.2.54	DDRETAIN	52
6.2.55	DECR	52
6.2.56	DELDUP	53
6.2.57	DELDUPRV	53
6.2.58	DELETERC	54
6.2.59	DETAIL	54
6.2.60	DISKONLY	55
6.2.61	DSPLIT (Obsolete)	55
6.2.62	DSPLITDP (Obsolete)	55
6.2.63	DTRANGE	56
6.2.64	DTVALOFF	57
6.2.65	DUMPCLEARALL	57
6.2.66	EDATE	58
6.2.67	EOF370	58
6.2.68	ESTAEON	58
6.2.69	EXCLUDE	59
6.2.70	EXCONLY	59
6.2.71	EXIT	59
6.2.72	FASTMOD	60
6.2.73	FORCE	60
6.2.74	GDATE	60
6.2.75	GDGCKPTI	61
6.2.76	GDGINPUT	62
6.2.77	GENDSND	63
6.2.78	GENDSNM	63
6.2.79	GENDSNw	64
6.2.80	GOTOEOF	64
6.2.81	IDTRANGE	65
6.2.82	INCLUDE	66
6.2.83	INCONLY	66
6.2.84	INDTVOFF	66
6.2.85	INITEXIT	67
6.2.86	INPUTDATADISTRIBUTION	67
6.2.87	INPUTDETAIL	68
6.2.88	INPUTLIMIT	68
6.2.89	IOELIMIT	68

6.2.90	ISSETNAME	69
6.2.91	ISMF	69
6.2.92	ISMFDATA	69
6.2.93	ITIME	70
6.2.94	ITIMERANGE.....	71
6.2.95	IUCB	72
6.2.96	IVOLUME	72
6.2.97	IVSTART.....	73
6.2.98	JOBNAME.....	73
6.2.99	LARGEBLOCK.....	74
6.2.100	LBI	74
6.2.101	LINESPERPAGE.....	74
6.2.102	LISTALL	75
6.2.103	LISTLICENSE.....	75
6.2.104	MANILEAVE	75
6.2.105	MANXALOC	76
6.2.106	MONTHCAL	76
6.2.107	MONTHPCAL.....	76
6.2.108	MQSPLIT	77
6.2.109	MSPLIT (Obsolete).....	77
6.2.110	MSPLITDP (Obsolete)	78
6.2.111	MULTICLEAR.....	78
6.2.112	NIORATIO	79
6.2.113	NOARCSID	79
6.2.114	NOLOCATE	80
6.2.115	NOSYSUT1	80
6.2.116	NOTIFY	80
6.2.117	ODTRANGE.....	81
6.2.118	ORDERCHECK	82
6.2.119	ORDERGAP	82
6.2.120	OSYNADOFF.....	83
6.2.121	OTIMERANGE	83
6.2.122	OUTPUTDETAIL	84
6.2.123	OUTPUTLIMIT.....	84
6.2.124	OUTPUTDATADISTRIBUTION	85
6.2.125	POSTEXIT.....	85
6.2.126	PREVIOUSMONTH	86
6.2.127	PREEXIT	86
6.2.128	PREVIOUSWEEK.....	87
6.2.129	PRINT	87
6.2.130	PROCESS	88
6.2.131	PROCSYS	88
6.2.132	RECEXIT	89
6.2.133	REPORTFIXED	89
6.2.134	RVALEXIT.....	89
6.2.135	SAT/SUN/MON/TUE/WED/THU/FRI	90
6.2.136	SEARCHARCHIVE	90
6.2.137	SETNAME.....	91
6.2.138	SCRATCHRETRY	91
6.2.139	SCLRTHLD	92
6.2.140	SDB.....	92
6.2.141	SHORTCLEAR	93
6.2.142	SHORTCOPY.....	94
6.2.143	SIDSPLIT	94
6.2.144	SIMULATE	95
6.2.145	SKIPHIDT	95
6.2.146	SKIPOLD.....	95
6.2.147	SMFIDCHECK.....	96
6.2.148	SORTTOLERANCE.....	96

6.2.149	SPIEOFF	97
6.2.150	SPINBACKUP	97
6.2.151	SRBUFF	97
6.2.152	SUBSET	98
6.2.153	SUBTYPE	99
6.2.154	SUBUFF	99
6.2.155	SUCLEAR	100
6.2.156	SWBUFF	100
6.2.157	SYMBOLS	100
6.2.158	SYNADOFF	100
6.2.159	SYSID	101
6.2.160	SYSTYPE	101
6.2.161	TAPEONLY	101
6.2.162	TERMEXIT	102
6.2.163	TERSE	102
6.2.164	TEXTSCAN	102
6.2.165	THSUFFIX	103
6.2.166	TIME	103
6.2.167	TODAY	104
6.2.168	TSOUSER	104
6.2.169	TYPETODD	105
6.2.170	USER1	106
6.2.171	USER2	106
6.2.172	USER3	107
6.2.173	USERID	107
6.2.174	VALUEONLY	108
6.2.175	VCBUFF	108
6.2.176	VRBUFF	108
6.2.177	VSAMMOD	109
6.2.178	VSINGLE	109
6.2.179	VWBUFF	109
6.2.180	WEEKEND	110
6.2.181	WEEKPCAL	110
6.2.182	WEEKSTART	110
6.2.183	WORKDAY	111
6.2.184	WSPLIT (Obsolete)	111
6.2.185	WSPLITDP (Obsolete)	112
6.2.186	XACCOUNT	112
6.2.187	XBGUSER	113
6.2.188	XDATASET	113
6.2.189	XERROFF	114
6.2.190	XJOBNAME	114
6.2.191	XSYSID	115
6.2.192	XSYSTYPE	115
6.2.193	XTIME	115
6.2.194	XTSOUSER	116
6.2.195	XUCB	116
6.2.196	XUSERID	117
6.2.197	XVOLUME	117
6.2.198	YEARFIRST	118
6.2.199	YEARFIRSTM	118
6.2.200	YEARFIRSTW	118
6.2.201	YESTERDAY	119
6.3	CONTROL KEYWORDS - DDA BLOCK COMMANDS	120
6.3.1	AEXPIRE	120
6.3.2	ASETNAME	120
6.3.3	DDABIDHR	121
6.3.4	BIDSTART	121
6.3.5	BLKSIZE	122

6.3.6	BUFNO	122
6.3.7	CKPTGDG	123
6.3.8	CKPTLAST	124
6.3.9	DATACLASS	124
6.3.10	DCB	125
6.3.11	DDALARGEBLOCK	125
6.3.12	DDAEND	126
6.3.13	DDASTART	126
6.3.14	DDACOPY0RC	126
6.3.15	DDAcreate	127
6.3.16	DDANOcreate	127
6.3.17	DDNAME	128
6.3.18	DISP	128
6.3.19	DSNAME	129
6.3.20	DUMMY	129
6.3.21	EXPDT	129
6.3.22	LABEL	130
6.3.23	LIKE	130
6.3.24	LRECL	130
6.3.25	MANAGEMENTCLASS	131
6.3.26	MODEL	131
6.3.27	RETPD	131
6.3.28	RECFM	132
6.3.29	RETRY	132
6.3.30	SPACE	133
6.3.31	SPLITDISP	134
6.3.32	STORAGECLASS	134
6.3.33	SUFFIX	135
6.3.34	TRTCH	137
6.3.35	UNIT	137
6.3.36	VOLCOUNT	137
6.3.37	VOLUME	138
6.3.38	VOLNUM	138
6.4	CONTROL STATEMENT EXAMPLES	139
7.	OPERATIONAL CONSIDERATIONS	141
7.1	LARGE BLOCK INTERFACE SUPPORT	141
7.2	SMFUTIL APF AUTHORIZATION	142
7.3	SMFUTIL EXECUTION SERIALIZATION	143
7.4	DYNAMIC ALLOCATION OF SYSPRINT OUTPUT DATASET	144
7.5	EXECUTION REGION SIZE REQUIREMENTS	146
7.6	SYMBOLIC SUBSTITUTION OF SYSTEM SYMBOLS	146
7.7	INSTALLATION DEFAULTS OVERRIDE AT EXECUTION	147
7.7.1	SPECIFYING THE PARMLIB DATA SET VIA SMFUDFLT	147
7.7.2	SPECIFYING THE PARMLIB VIA JCL	147
7.7.3	SPECIFYING THE MEMBER NAMES VIA SMFUDFLT	147
7.7.4	SPECIFYING THE MEMBER NAMES VIA "PARM="	148
7.7.5	SPECIFYING THE PARMLIB AND MEMBER VIA JCL	149
7.7.6	DEFAULT OVERRIDE EXCLUSIONS	149
7.7.7	LICENSE PARMLIB AND MEMBER VALIDATION	150
7.8	SYMBOLIC SUBSTITUTION FOR DEFAULT VALUES	150
7.9	RECORD SUBTYPES	150
7.9.1	SELECTING RECORDS BY RECORD SUBTYPE	150
7.9.2	SUBTYPE SELECTION CRITERIA	151
7.10	USERID RECOGNITION	151
7.10.1	SOURCE OF USERID	151
7.10.2	SECURITY SYSTEMS AND USERID'S	151
7.10.3	TYPE 30 RECORDS AND USERID'S	151
7.10.4	USERID MASKING	152

7.10.5	USERID CHECKING AND OTHER RECORD TYPES	152
7.11	JOBNAME RECOGNITION	152
7.11.1	TYPE 30 RECORDS AND JOB NAMES	152
7.11.2	JOBNAME MASKING	152
7.11.3	JOBNAME CHECKING AND OTHER RECORD TYPES	152
7.12	ACCOUNT CODE RECOGNITION	153
7.12.1	SOURCE OF ACCOUNT CODE	153
7.12.2	ACCOUNT CODE LOCATION WITHIN RECORDS	154
7.13	SELECTING RECORDS BY DATA SET NAME	155
7.14	SMF DATA ORDER CHECKING	156
7.15	CHECKPOINT DATA SET – USES AND RESTRICTIONS	157
7.15.1	CHECKPOINT BROWSER UTILITY	157
7.15.1.1	CHECKPOINT ISPF INTERFACE - INSTALLATION	157
7.15.1.2	CHECKPOINT ISPF INTERFACE - ACCESS CONTROL	157
7.15.1.3	CHECKPOINT ISPF INTERFACE - INVOCATION	158
7.15.1.4	CHECKPOINT ISPF INTERFACE - OPERATION	158
7.15.2	CHECKPOINT RESTART FOR MOD DATA SETS	160
7.15.2.1	TLMS AND MOD RESTART	161
7.15.2.2	CA1/TMS AND MOD RESTART	161
7.15.3	CHECKPOINT RESTART FOR NEW DATA SETS	161
7.15.4	CHECKPOINT PROCESSING FOR DUMPCLEARALL	162
7.15.5	CHECKPOINTING INPUT DATA	162
7.15.6	CHECKPOINTING DUMPING SEQUENCE	162
7.15.7	BYPASSING PREVIOUSLY PROCESSED DATA	163
7.15.8	HIGH SPEED CARTRIDGE POSITIONING FOR MOD PROCESSING	163
7.16	RECOVERING FROM B37/D37 ABENDS ON OUTPUT DATASETS	164
7.17	OUTPUT DATA SET I/O ERRORS	164
7.18	INPUT DATA SPLITTING	166
7.19	AUTOMATIC DATA SET NAME SUFFIX GENERATION AT ALLOCATION	167
7.20	CATALOGED DATA SET INDEX LEVEL INPUT	170
7.21	GDG DATA SET HANDLING	170
7.21.1	CONTROLLING RELATIVE GDG DSNAMES INTERNALLY	171
7.21.2	INPUTTING GDG GENERATIONS VIA SYSCKPT RECORDS	171
7.22	SELECTING INPUT DATA VIA THE CATALOG	172
7.22.1	INPUTTING ENTIRE GDG INDEXES	172
7.22.2	INPUTTING ENTIRE CATALOG INDEXES	172
7.22.3	AUTOMATED INPUT DATA SET DELETION	173
7.23	HIGH-SPEED CARTRIDGE POSITIONING	173
7.23.1	POSITIONING FOR DATA RETRIEVAL	173
7.23.2	POSITIONING FOR DATA SET EXTENSION (MOD)	174
7.24	MIH TIMING REQUIREMENTS	174
8.	ARCHIVING SMF DATA	175
8.1	DEFINING THE ARCHIVE DATABASE	176
8.2	SERIALIZATION OF UPDATES TO THE ARCHIVE DATABASE	177
8.3	PROTECTING UPDATES TO THE ARCHIVE DATABASE	178
8.4	RECOVERING ARCHIVE DATA	179
8.5	SUPPLYING THE ARCHIVE DATABASE NAME TO SMFUTIL	179
8.6	ARCHIVING OUTPUT DATA	180
8.7	ARCHIVING USER DATA SETS	180
8.8	ARCHIVING INPUT DATA	181
8.9	ARCHIVE BLOCK ID RETENTION	181
8.10	ARCHIVE DATABASE UPDATE SEQUENCE	182
8.11	ARCHIVING MULTIPLE SYSTEMS	182
8.12	RESTRICTIONS AND LIMITATIONS ON ARCHIVE PROCESSING	183
8.13	ARCHIVE DATABASE BACKUP	184
8.14	ARCHIVE DATABASE MAINTENANCE – BATCH MODE	186
8.15	ARCHIVE DATABASE MAINTENANCE – ISPF MODE	187
8.15.1	ARCHIVE ISPF INTERFACE - ACCESS CONTROL	187

8.15.2	ARCHIVE ISPF INTERFACE - INSTALLATION	187
8.15.3	ARCHIVE ISPF INTERFACE - OPERATION	188
8.16	ARCHIVE DATABASE REPORTING	190
8.17	RETRIEVING SMF DATA USING THE ARCHIVE DATABASE	191
8.18	ARCHIVED DATA SET UNIT NAME DEFINITION	193
9.	DYNAMIC ALLOCATION OF DATA SETS	195
9.1	DDA BLOCK EXAMPLES	195
9.2	DDA BLOCK SPLIT REQUESTS	195
9.3	DDA BLOCK DATA SET NAME SUFFIX GENERATION	196
9.4	DDA BLOCK SYMBOLIC SYSTEM ID SUBSTITUTION	196
9.5	AUTOMATIC DATE INSERTION INTO DATA SET NAME STRING	197
10.	MODIFICATION AND CUSTOMIZATION	199
10.1	SETTING EXECUTION CONTROL DEFAULTS	199
10.2	EXECUTION ACCESS CONTROL	207
10.3	RESTRICTED KEYWORD ACCESS CONTROL	208
10.4	DEFINING SUBSETS	208
10.5	MODIFYING THE RECORD ACCOUNTING TABLE	209
10.6	SECURITY EXIT – INSTALLATION AND USE	210
10.7	DEFINING INSTALLATION CALENDARS	211
10.8	AUTOMATING THE GENERATION OF USER CALENDARS	212
10.9	MODIFYING THE UNIT NAME TABLE	213
10.10	ESOTERIC TAPE UNIT NAME IDENTIFICATION	213
11.	SMF DATA COLLECTION METHODOLOGIES	215
11.1	THE COLLECTION AND ARCHIVING PROCESS	215
11.1.1	SINGLE SMF DATA SET DUMPING	216
11.1.2	MULTIPLE DATA SET DUMPING	216
11.1.2.1	DUMPCLEARALL PROCESSING	217
11.1.2.2	MANILEAVE OPTION	218
11.1.2.3	SMF DATA SET SWITCHING	218
11.2	CLEARING SMF DATA SETS	219
11.2.1	SAFE CLEAR PROCESSING	220
11.2.2	SHORTCLEAR THRESHOLD PROCESSING	221
11.3	DUMPING FREQUENCY	221
12.	USER EXITS	223
12.1	SMFUTIL DEFINED EXIT POINTS	223
12.2	COMPATIBILITY SUPPORT FOR IFASMFD TYPE EXIT MODULES	225
13.	SMFUTIL REPORTS	227
14.	RETURN CODES	231
15.	APPENDIX A - CONTROL & EXECUTION ERROR MESSAGES	233
16.	APPENDIX B - EXECUTION WARNING/INFORMATION MESSAGES	265
INDEX	271

1. INTRODUCTION

SMFUTIL is a general purpose SMF data movement utility designed to simplify the task of dumping, moving, organizing, validating, and selecting SMF data for post processing. It operates as a batch utility and provides extensive error correction and recovery facilities to optimize data extraction from damaged or incomplete input SMF data sources. Support for VSAM data sets (i.e. SMF) is included and a CLEAR function enables SMFUTIL to replace the IBM utility IFASMFDP at the functional level. Please note that type 2 and 3 records (dump header and trailer) are not produced by SMFUTIL. Output data may be written to tape or disk (sequential or VSAM). If invoked without any input control statements, SMFUTIL will perform as instructed in a site definable default functions table. This would normally be to copy all valid records from the input to the output. Input blocks are validated with respect to block length. Input records are always validated with respect to record length, header content (valid date/time), and record structure/segment order. Any record not meeting the validation criteria is discarded. Control statements may be used to cause the input data to be further validated and/or specific data records to be selected.

Data validation checks available are:

1. Input data in sorted order.
2. Output data in sorted order.
3. Missing date/time span checking.
4. Duplicate record deletion.
5. Duplicate data block deletion.

Data selection controls available are:

1. By record type.
2. By record subtype.
3. By date range (Julian, Gregorian, or European).
4. By time range (Normal and Inclusive).
5. By subset type (Job, RMF, JES, SECURITY, etc.).
6. By Work week day only.
7. By Weekend day only.
8. By one or more specific day of week by name.
9. By TSO User (TSU Type).
10. By Batch User (JOB/STC Type).
11. By Batch Job name (JOB/STC Types).
12. By User ID (JOB/STC/TSU Types).
13. By System SMF ID.
14. By operating system type.
15. By Account Number.
16. By Data set Name.

Data exclusion controls available are:

1. By record type.
2. By record subtype.
3. By date range (Julian, Gregorian, or European).
4. By time range (Exclusive).
5. By TSO User (TSU Type).
6. By Batch User (JOB/STC Type).
7. By Batch Job name (JOB/STC Types).
8. By User ID (JOB/STC/TSU Types).
9. By System SMF ID.
10. By operating system type.
11. By Account Number.
12. By Data set Name.

A PRINT function is provided to print the contents of selected records. Both a hex dump and EBCDIC translation are printed. Output data may be split on daily, weekly, or monthly boundaries (or any two at once). Specific record types can be directed to user specified DD names with the selected types optionally being also copied to the standard output data sets. A checkpoint restart facility is provided to prevent duplication of data on output MOD data sets in the event of a system failure or job cancellation.

2. PROGRAM EXECUTION JCL REQUIREMENTS

SMFUTIL may be executed as a batch task or called task in a TSO CLIST environment. It requires the following JCL (or like TSO allocations):

```
//job name      JOB (acct),'as required',CLASS=?
//SMFUTIL       EXEC PGM=SMFUTIL,REGION=0M
//SYSPRINT     DD SYSOUT=A
//BADSNAP      DD SYSOUT=A
//SYSUT1       DD DSN=SYS1.MAN?,DISP=SHR (or archive tape)
//SYSUT2       DD DSN=** output smf dsname **,
//              DISP=(,CATLG),
//              VOL=(,,xx),   xx=NUMBER OF VOLUMES
//              UNIT=??????
//SYSIN        DD *
input control statement set if required
/*
/* The following required only for data splitting
//SYSUT3       DD DSN=?????,UNIT=SYSDA,VOL=SER=??????,
//              VOL=(,,xx),   xx=NUMBER OF VOLUMES
/* The following are optional.  If present in the
/* JCL, SMFUTIL will automatically Duplex copies
/* of their corresponding DDNAME data set.
//SYSUT2D      DD DSN=**dsn for duplex SYSUT2 data set,
//              VOL=(,,xx),   xx=NUMBER OF VOLUMES
//              DISP=(,CATLG),UNIT=??????
//SYSUT3D      DD DSN=**dsn for duplex SYSUT3 data set,
//              VOL=(,,xx),   xx=NUMBER OF VOLUMES
//              DISP=(,CATLG),UNIT=??????
/* The following is required only if checkpoint
/* restart for MOD data sets or CLEAR function
/* is desired.
//SYSCKPT      DD DSN=SMFUTIL.CKPT.PDS,DISP=OLD
//SYSARCH      DD DSN=SYSX.SMFUTIL.ARCHIVE,DISP=SHR
```

Please note that the above is an example basic skeleton. Not all DD cards shown will be required, depending on the chosen functions, and additional DD cards may be needed as defined by special function needs (i.e. DUPSAP, DUPSAVE, SYSOUT, etc.). If the BADSNAP DD statement is supplied, invalid records that are discarded will be printed to SYSOUT if possible. Some invalid records cannot be printed (i.e. record length 0).

If the SYSUT1 input statement is omitted, and the SMFUTIL archive database is available, and the execution is attempting to retrieve one or more specific date/time range(s) of data, SMFUTIL will attempt to locate the data required via the archive. See the section titled "SUPPLYING THE ARCHIVE DATABASE NAME" on page 179 for more information on defining the SMFUTIL archive database during an execution.

If the CLEAR function is to be used, SMFUTIL should be executed from an authorized STEPLIB or JOBLIB library (i.e. one listed in the IEAAPFxx or PROGxx member of system parmlib) or from a data set contained in the LNKLSSTxx parm member of the operating system.

NOTE: SMFUTIL is not REENTRANT. Do not place it in SYS1.LPALIB or load it via IEALPAXx in SYS1.PARMLIB.

Input data sets may be VSAM (SMF MANx data set) or one or more (concatenated) physical sequential (PS) data sets. VSAM and non-VSAM data sets may not be concatenated nor may two VSAM data sets be concatenated together on SYSUT1. As an alternative, if SYSUT1 is not present, all DD statements beginning with SYSUT1 will be used for input. As many SYSUT1xx DD statements as required may be specified. All VSAM input data sets must conform to the required SMF allocation parameters required by the SMF system. VSAM data sets (i.e. SMF) may be optionally cleared for reuse by the SMF system.

For multiple volume input data sets, input may begin on any volume of the set. SMFUTIL will automatically drop any segment in the first block that is invalid because it is a continuation of a spanned record from the previous tape in the set. For catalogued data sets, the volume count parameter need not be specified in the JCL. An input control keyword (IVSTART) is provided to specify which volume number to start with.

Output files from SMFUTIL may be sequential or VSAM (SMF image VSAM files). VSAM output files must be allocated as SMF files are allocated. The member DEFVSAM in the INSTLIB may be used as an example for allocating VSAM output files. Output to VSAM files defaults to RESET mode meaning that the file will be overwritten. Any data already in the file will be lost. The VSAMMOD keyword may be specified to indicate that specific (or all) VSAM output files are to be extended (MOD'ed).

Sequential output files must have a record format of V, VB, or VBS. Any record format specification that indicated FIXED (F) or UNFORMATTED (U) will automatically be reset to VBS. This may be the case when a model DSCB is used to create a GDG and no other DCB parameters are specified. SMFUTIL will override the model DSCB's record format and reset to VBS. This will cause a message to be produced but does not cause a failure in the copy operation. If not specified, DCB parameters for output data sets will be automatically set to LRECL=32767, RECFM=VBS. If BLKSIZE is omitted and the SDB parameter in SMFUDFLT is set to 'NO' SMFUTIL will set a BLKSIZE value optimized for the type of device. If LBI (Large Block Interface) is NOT in effect, output tape is set to BLKSIZE=32767. If LBI is in effect, tape is set to the largest optimum LBI block size available for the device. Disk is set to the largest optimum full track (half track on 3380/3390) block size for the device type for supported devices. Supported device types are 2305, 3330, 3350, 3380, 3390, and 9345. If another device type is used or if alternate DCB characteristics are desired, the DCB parameters must be specified on the JCL DD statement. If SDB is set to 'YES' or a BLKSIZE value of '0' is specified for a dynamically allocated data set, SMFUTIL will honor the value in the expectation that the operating system will choose an appropriate BLKSIZE value with the exception that an LBI request will always be honored and an LBI block size used. Note that for JCL allocated data sets SMFUTIL cannot differentiate between an omitted BLKSIZE and a specified value of '0'. For the '0' to be honored, SDB must be set to 'YES' in the SMFUDFLT module or the SDB keyword must be specified at execution time.

If the DCB contains RECFM=VB with no other parameters, SMFUTIL will reset its chosen LRECL to the chosen BLKSIZE minus 4. If RECFM=VB is specified in conjunction with LRECL or BLKSIZE or both, SMFUTIL will validate the supplied and defaulted parameters for consistency. This process eliminates errors and aborted runs when the DCB is inconsistent within itself.

Installations that utilize SMS to automatically alter the DCB characteristics of output data sets should make certain that inappropriate DCB values are not chosen for SMFUTIL output data sets. When the DCB is not present in the JCL and SMFUTIL chooses the values, SMS can override these values, if allowed to do so, with an ABEND as the result. For example, assume that an SMS data class is set up that indicates that all data set names ending in '.DATA' are to have an LRECL of 80. If an SMFUTIL output data set data set name of 'SMF.MONTHLY.DATA' is used without a JCL DCB specification, SMFUTIL will set up the DCB properly but SMS will override it with LRECL=80. This will result in an S002-18 ABEND (MSG=IEC036I). An override JCL DCB specification will prevent the problem, but SMFUTIL automatic DCB optimization will be bypassed. The correct solution is to ensure that SMS knows about the SMF output data sets as a unique data class and will not alter their DCB characteristics.

For output data sets that have a MOD disposition and already exist, SMFUTIL will automatically start with the last volume of the series. This automatic MOD last volume mounting is functional for any type of output data set, even GDG sets. A copy run utilizing MOD to add data to an existing data set can be protected via an internal checkpoint processing routine. This prevents data from being added twice in the event of a system failure (or operator cancel) during the copy process.

The check pointing feature may also be used to indicate when a clear function against an SMF data set has not completed successfully and to prevent a production job stream from copying the same data twice (preventing duplication of data on an archive file as a result of accidentally rerunning the job).

During processing, the control keywords can be used to select and/or exclude certain records based on record type, date, and/or record content. Selected records are written to the normal SMFUTIL output files (SYSUT2, SYSUT2D, SYSUT3, and SYSUT3D) and/or the user specified DD names. In addition to the normal output data sets, SMFUTIL supports an 'exclude' data set DDNAME of SYSUT2X. If this DDNAME is present, any record not written to any other output file will be automatically written to the exclude file SYSUT2X.

2.1 EXECUTION LOGGING

If a SYSLOG DD statement is present in the execution JCL, SMFUTIL will produce a time-stamped execution log of the processing performed during the run. The amount of information produced is regulated by the LOGLEVEL setting. This level is a single digit number in the range of 1 to 5. The higher the number the greater the amount of logged information. The default LOGLEVEL value is 1 and cannot be changed at install time. The LOGLEVEL may be overridden at execution time by specifying an execution parameter of LOGx where “x” is a single digit of 1 to 5. A value of 1 produces major event logging. A value of 2 will produce checkpoint event logging. Values of 3 or higher cause logging for every record processed and may produce extremely large amounts of print in the SYSLOG dataset.

3. DATA HANDLING METHODOLOGY

3.1 INPUT DATA HANDLING

SMF data is input to SMFUTIL via the DDNAME SYSUT1 or any DDNAME that begins with SYSUT1. Input data sets may be provided by the user or be dynamically determined by SMFUTIL via the ARCHIVE database.

The user may explicitly supply one or more input data set via the JCL or by dynamically allocating them through control statements supplied to SMFUTIL. Input data may be from tape or disk and concatenation of unlike devices for sequential data sets (NON-VSAM) is fully supported on any input (SYSUT1/SYSUT1xx) DD statement. The active SMF data set may also be used for input, but the CLEAR function may not invoked upon it.

Multiple input DD statements are allowed only if SYSUT1 proper is omitted. If the DDNAME SYSUT1 is not present in the JCL or dynamically allocated, any DDNAME of the form 'SYSUT1xx' encountered via JCL or dynamically allocation (or both) will be used for input. The characters 'xx' must be unique on each one. Each SYSUT1xx DD statement may be a VSAM or sequential data set. These data sets will be processed for input in the order they are specified in the JCL and then via dynamic allocation control statements. The entire set of SYSUT1xx data sets comprises one logical input data set. This should be kept in mind when keywords that affect how far SMFUTIL processes into the input data set (such as GOTOEOF or SHORTCPY) are used.

Both VSAM and sequential data sets are supported in this alternative format and mixing is allowed. As an example:

```
//SYSUT1H      DD  DSN=SYS2.SMF.SYSA.MTD(0),DISP=OLD
//SYSUT1M1     DD  DSN=SYS1.MAN1,DISP=SHR
//SYSUT1M2     DD  DSN=SYS1.MAN2,DISP=SHR
//SYSUT1M3     DD  DSN=SYS1.MAN3,DISP=SHR
```

The above JCL could be used to input the current month-to-date tape and any data in the SMF files that had not been dumped as yet. This is a convenient way to ensure that all available SMF data is input to an SMF processing program such as a SAS routine or an RMF post processing job. This same input could be achieved more flexibly and with possibly better results by using the following control keywords:

```
.
.
//SYSIN        DD  *
DDASTART
    DSNNAME(SYS2.SMF.&SID.MTD(0)) DDNAME(SYSUT1H)
DDAEND
MANXALOC
```

The DDA block (DDASTART to DDAEND) defines the historical month-to-date data set as input on DD name SYSUT1H. The &SID parameter within the data set name is replace by SMFUTIL with the four character SMF system ID on which the job executes. The MANXALOC parameter causes all current SMF data set containing data to be allocated and logically concatenated to SYSUT1H. As an added advantage, the MANXALOC allocated data set will be in the proper order (i.e. oldest data first).

A further simplification of the required input could be achieved if the SMFUTIL ARCHIVE directory database were available to the run.

```
.
.
//SYSIN        DD  *
ARCHIVDB(SYSX.SMFUTIL.ARCHIVE)
THISMONTH MANXALOC
```

The ARCHIVEDB keyword tells SMFUTIL the cluster name of the ARCHIVE database. The THISMONTH keywords implicitly states a request for dates for selection of input data as those of the current month. The SMFUTIL archive search routine would search the ARCHIVE database and locate the input archive data set and allocated it for input. The MANXALOC keyword causes the SMF data set to be allocated as well for up to the minute SMF data to be available. Dynamic allocation of data set via the DDA block commands, and utilizing the ARCHIVE database directory to locate input data will be discussed in detail later in this manual.

If desired, multiple data sets may be concatenated on each SYSUT1xx DD name, with the exclusion of VSAM files. Each VSAM file must be on a unique DD. As before, unlike devices (disk and tape) for sequential data sets may be mixed on one DD name. If checkpoint processing is not in force (i.e. SYSCKPT DD statement NOT present in the execution JCL), input tapes will be handled sequentially and each will be closed upon completion. This allows the coding of UNIT=AFF=SYSUT1xx to conserve tape drives. SYSUT1xx in this case refers to the DDNAME of the first tape data set in the JCL. When checkpoint processing is in force, SMFUTIL does not close an input file until all input data has been processed. This is to insure that proper catalog processing will occur only after all data has been processed and the output file is complete. This is normally used only for DUMP and CLEAR executions and should not be a problem for day to day use in moving data around for analysis.

The CLEAR function may only be executed against a non-active SMF VSAM file pointed to by SYSUT1. One and only one SMF file may be Cleared per execution of a CLEAR function. An additional job step would be necessary to clear an additional SMF data set.

Note that when using SMF VSAM data sets for input via JCL or by way of a DDA block structure, the cluster level data set name should be specified, not the data component name (i.e. SYS1.MAN1 not SYS1.MAN1.DATA). SMFUTIL must access VSAM data sets at the cluster level.

3.2 OUTPUT DATA TARGETS

Output data from SMFUTIL is normally placed to DD nameSYSUT2. In addition, an automatic duplex (copy) of the data may be made by also supplying SYSUT2D to the execution. These two output files are also referred to as standard output data sets. Additional output data sets, using user defined ddnames, may be produced as well. These user defined data sets will receive data that is specifically targeted to them via the keywords TYPETODD, CICSSPLIT, or DB2SPLIT. These keywords allows subsets of one or more specific record types to be created. By default, records targeted to a user defined data sets are not placed to the standard output (SYSUT2/SYSUT2D) but this can be overridden via the COPYBOTH keyword for TYPETODD and COPYVALUEBOTH for CICSSPLIT and DB2SPLIT.

Output data sets may be defined in the JCL or may be dynamically allocated via the DDASTART/DDAEND keyword block usage. The dynamic allocation has several distinct advantages over JCL allocation, the most important being that the data set is not physically allocated until it is needed. This avoids tying up devices until they are actually required. In addition, previous devices are dynamically released after the data set has been closed.

Each output data set may be logically split on a date boundary (i.e. day, week, or month). For standard output DD names (i.e. SYSUT2, SYSUT2D, and SYSUT2X) this split causes the sixth character of the DD nameto be “bumped” to create a new ddname. For example, if a daily split is done on SYSUT2 and midnight is encountered in the data stream, SYSUT2 would stop receiving data, and SYSUT3 would begin receiving data. For user defined DD names of less than 8 characters (5 is the minimum) the DD names will be ‘padded’ with a suffix character to make it unique. For 8 character DD names, the eighth character will be overlaid with the suffix character. For example, DDNAME CICSOUT would become CICSOUT3 and DDNAME CICSDATA would become CICSDAT3.

If the output data sets were supplied in the JCL, all required DD names (i.e. both SYSUT2 and SYSUT3) must be available in the JCL for the split to work. If dynamic allocation blocks are used and the split is requested in the DDAEND parameter, the allocation of SYSUT3 is automatically done by SMFUTIL as a mirror image of SYSUT2, with only the data set name suffix or GDG index being changed.

SYSUT2D may also be split on a date boundary and it may be split differently from SYSUT2. After the split takes, the second data set is SYSUT3D.

A special output DDNAME, SYSUT2X, is reserved as a “spill” data set. If present in the JCL or dynamically allocated via control statements, SYSUT2X will receive all records input to SMFUTIL and rejected for any reason. For example, if you process an input data set and only select record type 30, if SYSUT2X is present it will receive all other record types. Special consideration must be given if data rejected via date and/or time is to be retained on the spill data set. When input data is to be filtered for one or more date/time ranges, normally SMFUTIL does a high speed bypass of input data blocks that do not contain any date ranges required. This is called a ‘SPIN’ of the input data set. It is designed to speed access to required data. This means that no input data from these input blocks is considered at the record level and thus will not be output to SYSUT2X. The SPIN process can be bypassed by specifying ‘NOSPIN’ in the input control stream. SYSUT2X would then receive all data input up to the first record that meets the first DATE/TIME requirement. Additionally, SMFUTIL normally terminated when it reaches data that is higher than the highest DATE/TIME requested. This is called SHORTCOPY. It is designed to avoid the overhead of processing data that is not required for the normal outputs. To enable the writing of records to SYSUT2X that contain DATE/TIME stamps after all requested DATE/TIME stamps are passed, you must specify GOTOEOF in the input control stream. This instructs SMFUTIL to continue processing the input data set(s) until it reaches the physical end of all input data sets.

The user may define output data sets using any valid DD NAMEnot reserved for use by SMFUTIL. These data sets are called directed output targets and will be used in control keywords designed to split data into specific subsets or classes based on record type (TYPETODD keyword) or data content with specific types (i.e. MQSPLIT, SIDSPLIT, CICSSPLIT, or DB2SPLIT keywords). The user DDNAMEs should be at least 5 characters in length if they are to be logically split on a daily, weekly, or monthly boundary. Any user defined data sets that request a boundary split in the DDAEND keyword will be “bumped” in the same way as SYSUT2 is bumped. A user defined DD nameof USERDD would be altered to USERD3 for the second data set, USERD4 for the third data set, and so on.

4. IMPLEMENTATION

The method chosen for implementation of SMFUTIL into productive use in an installation is largely dependent upon the local requirements. The wide range of capabilities of SMFUTIL open up a great number of approaches for managing SMF data that were not feasible prior to its availability. While it is not possible to exhaustively outline all possible approaches for using SMFUTIL, this section will attempt to outline a general approach for implementation.

4.1 INITIAL TESTING OF SMFUTIL

After unloading the product tape and installing the load modules, the first task is to ensure that the product is functional.

To ensure that SMFUTIL is functional, edit member IVPJOB of the INSTLIB. Modify the job card and STEPLIB and submit the job. This job will read the current SMF data set and write the data to a temporary data set just to make sure everything is basically functional. Do the same for IVPJOB2. This job does a more intensive test of some of the SMFUTIL features such as reading the catalog to find input datasets.

4.2 PRODUCTION PHASE IN – DATA MOVING

Any existing process (job) that copies SMF data from one place to another is a likely candidate for implementing SMFUTIL. This is especially true of 'production' jobs that pull SMF data from the archives and place a subset of it into a data set for post processing via SAS or RMF or other analysis tool.

SMFUTIL can be used to substantially speed up these processes, especially if multiple steps or jobs can be replaced with a single SMFUTIL execution that creates multiple output files. As an example, the control card set:

```
TYPETODD( 70:79-RMF, 4, 5, 14, 15, 30-JOBDATA )
TYPETODD( 110-CICS, 14:15-DISKDATA )
TYPETODD( 129:255-USERALL, 230-BBREC, 80:81-RACF )
```

Would create seven output data set on DD names RMF, JOBDATA, CICS, DISKDATA, USERALL, BBREC, and RACF with only a single pass of the input data. A single record type can be targeted to multiple files and a single file name can be targeted multiple times. Note in the illustration above, record types 14 and 15 are targeted to files JOBDATA and DISKDATA.

By default, if the TYPETODD keyword is specified, SMFUTIL will only create the output files requested. If we wanted a 'catch all' output file that the rest of the data was to go to, we could add the following card:

```
INCLUDE( 0-255 )
```

The TYPETODD keywords implicitly suppress all record types not selected. The INCLUDE card above 'turns on' record types not selected via a TYPETODD card. Any records encountered that are not specifically targeted to a user defined file (via a TYPETODD keyword) will be output to the normal SMFUTIL output file name SYSUT2.

In addition, if we wanted to make a master output file that contained all the records and create the user subset data set, we could add the keyword:

```
COPYBOTH
```

to the run. This will cause records targeted to specific DDNAMEs via a TYPETODD keyword to also be output to the SYSUT2 output data set (master file output). For example, the RMF record types 70 through 79 would go to DD nameRMF and to SYSUT2.

Alternatively, if only some of the record types targeted to user defined DDNAMEs are to be duplicated on the standard outputs, we could specify:

```
COPYBOTH( aaa:bbb, ccc, ddd )
```

where 'aaa' and 'bbb' represent a range of record types and 'ccc' and 'ddd' represent individual record types. These record types are to be copied to any designated TYPETODD target and also to the standard output data sets.

If a record that is split off to a user defined data set via a data value within the record (i.e. SIDSPLT keyword) is also to be copied to the standard outputs, the COPYVBTH keyword functions the same as the COPYBOTH keyword, except for value split records.

4.3 PRODUCTION PHASE IN – DATA DUMPING & CLEARING

The utilization of SMFUTIL to dump and clear the SMF data set(s) would be the final objective of implementation. The easiest way to accomplish this task is to trial it on a test or development system to perfect the JCL required. The IEFU29 sample exit distributed with SMFUTIL can be used to completely automate the execution of the process. The IEFU29 exit gets control whenever an SMF data set is marked 'DUMP REQUIRED'. This includes at automatic switch (when a data set fills up), at manual switch (when an operator performs an 'I SMF' command) and at IPL time during SMF initialization (if any data sets contain data other than the one recording begins on). The start command issued by IEFU29 during an IPL will be 'queued' until JESx becomes active. At this point, it will be run.

The JCL and control cards below are a sample of a system to dump and clear SMF automatically. This PROC can be found in the INSTLIB data set under name SMFUDUMP.

```
//SMFUDUMP PROC
//DUMP      EXEC PGM=SMFUTIL,REGION=6000K
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN     DD DSN=SMFUTIL.INSTLIB(SAMPDCA),DISP=SHR
```

Note that the above JCL assumes that SMFUTIL is in a LNKSTxx data set.

The name(s) of the SMF data sets to be dumped and cleared are determined by the SMFUTIL DUMPCLEARALL function. No input data set is defined by JCL or DDA block.

Input to the DUMP step is as follows (member SAMPDCA of INSTLIB):

```
DETAIL
COPY00K
ARCHIVDB(SYSX.SMFUTIL.ARCHIVE)
ARCHIVE
CKPTDB(SYSX.SMFUTIL.&SIDCKPT)
DDASTART
    DSNAME(SYSX.SMF.&DID.NORMAL)
    DDNAME(SYSUT2)
    SUFFIX(DM)
    UNIT(TAPE)
    DISP(MOD,CATLG,KEEP)
DDAEND(DEFER,SPLIT(M,2))
DDASTART
    DSNAME(SYSX.SMF.&DID.DUPLEX)
    DDNAME(SYSUT2D)
    SUFFIX(DM)
    UNIT(TAPE)
    DISP(MOD,CATLG,KEEP)
DDAEND(DEFER,SPLIT(M,2))
DUMPCLEARALL
```

The control card set issues commands to SMFUTIL on how and where to place the data. The 'ARCHIVE' keyword tells SMFUTIL to record the location of the dumped data in the SMFUTIL Archive Directory database, the name of which is supplied by the 'ARCHIVDB' keyword. A checkpoint data set is provided via the 'CKPTDB' keyword. This is a one track PDS data set that is all directory blocks. If the dataset does not currently exist, SMFUTIL will automatically allocate it (as long as DACKPT=YES is specified in the SMFUDFLT module). Alternately, use ISPF, option 3.2 to allocate the CKPT PDS dataset and specify 40 directory blocks and 1 track. The DCB can be FB/80/800. Note that only the directory area is used. The checkpoint data set will never contain data as such and will never need compressing. The checkpoint data set is used to protect against duplicated data in the case of run failures and provides various other protection services. The name of the checkpoint data set in the above example contains an '&SID' parameter. This is automatically overlaid by SMFUTIL with the SMF system ID on which the run is executed. Note that the above example would require that the SMF ID for the system begin with an alpha character to make it a legal data set name qualifier. If your system uses numeric SMF ID's, you could change the data set name construct to 'SYSX.SMFUTIL.CKPT&SID' to make it legal. Each SMFUTIL 'process' must have a unique checkpoint data set. This means that the 'SMFUDUMP' started task would need a unique one and any subsequent production job that handled SMF data on an automated basis would need one. Each system ID would require a different set of checkpoint data sets.

The two standard SMFUTIL output files are SYSUT2 and SYSUT2D. In the above example, they are dynamically allocated on a standard tape device. The DCB for the output data sets will be filled in automatically by SMFUTIL. Each data set name has the '&DID' parameter in it. This is automatically replaced by SMFUTIL with the SMF ID from the first SMF record to be output to the data set. This override symbolic capability within the data set name allows a single set of control cards to dump and clear all systems in the shop. During dump and clear, of course, this would be the same as the SMF id of the system where the task was being executed. The 'SUFFIX(DM)' keyword requests that a suffix be appended to the data set name prior to allocation. An SMFUTIL suffix is always created from a date. The 'D' requests that the date from the first record be used to create the suffix. The 'M' requests the format of the suffix be monthly (i.e. '.DEC2002'). A split of the data on a month boundary is requested for both output files and up to 2 months (1 split) is to be anticipated. During split processing, the new data set will be written to a new DD name. For a SYSUT2 split, the new file would be SYSUT3. For SYSUT2D, it would be SYSUT3D. The DISP parameter for the new data set will be automatically set to 'NEW,CATLG,DELETE'. This can be overridden via the SPLTDISP keyword of the DDA block command structure.

After the DUMP process is successfully executed, all data sets are closed, and ARCHIVE processing is complete, the DUMPCLEARALL processor invokes a clear function on each SMF data set that was dumped.

Note that the checkpoint data set is required and must be unique to this system and DUMPCLEARALL process. If the DUMPCLEARALL processing is interrupted and does not complete normally (i.e. canceled by operator prior to completion), a remaining SYSCKPT \$DUMPALL record will cause a subsequent execution to be restarted at the appropriate point.

5. INSTALLATION

The installation of the product consists of the following steps:

1. Unload the product tape. (See Section 5.2 below for more information.)
2. Modify the defaults module (use the copy named USERDFLT in INSTLIB) to local requirements. Modify the AUTHUSER, AUTHWORD, and WORDUSER tables in the USERDFLT source member to control access to SMFUTIL execution and to critical keywords of SMFUTIL. By default these tables allow full access to SMFUTIL and to all SMFUTIL keywords for all users. See the section entitled "MODIFICATION AND CUSTOMIZATION" on page 199 for more information on the keyword definitions in the defaults module and their usage.
3. Modify the LICENSE member of the INSTLIB data set to contain the license information from the product cover letter.
4. Modify the INSTALL job stream in the INSTLIB data set. Change the 'DISTOBJ=', 'INSTLIB=', and 'LINKLIB=' parameters to match local requirements.
5. Submit the INSTALL job stream to install SMFUTIL into the specified LNKSTxx data set or authorized library for STEPLIB data set usage.
6. Perform a link list refresh (F LLA,REFRESH).

5.1 PRODUCT DISTRIBUTION TAPE

The distribution tape is a standard label tape with VOL=SER=SMFUTL. It contains 5 files. The files are named as follows:

1. SMFUTIL.INSTJCL
2. SMFUTIL.INSTLIB
3. SMFUTIL.DISTOBJ
4. SMFUTIL.ISPPLIB
5. SMFUTIL.ISPMLIB

5.2 UNLOADING THE PRODUCT TAPE

File 1 is a JCL job stream sequential file that may be IEBGENER'ed off to facilitate down loading the other files. The following JCL may be used as an example to accomplish this:

```
//UNLOAD1      JOB (acct),'as required',CLASS=?
//GENER        EXEC PGM=IEBGENER
//SYSPRINT     DD SYSOUT=A
//SYSIN        DD DUMMY
//SYSUT1       DD DSN=SMFUTIL.INSTJCL,DISP=OLD,
//              UNIT=TAPE,
//              VOL=SER=SMFUTL
//SYSUT2       DD DSN=?????,UNIT=SYSDA,VOL=SER=?????,
//              SPACE=(TRK,(1)),
//              DCB=(LRECL=80,BLKSIZE=????,RECFM=FB)
```

5.3 INSTALLING THE PRODUCT

NOTE: In most installations, SMFUTIL is licensed by processor on a calendar period basis. The product is sensitive to the serial and model number of the processor on which it is executed and the date of execution. Each processor to execute the product must be licensed and appear in the license table matrix that is inserted during installation or a SITE license must be obtained. Up to 16 individual serial numbers may be specified. Beyond 16, a site license is required. A site license will enable the product on all processors at a single location.

Each Customer site is supplied with a personalized cover letter containing the CPU's that are licensed to operate SMFUTIL, the date of the expiration of the license, and a code key value that enables the execution of the product on the specified processors. **THIS INFORMATION MUST BE CORRECTLY SPECIFIED DURING THE INSTALLATION OF THE PRODUCT FOR SMFUTIL TO BE EXECUTABLE.**

SITE licensed installations will be able to execute the product at a “HOT” site for disaster recover purposes. For non-SITE licensed installations, special provision has been made for execution of the product at an alternate site. If your disaster recovery plans call for collection of SMF data at the “HOT” site, contact ASPG Technical Support in the event of an emergency and a special code will be supplied to enable the product at the “HOT” site.

Once unloaded from the product tape, SMFUTIL must be tailored to local requirements and installed via the INSTALL job stream in the INSTLIB data set. Local modifications to the execution defaults consists of making changes to a supplied source module named SMFUDFLT and other tables as required. All modifications are optional and are detailed in the section entitled “MODIFICATION AND CUSTOMIZATION” on page 199. If desired, all changes may be deferred until a better understanding of the meaning of the changes to be made is developed. Many installations do not have to make any modifications at all and run with the defaults module as shipped.

The following tables are assembled by the INSTALL job:

TABLE NAME	PURPOSE
SMFUDFLT*	SMFUTIL execution defaults defined by the site including SMFUTIL authorizations for userids.
SMFUUNIT*	Unit name substitutions for ARCHIVE extracts in non-authorized environments and for detection of tape devices prior to dynamic allocation for output datasets.
SMFCICTB	CICS version identification table.
SMFDB2TB	DB2 version identification table.
SMFUTLRT	Record type structure identification table.
SMFSSTBL	Subset name table.
ADBUUSRS*	Archive Database access authorization table for ISPF users.
CKPUUSRS*	Checkpoint dataset access authorization table for ISPF users.

* The member name may be overridden by the user to specify a local copy of the table.

The DISTOBJ and INSTLIB parameters of the INSTALL job stream specify the names of the data sets unloaded from the product tape. The LINKLIB parameter specifies the target LNKSTxx data set for the load modules to be installed in. The DFLT SRC parameter specifies the name of the defaults source member. This may be a locally modified copy of the SMFUDFLT member.

The cover letter that was shipped with the product tape contains a PRODUCT, VERSION, DATE, LICENSE CODE, and a list of CPU serial numbers that your site is licensed for. This information must be inserted into the “LICENSE” member of the INSTLIB data set prior to running the INSTALL JCL job stream. Modify only the information to the right of the “=” sign as indicated in the cover letter. Prior to modification, the LICENSE member to be modified looks like this:


```

[ LICENSE ]
PRODUCT=SMFUTIL
VERSION=700
DATE=0000000
LICENSECODE=0000000000000000
CPUSER#1=00000000
CPUSER#2=00000000
CPUSER#3=00000000
CPUSER#4=00000000
CPUSER#5=00000000
CPUSER#6=00000000
CPUSER#7=00000000
CPUSER#8=00000000
CPUSER#9=00000000
CPUSER#10=00000000
CPUSER#11=00000000
CPUSER#12=00000000
CPUSER#13=00000000
CPUSER#14=00000000
CPUSER#15=00000000
CPUSER#16=00000000

```

For documentation purposes, the LICENSE member may contain comment lines by placing an “*” in column 1. Note that ALL of the existing VERB lines, as shown above, must be present.

Insert the information from the cover letter EXACTLY as shown. Do NOT omit leading or trailing zeros. The date field is 7 decimal digits long and each serial number field is 8 characters long. The CODE field is 16 hex digits long. All unspecified serial numbers are left as ‘00000000’. After modification, the LICENSE member it would look something like the following:

```

[ LICENSE ]
PRODUCT=SMFUTIL
VERSION=800
DATE=2004003
LICENSECODE=1234ABCD1234ABCD
CPUSER#1=12342064
CPUSER#2=1F120700
CPUSER#3=00000000
CPUSER#4=00000000
CPUSER#5=00000000
CPUSER#6=00000000
CPUSER#7=00000000
CPUSER#8=00000000
CPUSER#9=00000000
CPUSER#10=00000000
CPUSER#11=00000000
CPUSER#12=00000000
CPUSER#13=00000000
CPUSER#14=00000000
CPUSER#15=00000000
CPUSER#16=00000000

```

This is for an example site only and is not a valid license. Note that this site only has two active serial numbers. The remaining ones are left as ‘00000000’. You must use the information from your “LICENSE CODE REPORT” cover letter. Do not alter any serial number unless instructed to do so by the cover letter.

The INSTALL job assembles the installation defined defaults set, applies all outstanding maintenance to the product, inserts the product license date, code, and valid CPU serial numbers, and links all SMFUTIL product modules to an installation defined LNKLSTxx data set. SMFUTIL must be APF authorized for the CLEAR or DUMPCLEARALL function to perform correctly. It is recommended that it be placed in a LNKLSTxx data set. As an alternative, a authorized STEPLIB data set may be used. In addition, in an authorized environment DDA block requests for dynamic allocations that cause mounts to be issued will appear to the operator as normal mount commands. No reply option that allows the operator to deny the mount will be issued. Note also, for the “DUMPCLEARALL” function to be usable, SMFUTIL must be authorized. See the section titled ‘SMFUTIL APF AUTHORIZATION’ on page 142 for more information.

Note that a valid and current license must be supplied in the LICENSE member of the INSTLIB data set in order for installation to proceed. After a successful installation, the License Code information may be overridden at execution time. See the section titled “INSTALLATION DEFAULTS OVERRIDE AT EXECUTION” on page 147 for more information.

From time to time a new license code will be required to enable the program for a new license period. A warning message will be given 45 days prior to expiration of the license code. The INSTALL should be run again with the new code. An ASPG Sales representative will supply you with the appropriate code as required.

Note that the license code is CPU serial sensitive. If you add or change a processor you must obtain a new license code to enable SMFUTIL to execute on the processor with the new serial number.

A separate section entitled “ARCHIVE ISPF INTERFACE – INSTALLATION” on page 187 gives instructions for implementing the ISPF interface to the SMFUTIL ARCHIVE database. This interface is only required if an ARCHIVE database is to be utilized.

Access to the ARCHIVE ISPF interface can be controlled on a userid basis via a table module called ADBUUSRS. By default, full access is allowed to all Userid's. This table can be used to limit access of to allow certain userid' read only access. To implement limited access follow the instructions in the ADBUUSRS table source module in the INSTLIB data set, then rerun the INSTALL job to re-assemble and re-link the modified table with SMFUADBU into a LNKLSTxx data set for access from TSO ISPF sessions

A separate section entitled “CHECKPOINT ISPF INTERFACE – INSTALLATION” on page 157 gives instructions for implementing the ISPF interface to the SMFUTIL ARCHIVE database.

Access to the CHECKPOINT ISPF interface can be controlled on a userid basis via a table module called CKPUUSRS. By default, full access is allowed to all Userid's. This table can be used to limit access of to allow certain userid' read only access. To implement limited access follow the instructions in the CKPUUSRS table source module in the INSTLIB data set, then rerun the INSTALL job to re-assemble and re-link the modified table with SMFCKPUI into a LNKLSTxx data set for access from TSO ISPF sessions

Note that the load modules for the ISPF components of SMFUTIL are installed by the INSTALL job but to be implemented local modifications to the ISPF menu must be performed. See the sections on page 157 and on page 187 for details on how to activate the ISPF interfaces.

5.4 ENSURING RESTART CAPABILITY

For the automatic restart of MOD tapes to be operational, SMFUTIL must be able to reopen an existing tape for output in order to remove duplicated data by writing an end of file mark. Tape management systems refer to this as a double open. Changes may be required to the site's tape management system for this to be enabled. CA1/TMS shops should refer to the TMS section on page 161 for more information. CADynam/TLMS shops should refer to the TLMS section on page 161 for more information.

5.5 INSTALLING THE IEFU29 EXIT (OPTIONAL)

The IEFU29 exit point within the MVS system receives control whenever an SMF data set is marked "DUMP REQUIRED". This includes normal switches (i.e. a data set fills up), an operator demanded switch (i.e. "I SMF" command issued), or at IPL time when a data set other than the primary has data in it. There are two versions of an IEFU29 exit distributed with SMFUTIL. The IEFU29S source code is for installations that DUMP and CLEAR a single SMF MANx data set at a time (old method). The IEFU29 source code is for installations that have implemented DUMPCLEARALL. The main difference is that IEFU29S passes a data set name parameter to the dump job for the single data set to be dumped. IEFU29 contains ENQ handshaking logic with SMFUTIL to prevent more than one dump job from being started when SMFUTIL forces an additional switch of data sets.

The chosen exit may be installed in one of two ways:

1. The old way is to assemble and link the module to SYS1.LPALIB under the name IEFU29 and then IPL the system with a CLPA.
2. For ESA 5.1 and up, the **RECOMMENDED** way is to assemble and link the module to a LNKLSTxx data set using a name of SMFUU29. The INSTU29 member of the INSTLIB data set can be used for this purpose. Perform an "F LLA,REFRESH" to make the module available. Place the following line into a PROGxx member name in SYS1.PARMLIB:

```
EXIT ADD EXITNAME(SYSSTC.IEFU29) MODNAME(SMFUU29)
```

Perform a "T PROG=xx" command on the console where the "xx" is the PARMLIB member name suffix. This will place the exit into use. Remember to add the "xx" to the "PROG=" line in the IEASYS00 member of PARMLIB so the exit will be planted automatically at IPL.

To ensure the exit is loaded automatically at IPL time you must add the PROGxx member suffix to the 'PROG=' suffix list in the appropriate IEASYSxx member of SYS1.PARMLIB. Alternatively, the PROGxx member created above can be folded into an existing PROGxx member already activated at IPL time.

NOTE: The IEFU29 and IEFU29S source code contain an optional WTOR that will be issued only when an IPL in progress is detected. This reply causes the IEFU29 exit to wait on the System Operator to issue a reply before proceeding. This delay is to allow for a time period during IPL for other devices required for the dumping of SMF data to become available (such as a SILO that must be initialized).

6. PROGRAM EXECUTION CONTROL

6.1 INTRODUCTION TO CONTROL KEYWORD SPECIFICATIONS

SMFUTIL may be operated under controls specified by input statements. These statements are 80 byte records input on the 'SYSIN DD' statement. Any control statement with an asterisk (*) in column one will be ignored and may be used as a comment card. Columns 1 through 71 are scanned. All other columns (72-80) are ignored. The control statements contain keyword control verbs that instruct SMFUTIL what data to copy and what data quality checks to perform. As many control cards as necessary may be used. Control keywords may not be continued from one statement to the next but control words may be repeated on multiple statements as many times as is necessary. Control keywords will be explained in detail in the next section. All text encountered on control statements will be checked for recognition as a keyword. No embedded comments are allowed.

Control keywords must be separated from each other by at least one space. Some keywords have required or optional parameter list that are enclosed in parenthesis (). The parameter list must immediately follow the keyword with no spaces. When multiple items are specified within a parameter list, they must be separated by commas with no spaces. In certain cases (i.e. Userid's and Job names) spaces are allowed as padding to the right and serve as significant place holders. This will be explained in detail when appropriate.

In general, the control keywords are scanned as they are encountered. Each keyword has an additive (or subtractive) effect when it is encountered. For example, if we were to specify

SUBSET(JOB) EXCLUDE(30)

type 30 records would not be copied even though they are part of the JOB subset specification because they have been excluded after the JOB subset was encountered. On the other hand, if we specify

EXCLUDE(30) SUBSET(JOB)

type 30 records would be copied because they were included implicitly via the JOB subset specification after they had been excluded explicitly.

There is some interdependency of certain keywords. For example, any keyword that refers to a DDNAME must occur after the specified DDNAME is defined. If the DDNAME is contained within the JOB JCL the keyword specifying the DDNAME may be placed anywhere within the control stream. If the specified DDNAME is to be dynamically allocated, the DDASTART/DDAEND block structure that defines the DDNAME must occur before any keyword that specifies (targets) the DDNAME. As a general rule, any keyword that is dependent upon a condition or structure that is defined by other keywords must come after the keywords that set up the required structure or condition.

Keywords are divided into two distinct categories. NORMAL and BLOCK keywords. NORMAL keywords are stand-alone in nature and are complete in and of themselves. BLOCK keywords are set apart from NORMAL keywords by a block start and block end notation. Between a start and end for a block only keywords defined as valid for that block type may be specified. Currently, the only defined block type is DDA or Dynamic Data set Allocation block commands. A typical block definition for a DDA block would be as follows:

DDASTART
DSNAME(SYS1.MAN1)
DDNAME(SYSUT1)
DDAEND

No 'NORMAL' keywords may appear between the DDASTART and DDAEND keywords. Only commands defined as DDA block commands are valid.

6.2 CONTROL KEYWORDS - NORMAL

Control keywords have the following general format:

KEYWORD[(parameter-list)]

In the following technical definitions, the required keywords will be shown in UPPERCASE. User supplied parameters contained within parenthesis will be shown in lower case. If a parameter list is required, it will be shown adjacent to the keyword. If a parameter list or any part of a parameter list is optional it will delimited by brackets []. The brackets are never coded in control statements. They are used here for clarification of required versus optional parameters only.

EXAMPLE: INCLUDE(rec-type[,rec-type,...])

The INCLUDE keyword requires that at least one record type number be specified. More than one may be specified if desired. If more record type numbers are required than will fit on one card, an additional INCLUDE keyword may be specified on the next control statement. All parameter lists must be terminated with a close parenthesis in or before column 71.

Each keyword will be illustrated with the following notation:

FORMAT:	KEYWORD(parmlist)
KEYWORD	Shows the exact spelling required to invoke the keyword being discussed.
parmlist	Shows the parameters that the user is to enter. Elements in UPPER CASE are constants and should be entered exactly as shown. Elements in lower case are variables and are replaced by an appropriate value as required by the user. Elements shown in brackets [] are optional and may be omitted if not desired.
ALIAS:	KEYWORD
	An alternate spelling of the keyword that may be easier to remember for some users.
INVERSE:	KEYWORD
	A KEYWORD with the opposite meaning of the one being presented. This inverse is used to negate the operative keyword if it has been set by default in the installation site.
PARM-LIST:	REQUIRED/OPTIONAL/NOT ALLOWED
REQUIRED:	If the keyword is specified in the input control set, an associated parameter list must be coded.
OPTIONAL:	The keyword has a default value that will be explained. The parameter list need only be coded to change the default value.
NOT ALLOWED:	The keyword does not use a parameter list and will be flagged in error if one is coded.
EXAMPLE:	Illustrates, for clarity, alternate methods of coding the keyword and describes the results.
FUNCTION:	Describes the purpose of the keyword and the processing results obtained by it's use. Restrictions or cautions in the use of the keyword are also detailed.
NOTE:	References the user to other sections of this manual that give more information that may have a bearing on the use of the keyword or gives additional information on interaction of the keyword with the use of other keywords..
R-CODE:	Shows the effect that action by the keyword may have on the final execution return code. The highest return code encountered will be returned at the end of the run.

6.2.1 ACCOUNT

FORMAT:	ACCOUNT(acctcode[,acctcode,...])
acctcode	A 1 to 8 character accounting string mask. If less than 8 characters are specified, all accounts that start with the specified string will be selected. To limit the selection to a specific account that is less than 8 characters, it is necessary to pad the string to the right with blanks within the parenthesis. A wildcard specification of '*' (an asterisk) may be used in any position to specify that any value is to be accepted in that position.
PARM-LIST:	At least 1 account code specification required.
FUNCTION:	To restrict record selection to SMF records that pertain to the specified account code(s). Records from the ACCOUNT subset with the selected account(s) will be selected. Only record types with account capability are screened. Other record types that are otherwise selected will bypass account validation and be processed.
EXAMPLE:	ACCOUNT(AC01) would select all account's that begin with "AC01". ACCOUNT(AC01) would select only records for which the account code was "AC01". ACCOUNT(**01) would select only records for which the account code has the characters '01' in the third and forth positions. This is equivalent to ACCOUNT(**ID***).
NOTE:	Remember that not all records that support an account code field will contain one and some may have a truncated portion of the account field from the job card (if more than 4 characters was specified as an account code). See the section titled "ACCOUNT CODE RECOGNITION" on page 153.
R-CODE:	0 Data for Account copied. 4 No data found for Account.

6.2.2 ACTIVEOK

FORMAT:	ACTIVEOK
INVERSE:	NOACTIVE
PARM-LIST:	NOT ALLOWED
FUNCTION:	To instruct SMFUTIL that input of the active SMF data set is acceptable. As shipped, ACTIVEOK is defaulted. Except when DUMPCLEARALL is used, NOACTIVE should be specified in the production executions that dump the data sets to the archives to prevent the active data set from being accidentally dumped. The rational behind this is that the copy (dump) step would succeed and the following CLEAR step would fail because the data set was still active. This would cause data to be possibly placed twice to the archive files.
R-CODE:	NO EFFECT FOR ACTIVEOK 0 - NOACTIVE specified and file is not active. 8 - NOACTIVE specified and file is active.

6.2.3 ALL

FORMAT: ALL
INVERSE: NOALL
PARM-LIST: NOT ALLOWED
FUNCTION: To instruct SMFUTIL to copy all records from the input to the output data set.
R-CODE: NO EFFECT.

6.2.4 ALLOW2

FORMAT: ALLOW2
INVERSE: NOALLOW2
PARM-LIST: NOT ALLOWED
FUNCTION: To instruct SMFUTIL to allow type 2 records to be passed to the output phase. If ALLOW2 is not specified, all type 2 records will be discarded on input without impacting any checkpoint processing or date/time statistics being kept. The default in the SMFUDFLT module is ALLOW2=NO meaning that type 2 records will be filtered out and not processed. This default may be changed if desired locally. If changed, a specification of NOALLOW2 in the input stream will cause filtering of type 2 records to be invoked for the current execution.
R-CODE: NO EFFECT.

6.2.5 ALLOW3

FORMAT: ALLOW3
INVERSE: NOALLOW3
PARM-LIST: NOT ALLOWED
FUNCTION: To instruct SMFUTIL to allow type 3 records to be passed to the output phase. If ALLOW3 is not specified, all type 3 records will be discarded on input without impacting any checkpoint processing or date/time statistics being kept. The default in the SMFUDFLT module is ALLOW3=NO meaning that type 3 records will be filtered out and not processed. This default may be changed if desired locally. If changed, a specification of NOALLOW3 in the input stream will cause filtering of type 3 records to be invoked for the current execution.
R-CODE: NO EFFECT.

6.2.6 ALTUNITS

FORMAT: ALTUNITS[(devtype-unitname,...)]

INVERSE: NOALTUNITS

PARM-LIST: OPTIONAL

devtype An 8 hex digit UCB device type

unitname A 1 to 8 character unit name to be used to replace the device type in allocations.

FUNCTION: To instruct SMFUTIL to always use either the optional user specified unit name as a substitute for the defined device type or (if a parm list is not specified) to use the Tape Alternate Unit Name Table, SMFUTALT, to substitute a unit name for a UCB device type string during allocation for data retrieval using the SMFUTIL Archive.

During SMF data retrieval via the ARCHIVE, ALTUNITS allows a local unit name (i.e. "CART") to be substituted for a UCB device type string that might otherwise cause allocation of devices that cannot actually mount the required dataset volume(s). An example of such a device would be a 4mm SCSI DAT tape drive being used to emulate an IBM 3490E type device on the same system where actual 3490E devices are used to store SMF data. The UCB device type string is identical for both of these devices so a unit name substitution is required to differentiate between them. For this to work correctly the substituted name ("CART" as shipped in the table or as shown in the example below) must be defined as a unit name for the real device addresses that are capable of reading the SMF datasets identified as input by the ARCHIVE search. If another unit name is already defined for this purpose in the installation, you can modify table TALT in the SMFUUNIT source code member to reflect the proper name and rerun the INSTALL job.

EXAMPLE: ALTUNITS(78048081-CART)

R-CODE: NO EFFECT.

6.2.7 ARCBACKUP

FORMAT:	ARCBACKUP(frequency[,sid,dsnprfx,unit,volser,spacetype,priqty,secqty,dclass,sclass,mclass])
frequency	The backup frequency specification. Valid values are “NEVER”, “ALWAYS”, “DAILY”, “SUNDAY”, “MONDAY”, “TUESDAY”, “WEDNESDAY”, “THURSDAY”, “FRIDAY” or “SATURDAY”. If omitted, the default will be used.
sid	The SMF system id of the system that is to perform the backup function. All other systems will bypass the backup function. If set to blanks, all systems will perform backups whenever an archive update is performed, as directed by the frequency specification. If omitted the default value from the SMFUDFLT table will be used.
dsnprfx	The data set name to be used for output of the backup function. If a GDG base name is specified, a relative generation number of “+1” will be automatically assigned, otherwise a suffix will be appended. If omitted, the default value will be used if a backup is performed. Blanks may be specified to override the default value. If it is specified as blanks, the ARCHIVEDB cluster name with the qualifier ‘.BACKUP’ appended to it will be used instead. The rules for GDG or date/time suffixing as defined for ARCBDSN still apply. Note that caution should be exercised with respect to the cluster name length so that sufficient room is available for the suffixes to be appended. As a rule, the following maximum lengths apply: DATE/TIME suffixing - ARCHIVDB max length is 20 BACKUP=ALLWAYS DATE suffixing - ARCHIVDB max length is 29 BACKUP=DAILY or greater GDG DSN structure - ARCHIVDB max length is 29
unit	Unit name to be used to allocate the backup data set. If omitted, the default value will be used.
volser	Specific volume that the backup data set is to be allocated on. If set to blanks, no specific volume will be requested. If omitted, the default value will be used.
spacetype	A keyword to indicate the space type to be allocated a DASD volume or that a tape device is in use. Valid values are “TRK”, “CYL” or “TAPE”. If the unit specified is a tape device, this parameter must be set to “TAPE”.. If omitted, the default value will be used.
priqty	The primary space quantity to be assigned. Valid values are 1 to 99 for a disk data set and 0 for a tape data set.. If omitted, the default value will be used.
secqty	The secondary space quantity to be assigned. Valid values are 1 to 99 for a disk data set and 0 for a tape data set. If omitted, the default value will be used.
dclass	The SMS Data Class to be used to allocate the backup data set. If blank, no Data Class will be used. If omitted, the default value, if any, will be used.
sclass	The SMS Storage Class to be used to allocate the backup data set. If blank, no Storage Class will be used. If omitted, the default value, if any, will be used.
mclass	The SMS Management Class to be used to allocate the backup data set. If blank, no Management Class will be used. If omitted, the default value, if any, will be used.
PARM-LIST:	REQUIRED, but all positional parameters within the list are optional. Only items required to override the defaults need to be specified. Specifying ARCBACKUP with no parameters within the parenthesis has no effect on the default settings defined during installation of the product.
FUNCTION:	To indicate when SMFUTIL is to automatically back up the ARCHIVE database. If the ARCHIVE database is not in use or if no updates have been applied, the backup function is bypassed. Note that a frequency specification of “NEVER” will disable the archive backup function. Any remaining parameters will be scanned for syntax but will have no effect as no backup will be done.

A specification of “ALWAYS” will cause a backup to be performed whenever updates have been applied to the archive database. A specification of “DAILY” or a specific day of the week will cause a backup to be performed only after the first archive update each day (DAILY) or on the day of the week specified for backup. The day of the week names may be abbreviated to the first 3 characters of the day name (i.e. “MON” may be used for “MONDAY”).

R-CODE: NO EFFECT.

6.2.8 ARCFORCE

FORMAT: ARCFORCE

INVERSE: NOARCFORCE

PARM-LIST: NOT ALLOWED

FUNCTION: To instruct SMFUTIL to force update the SMFUTIL ARCHIVE with ARCHIVE records that already exist at the time an ABEND occurs. This allows updating the ARCHIVE with available data and serves to shorten restarts if some split datasets are already complete and cataloged. Caution should be exercised when restarting a run to ensure that any checkpoint records are in the correct state. If an interrupted run that used ARCFORCE is rerun from the beginning, any ARCHIVE records previously recorded will be replaced (if output to the same volumes) or rendered logically deleted (if output is to new volumes and dataset is re-cataloged to the new volumes).

For output datasets, records will be written for volumes that have completed the output phase (i.e. hit an EOV or EOT mark and/or been closed). For input datasets being ARCHIVED via the ARCHINPUT command all volumes that have begun input, even partially completed ones, will be archived.

NOTE: The conditional disposition of output datasets should be CATLG to ensure they are retained, otherwise the archive record will be moot. If these datasets are recreated by a rerun of the job, the previous catalog entries should be remove prior to the rerun.

R-CODE: NO EFFECT.

6.2.9 ARCHBIAS

FORMAT: ARCHBIAS(biasdays)

biasdays An integer value from 1 to 99.

PARM-LIST: REQUIRED

FUNCTION: To supply a bias value for SMFUTIL archive searches. The bias value is the number of days that SMFUTIL backs up from the lowest required date to begin searching for matching records in the archive. A bias value for each set is normally computed automatically and stored in the archive. In the event the computed bias value is corrupted, wrong or otherwise not allowing all data to be extracted, the user may supply an override value via the ARCHBIAS keyword.

R-CODE: NO EFFECT.

6.2.10 ARCHINPT

FORMAT: ARCHINPT

PARM-LIST: NOT ALLOWED

FUNCTION: To instruct SMFUTIL that the dsname, volume, and date and time ranges of data read as input is be recorded to the SMFUTIL SYSARCH archive database. The SYSARCH DD statement must be available in the JCL or the ARCHIVDB parameter must be supplied to define the name of the database. Alternatively, the database name prefix may be permanently installed into the SMFUTIL defaults via an assembly and link of the defaults table. The 'ARCHINPT' keyword is mutually exclusive with VSAM input data sets. The 'NORMAL' set name is assigned to archived input data unless the 'ISETNAME' is used to assign a different name. See the section titled "ARCHIVING INPUT DATA" on page 181 for more information on this subject.

NOTE: The "SIMULATE" keyword must be specified if the data is not to be copied to an output data set and the "SYSUT2" output data set has been omitted. This will substantially speed up the run.

R-CODE: NO EFFECT.

6.2.11 ARCHIVDB

FORMAT: ARCHIVDB(cluster-dsname)
NOARCHDB

PARM-LIST: REQUIRED

FUNCTION: ARCHIVDB is used to supply the Archive database cluster to be used for ARCHIVE and data retrieval functions for this SMFUTIL execution. The name supplied must have been previously allocated via the ARCHDEFN job in the INSTLIB data set supplied on the SMFUTIL distribution tape. See the section titled "ARCHIVING SMF DATA" on page 175 for more information on this subject.

NOTE: If a database was defined in the SMFUDFLT defaults module during installation, the ARCHIVDB keyword will override it with the new name and the NOARCHDB keyword will disable the usage of the archive for the current execution.

NOTE: If ARCBDSN is specified as null in the DEFAULTS module or overridden with blanks via the ARCBACKUP keyword, the ARCHIVEDB cluster name with the qualifier '.BACKUP' appended to it will be used instead. The rules for GDG or date/time suffixing as defined for ARCBDSN still apply. Note that caution should be exercised with respect to the cluster name length so that sufficient room is available for the suffixes to be appended. As a rule, the following maximum lengths apply:

DATE/TIME suffixing	- ARCHIVDB max length is 20 BACKUP=ALLWAYS
DATE suffixing	- ARCHIVDB max length is 29 BACKUP=DAILY or greater
GDG DSN structure	- ARCHIVDB max length is 29

R-CODE: NO EFFECT.

6.2.12 ARCHIVE

FORMAT: ARCHIVE[(ddnmask,ddnmask,...)]

PARM-LIST: OPTIONAL

ddnmask A prefix mask that will match one or more standard output DD names.

FUNCTION: To instruct SMFUTIL that the date and time ranges of data placed to one or more standard output data sets is to be recorded to the SMFUTIL SYSARCH archive database. The SYSARCH DD statement must be available in the JCL or the ARCHIVDB parameter must be supplied to define the name of the database. Alternatively, the database name prefix may be permanently installed into the SMFUTIL defaults via an assembly and link of the defaults table. See the section titled “ARCHIVING SMF DATA” on page 175 for more information on this subject.

NOTE: If no parameter list is supplied, ALL standard output data sets (i.e. SYSUT2, SYSUT2D, etc.) will be archived. If a list of one or more prefix masks are specified, only DD names that match a mask will be archived. Caution should be exercised when specifying a mask. For example a specification of ARCHIVE(SYSUT2) would match both SYSUT2 and SYSUT2D (because they both start with SYSUT2) while ARCHIVE(SYSUT2) would match only SYSUT2 because of the blank after the mask. Note that asterisks (*) may be used to indicate any character is acceptable in a position.

R-CODE: NO EFFECT.

6.2.13 ARCHIVEACTIVE

FORMAT: ARCHIVEACTIVE

PARM-LIST: NOT ALLOWED

FUNCTION: To instruct SMFUTIL that data read from the active MANx dataset may be ARCHIVED. Normally, during a MANXALOC run, SMFUTIL will not accept input from the active MANx dataset even though ACTIVEOK is specified or defaulted. The ARCHIVEACTIVE specification allows this to be overridden for testing purposes only.

NOTE: ARVHIVEACTIVE is for testing purposes only and should never be specified in a production SMFUTIL execution. It is usually specified in conjunction with MANXARCH to simulate a DUMPCLEARALL run without actually clearing the MANx files.

R-CODE: Not affected.

6.2.14 ARCHIVERECOVER

FORMAT: ARCHIVERECOVERY

PARM-LIST: NOT ALLOWED

FUNCTION: To instruct SMFUTIL that the ARCHIVE database is be rebuild using the input SMF records of previous ARCHIVE updates. For ARCHIVERECOVER to be available the ARCHSMF parameter must be set to “YES” and the ARCHSMF# parameter must be set to a valid SMF record number in the SMFUDFLT module during installation or the ARCHSMFRECNUM keyword must have been previously specified.

NOTE: This function allows the reconstruction of the ARCHIVE database in the event it is corrupted or deleted. The database must be defined and initialized prior to use. No data movement is allowed (“SIMULATE” must be specified). Only the ARCHIVE update records are processed to produce updates to the SMFUTIL ARCHIVE database. See the section titled “RECOVERING ARCHIVE DATA” on page 179 for more information on rebuilding the Archive Database.

R-CODE: Not affected.

6.2.15 ARCHSID

FORMAT: ARCHSID

INVERSE: NOARCHSID

PARM-LIST: NOT ALLOWED

FUNCTION: To instruct SMFUTIL to use (ARCHSID) or disregard (NOARCHSID) in the searching of the ARCHIVE database for appropriate input data. If NOARCHSID is specified, any ARCHIVE records that meets the required input date/time specifications and set name criteria will be considered valid for usage without consideration of the ARCHIVE’s indication of the SID values present on the data set.

R-CODE: Not affected.

6.2.16 ARCHSMF

FORMAT: ARCHSMF
INVERSE: NOARCHSMF
PARM-LIST: NOT ALLOWED
FUNCTION: To instruct SMFUTIL that the updates to the ARCHIVE database are to be recorded in before and after images via SMF records of the type indicated by the ARCHSMF# value in the SMFUDFLT module or via a previous ARCHSMFRECNUM keyword in the same execution. Conversely, NOARCHSMF turns off the SMF recording function if it was defaulted to “YES” via the ARCHSMF parameter in the SMFUDFLT module.
R-CODE: Not affected.

6.2.17 ARCHSMFRECNUM

FORMAT: ARCHSMFRECNUM(rnum)
rnum An SMF record number type in the range of 129 to 255.
PARM-LIST: REQUIRED
FUNCTION: To instruct SMFUTIL to enable ARCHIVE update image recording to SMF records and to use the specified record number to record the before and after images. This value overrides the ARCHSMF# value in the SMFUDFLT module even if is zero (disabled).
NOTE The ARCHSMF keyword must also be specified (after this keyword) to request recording to begin.
R-CODE: Not affected.

6.2.18 ARCHSRCH

FORMAT: ARCHSRCH
PARM-LIST: NOT ALLOWED
FUNCTION: To instruct SMFUTIL that the ARCHIVE database is be searched for data to meet the date/time spans requested even though a user specified input data set was supplied (via JCL or DDA block). Note that any user supplied data set will be input first and any ARCHIVE supplied data set(s) input after the user supplied input.
NOTE: Use of the ARCHSRCH keyword and a user supplied input data set may result in duplicated data and/or erroneous early termination of the execution because the user data set is being read out of order with the rest of the data. Note that the ARCHIVE supplied data set(s) will be allocated AFTER the user supplied data set. This means the user supplied data set(s) are input first. If the user supplied data set(s) contains data outside the scope of the supplied explicit or implicit DATE/TIME range, it will be skipped. The GOTOEOF keyword should be specified to prevent early termination of the run in the event that the user supplied data set(s) contain data after the requested date range. A post process SORT step may then be necessary to ensure that the data is in the desired sequence.
R-CODE: Not affected.

6.2.19 ARCHIVEUSER

FORMAT: ARCHIVEUSERUSER[(ddnmask,ddnmask,...)]

PARAM-LIST: OPTIONAL

ddnmask A prefix mask that will match one or more standard output DD names.

FUNCTION: To instruct SMFUTIL that the date and time ranges of data placed to user defined output data sets (i.e. TYPETODD specifications) is to be recorded to the SMFUTIL SYSARCH archive database. The SYSARCH DD statement must be available in the JCL or the ARCHIVDB parameter must be supplied to define the name of the database. Alternatively, the database name prefix may be permanently installed into the SMFUTIL defaults via an assembly and link of the defaults table. Data placed to a user defined DD statement will be given an archive set name of the first five characters of the DDNAME unless the user defines the data set via a DDA block and also supplied an archive set name via the ASETNAME DDA block parameter. This serves to segregate the user data from the NORMAL and DUPLEX output to SYSUT2 and SYSUT2D respectively. See the section titled "ARCHIVING SMF DATA" on page 175 for more information on this naming archive sets.

NOTE: If no parmlist is supplied, ALL user defined output data sets (i.e. those defined via TYPETODD statements) will be archived. If a list of one or more DD nameprefix masks are specified, only DD names that match a mask will be archived. Caution should be exercised when specifying a mask. For example a specification of ARCHIVE(RMFDATA) would match both RMFDATA and RMFDATA3 (a split data set) because they both start with RMF, while ARCHIVE(RMFDATA) would match only RMFDATA because of the blank after the mask. Note that asterisks ('*') may be used to indicate any character is acceptable in a position.

R-CODE: NO EFFECT.

6.2.20 ASMFIDLN

FORMAT: ASMFIDLN(n)

PARAM-LIST: REQUIRED

n An integer value of 1 to 4.

FUNCTION: To indicate the number of characters of the SID value in the ARCHIVE record that is to be considered when validating a record for possible usage. The shipped default is 4 as defined in the SMFUDFLT defaults module.

R-CODE: NO EFFECT.

6.2.21 BADSNAPLIMIT

FORMAT: BADSNAPLIMIT(n)

PARAM-LIST: REQUIRED

n An 1 to 9 digit integer value.

FUNCTION: To place an upper limit the number of invalid records snapped out to print on the BADSNAP DD statement.

R-CODE: NO EFFECT.

6.2.22 BGUSER

FORMAT: BGUSER(userid[,userid,...])

userid A 1 to 8 character user identification string mask. If less than 8 characters are specified, all users that start with the specified string will be selected. To limit the selection to a specific userid that is less than 8 characters, it is necessary to pad the userid name to the right with blanks within the parenthesis. A wildcard specification of '*' (an asterisk) may be used in any position to specify that any value is to be accepted in that position.

PARM-LIST: At least 1 userid required.

FUNCTION: To restrict record selection to SMF records that pertain to the specified batch resource user(s). Records from the BATCH subset with the selected userid(s) will be selected. Only record types with userid capability are screened. Other record types that are otherwise selected will bypass userid validation and be processed.

EXAMPLE: BGUSER(MYID) would select all user id's that begin with "MYID".
 BGUSER(MYID) would select only records for which the user id was "MYID".
 BGUSER(**ID) would select only records for which the user id has the characters 'ID' in the third and fourth positions. This is equivalent to BGUSER(**ID****).

NOTE: See the section titled "USERID RECOGNITION" on page 151.

R-CODE: 0 Data for Batch user copied.
 4 No data found for User ID.

6.2.23 BIDBYHOUR

FORMAT: BIDBYHOUR

PARM-LIST: Not Allowed.

FUNCTION: To instruct SMFUTIL that all datasets being archived (input and output) are to have Block ID (BIDs) recorded on an hourly basis. Each BID slot will represent one hour of data.

NOTE: Each dataset must not have more than a single day of data and all the data must be from the same date in ascending order. If more than a single date is detected, hourly BIDs for that dataset will be suppressed and a per day value will be kept.
 BIDBYHOUR is valid for input and output datasets but has no effect unless archiving is actually done for the dataset.

NOTE: BIDBYHOUR should only be specified if all datasets being archived are to be hourly. If some datasets are to be daily (i.e. weekly and monthly outputs) BIDBYHOUR should not be used and DDABIDHR for each DDA block representing a daily type dataset should be specified instead.

R-CODE: Not affected

6.2.24 BIDINDEX

FORMAT: BIDPOINT(index)

index A integer value of 1 to 31.

PARM-LIST: REQUIRED

FUNCTION: To instruct SMFUTIL to use a specific Block ID Index value for this run during Archive updates. This index value represents the number of days of data to be recorded per block identification slot in the archive record. There are 31 block identification slots in each archive record. The default block ID index value of 1 allows each day of data on a volume to be individually addressable by block ID.

NOTE: Normally, the default value of BIDINDEX=1 should be retained. Change this value only if you are recording a large time span (i.e. more than 1 month) on a single volume.

R-CODE: Not affected.

6.2.25 BIDPOINT

FORMAT: BIDPOINT

INVERSE: NOBIDPOINT

PARM-LIST: NOT ALLOWED

FUNCTION: To instruct SMFUTIL to use (BIDPOINT) or ignore (NOBIDPOINT) block identification information stored in the archive record when positioning input cartridge volumes for input start.

NOTE: Normally, the default value of BIDPOINT=YES should be retained to allow SMFUTIL to do high-speed positioning of input cartridge data sets where possible. The only reason to override this with NOBIDPOINT is in the event that such positioning is causing errors. Please contact ASPG Technical Support in this situation.

R-CODE: Not affected.

6.2.26 BYTEDETAIL

FORMAT: BYTEDETAIL

INVERSE: NOBYTEDETAIL

PARM-LIST: NOT ALLOWED

FUNCTION: To instruct SMFUTIL to produce an additional report showing the number of bytes moved for each record type and in total.

R-CODE: Not affected.

6.2.27 CATINPUT

FORMAT:	CATINPUT(dsnprefix[[[SKIPLAST],[DELETE],[limit],[latest],[BIDH]]],...)
dsnprefix	A 1 to 35 character data set name prefix or GDG index. Multiple dsnprefix specifications are supported (each with the optional parameter).
SKIPLAST	An optional keyword to indicate not to process the last data set found under this prefix. If SKIPLAST is omitted and DELETE is specified, the ',' must be coded. If a 'limit' is specified without SKIPLAST and DELETE, the preceding ',' must be specified.
DELETE	An optional keyword to indicate data sets found under this prefix and used as input are to be deleted at successful completion of the run. The maximum return code allowing deletions is defined by the DELETERC default value.
limit	An optional integer value in the range of 1 to 255 to limit the number of data sets that may be selected for this dsn prefix.
latest	An optional integer value in the range of 1 to 255 to limit the dataset selection to this number of the "latest" (newest) data sets in the list.
BIDH	If coded, the BIDH keyword indicates that, during an ARCHIVEINPUT execution, datasets under the specified index are to have BID values recorded on an hourly rather than a daily basis. The dataset(s) must contain a single days (single date up to 24 hours total) worth of data for this to be effective else SMFUTIL will revert to daily BID values for the dataset.
FUNCTION:	<p>To instruct SMFUTIL that data sets in the system catalog under a specified data set name prefix are to be dynamically allocated and processed as input data sets in the order in which they appear in the catalog. If 'SKIPLAST' is specified, the last data set under the index level will be omitted from processing. If 'DELETE' is specified, all data sets that were allocated for the dsnprefix will be deleted at the end of the run. This deletion is done after all processing has completed. If the return code of the run is 8 or greater, the 'DELETE' function is bypassed.</p> <p>If the 'limit' parameter is specified, no more than the specified number of data sets will be selected for input processing for this prefix. All data sets in the catalog above this limit will be ignored during the current execution. Note that you may specify the 'DELETE' keyword with a limit value because delete only applies to data sets actually selected for input. This means that data sets NOT selected for input because of a user specified limit will NOT be subject to deletion.</p> <p>If the 'latest' parameter is specified, only the last number of datasets specified will be considered for input. This means if there are 10 datasets under the prefix specified and a 'latest' value of 3 is specified, only the top (last) 3 datasets on the list will be considered. Use the 'latest' parameter carefully, especially if the DELETE keyword is specified, as out of order data processing is likely.</p> <p>If 'limit' and 'latest' are both specified, 'limit' will only be applied to datasets eligible for selection after 'latest' has been applied. This effectively means that 'limit' can never be larger than 'latest'.</p> <p>Judicious usage of 'limit' and 'latest' can be utilized to selectively process portions of a catalog list.</p>
NOTE:	If 'DELETE' is specified, only datasets actually selected for input processing will be eligible for DELETE processing. Datasets skipped because of SKIPLAST, 'limit' or 'latest' specifications will not be deleted.
CAUTION:	If a user exit issues a return code terminating the run the resulting return code for SMFUTIL will be 4. If the 'DELETE' keyword is specified, this will result in deletion of the input datasets even if all of them have not been read. Use caution in implementing user exits when the 'DELETE' keyword is to be utilized.
R-CODE:	Not directly affected but if no input data sets are found a 12 will result.

6.2.28 CHKISORT

FORMAT: CHKISORT

PARM-LIST: NOT ALLOWED

FUNCTION: Checks the input data for sorted ordering by date and time.

NOTE: Only record types eligible for output processing will be checked for input sorted ordering. Both CHKISORT and CHKOSORT may be specified in the same run. An indication will be given as to the sort condition of both input and output. If either is out of sort, a return code of 4 will be issued for the run. See the section titled “SMF DATA ORDER CHECKING” on page 160 for more information about sorted ordering in SMF data.

R-CODE: 0 If input data is sorted.
4 If input data is not sorted.

6.2.29 CHKOSORT

FORMAT: CHKOSORT

PARM-LIST: NOT ALLOWED

FUNCTION: Checks the output data for sorted ordering by date and time.

NOTE: Both CHKISORT and CHKOSORT may be specified in the same run. An indication will be given as to the sort condition of both input and output. If either is out of sort, a return code of 4 will be issued for the run. See the section titled “SMF DATA ORDER CHECKING” on page 160 for more information about sorted ordering in SMF data.

R-CODE: 0 Output data is sorted.
4 Output data is not sorted.

6.2.30 CICSSPLIT

FORMAT: CICSSPLIT(applid-ddname[(subtype,class,ver)])

applid A CICS application identifier or mask to match one or more CICS regions. This is also known as the CICS Product Name.

ddname An output file name the selected records are to be targeted to.

subtype An optional one digit subtype specification to limit record selection. If omitted, all subtypes are selected. Currently meaningful values are '0' for journaling records, '1' for monitoring data, and '2' for statistics records.

class An optional one digit data class indicator to limit data selection. Currently '1' is dictionary, '2' is accounting, '3' is performance, and '4' is exception data.

ver An optional one to three character filter to limit data selections to data records produced by a specific version of CICS.

PARM-LIST: At least one applid-ddname pair required. Multiple applid-ddname pairs may be specified and multiple CICSSPLIT keywords may be entered.

FUNCTION: Instructs SMFUTIL to split CICS 110 records by APPLID and place the data on the indicated DDNAME file. The APPLID specification may be a prefix or mask. If an '*' appears in the APPLID specification, any character in that position in the record will be accepted. If less than 8 characters are specified, the specification will be considered a prefix and all CICS regions with an APPLID that match the prefix will be targeted to the associated DDNAME. To limit selection to a CICS region with an APPLID of less than 8 characters place at least one space between the APPLID and the '-ddname' string.

EXAMPLE: CICSSPLIT(CSHORT -MYDDNAME)

The offset in the record to the applid is located by a record version recognition scan using the SMFCICTB table module. This offset is assumed to contain the applid for the CICS started task that created the type 110 record. The record is scanned for CICS version recognition and the resulting offset is used. If the content of the field in the record matches the specified 'applid' string, the record will be written to file associated with the specified DDNAME. If the applid string is directly followed by the '-' separator character, it will be assumed that the applid specification is a prefix. Any record containing an applid that matches the specification for its length will be considered selected. If a blank follows the applid string (between it and the '-') it will force the applid specification to be a full match for the applid in each record. This syntax structure allows one specification to select multiple applid's for selection to one output file.

The optional filtering sub-parameters 'subtype', 'class', and 'ver' may follow the DD name field, enclosed in parenthesis. Any omitted field must have the intervening comma coded. These fields serve to limit data selection to specific data criteria. The valid subtype values are '0', '1' and '2'. The valid class values are 1 through 4. The valid CICS version specifications are found in the SMFCICTB table.

Note that '1', '2', '170' and '210' will all select all CICS versions for '210' and below because there is no differentiation possible for data records below version 3.0 as no identification of the producing version is available in the record. Note also that version 3.0 and up are considered ESA type data and 2.1 and down is considered MVS type data.

A 'ver' specification of '2' will select 210 and below. A 'ver' specification of '3' will select versions 300 through 321. A specification of '4' will select 400 and up.

NOTE: Multiple specifications are allowed to the same DD name. For example:

```
CICSSPLIT(*-CICSDATA(, , 3))
CICSSPLIT(*-CICSDATA(, , 4))
```

would select all CICS version 3 and 4 data.

NOTES: If a type 110 record is found with an applid not specified in a CICSSPLIT statement, it will be written to the standard output data sets. By default, type 110 records written to a user specified DDNAME will not be written to the standard output files. The keyword COPYVBTH may be specified to cause the selected 110 records to be written to both the user specified DDNAME and to the standard output data sets.

Unless specific record types have already been selected, either explicitly or implicitly, when a CICSSPLIT function is encountered, all record types will be turned off and only type 110 records will be turned on for selection. This can be negated by a specification of 'ALL' after the CICSSPLIT keyword.

If VSINGLE is specified (or defaulted), CICS records will be written to the **first** output DDNAME that matches the specified APPLID or mask. If VMULTI is specified (the default as shipped) CICS records will be written to **ALL** output targets that match the requested APPLID or mask.

If multiple CICS's share a common prefix in their applid, caution should be exercised in the specification of the applid filters or unwanted or duplicated data may result. For example if "CICS", "CICSPROD" and "CICSTEST" were valid CICS applid's and:

```
CICSSPLIT(CICS-DDPROD)
CICSSPLIT(CICSTEST-DDTEST)
CICSSPLIT(CICSPROD-DDPROD)
```

were coded, DDNAME DDPROD would receive all three sets of records and CICSPROD would be duplicated because it matches both "CICS" and "CICSPROD". This should be coded:

```
CICSSPLIT(CICS -DDPROD)
CICSSPLIT(CICSTEST-DDTEST)
CICSSPLIT(CICSPROD-DDPROD)
```

with an extra space after the "CICS" applid in the first **CICSSPLIT** keyword. This would target only the CICS named specifically "CICS" and achieve the desired result.

R-CODE: 0 CICS data split as requested.
8 No type 110 records found to match criteria.

6.2.31 CICSSUBC

FORMAT: CICSSUBC

PARM-LIST: NOT ALLOWED

FUNCTION: Instructs SMFUTIL to use the data class field in the CICS 110 record for record subtype discrimination, where required. The default subtype definition, as defined in the SMFCICTB CICS table module, differentiates between true subtypes. Currently meaningful subtype values are '0' for journaling records, '1' for monitoring data, and '2' for statistics records.

If CICSSUBC is specified, the class field from the first product section in the record will be used, instead of subtype, to limit data selection. Currently '1' is dictionary, '2' is accounting, '3' is performance, and '4' is exception data.

NOTE: Alternatively, the **CICSSPLIT** keyword can be used to copy CICS records with both specific subtype and class values. In addition, the producing CICS version may be limited as well. This allows CICS data to be finely split into multiple subsets as required.

R-CODE: Not affected.

6.2.32 CICSTYPE

FORMAT:	CICSTYPE([applid],[subtype],[class],[ver],...)
applid	An optional CICS application identifier or mask to match one or more CICS regions. This is also known as the CICS Product Name. Masking characters may be used to select more than one applid.
subtype	An optional one digit subtype specification to limit record selection. If omitted, all subtypes are selected. Currently meaningful values are '0' for journaling records, '1' for monitoring data, and '2' for statistics records.
class	An optional one digit data class indicator to limit data selection. Currently '1' is dictionary, '2' is accounting, '3' is performance, and '4' is exception data.
ver	An optional one to three character indicator to limit data selections to data records produced by a specific version of CICS. See the CICSSPLIT keyword for more information on CICS version specifications.
PARM-LIST:	Required but each positional is optional. More than one set of parameters may be specified.
FUNCTION:	Instructs SMFUTIL to select only CICS type records (type 110) and to limit the records selected to those meeting the specified criteria. The 'ver' specification is used to select the CICS version that data is desired for. The supported identifiers are 'MVS', 'ESA', '31', '32' and '33'. 'MVS' denotes all version 1 and version 2 levels of CICS releases. 'ESA' denotes all version 3 levels and up. Specific version 3 levels may be selected by '31', '32', and '33'.
NOTE:	The CICSTYPE parameter causes all record types other than 110 to be turned off. Only CICS data selected will be passed to the output files unless other record types are turned back on (explicitly or implicitly) after the CICSTYPE parameter in the SYSIN control stream.
R-CODE:	0 - Data selected 8 - No data found

6.2.33 CKPTDB

FORMAT:	CKPTDB(ckpt-dsn)
ckpt-dsn	The name of the checkpoint PDS data set to allocate to the SYSCKPT DD name.
PARM-LIST:	REQUIRED
FUNCTION:	To request that the checkpoint PDS data set be dynamically allocated. The data set name supplied will be allocated to the SYSCKPT file name during initialization. SYSCKPT must not be present in the JCL. If the supplied data set name contains the character string '&SID', it will be replaced with the SMF system identifier on which the job is executing. This is useful for defining a unique checkpoint data set for each system to be processed and still maintain a single set of control cards and JCL procedures.
NOTE:	If the ckpt-dsn specified does not currently exist and the DACKPT parameter in the SMFUDFLT module is set to 'YES', SMFUTIL will dynamically create the data set using the default values specified. As shipped this is a 1 track PDS with 40 directory blocks. The unit value is 'SYSDA' and no specific volume is specified.
R-CODE:	Not affected.

6.2.34 CKPTINPT

FORMAT: CKPTINPT

INVERSE: NOCKPTIN

PARM-LIST: NOT ALLOWED

FUNCTION: Instructs SMFUTIL that first input record is to be check pointed. During each execution, SMFUTIL will check the previous input record checkpoint (if one exists) against the first record of the current execution. If they match, SMFUTIL will report that duplicate input data has been detected and the run will be aborted.

NOTE: This feature is only active if the SYSCKPT DD statement is present in the JCL. If it is absent, CKPTINPT will have no affect on the run.

CKPTINPT is set to YES in the SMFUDFLT module as shipped. To temporarily disable this feature, use the NOCKPTIN keyword during execution. To permanently disable this feature by default, change the SMFUDFLT module to specify CKPTINPT=NO and reassemble the module. The CKPTINPT keyword can be then be used on a selective basis to temporarily enable the feature.

R-CODE: 0 Input data is different from last run.
8 Duplicate input data detected.

6.2.35 CLEAR

FORMAT: CLEAR

PARM-LIST: NOT ALLOWED

FUNCTION: Instructs SMFUTIL to clear the input SMF VSAM data set of all data. This is to be executed in a separate step after a successful copy step. SMFUTIL must be executed from an authorized library for SMF to be notified that the data set has been cleared. In addition, if a security system is in use, SMFUTIL must have write access to the SMF data set.

NOTE: The CLEAR function only will be executed. No data movement will be performed. SYSUT2 may not be present in the JCL for this step. If any other input commands are detected, execution will be aborted. By its nature, the CLEAR function destroys the data in the input (SYSUT1) data set. If data movement was done in the same step as a CLEAR function and the system failed during the CLEAR execution, the output data set(s) would be in jeopardy and the input data would no longer be available. For this reason, CLEAR must be performed in a separate step. Return codes should be carefully checked to insure that the copy step completed successfully. One added benefit of this limitation is that the output data sets are available for processing immediately, without waiting for the CLEAR to complete. Additionally, the CLEAR function is ignored if the input data set is not a VSAM format SMF data set. SMF data sets will be cleared and returned for reuse by the operating system if they are currently known to the SMF system of the operating system SMFUTIL is being run on. If the data set is not known to SMF, it will be cleared but not used by the SMF system. Shops that use multiple operating systems should exercise caution when dumping and clearing 'foreign' data sets on a system. The data set will be cleared using the format of the operating system upon which SMFUTIL is run. MVS/XA SMF data sets have a different clearing format than MVS/SP SMF data sets. Clearing SMF data sets should normally only be done by the system on which they are in use. Caution should be exercised when using the clear function to make certain that required data is was copied before the clear. Normally, CLEAR will only be used by a job (or started task) that moves all of the data to one or more archive files first.

R-CODE: 0 SMF data set cleared for reuse.
8 Clear function failed.

6.2.36 COPY0OK

FORMAT: COPY0OK

INVERSE: NOCOPY0

PARM-LIST: NOT ALLOWED

FUNCTION: Instructs SMFUTIL that a data copy execution that results in no data being copied (i.e. no data records were written to an output data set) is acceptable and a return code greater than zero is not to be set because of the lack of output data.

NOTE: COPY0OK is useful during production DUMP and CLEAR jobs for SMF data sets as a data set may occasionally be dumped that contains no data. A return code of 8 would appear to be an error when in reality it is not. Specifying COPY0OK will prevent the return code and allow the job to continue.

R-CODE: Prevents 8 from no data.

6.2.37 COPYBOTH

FORMAT: COPYBOTH[(rrr[,rrr-rrr,rrr])]

INVERSE: NOCOPYBOTH

PARM-LIST: OPTIONAL

rrr An integer record type specification in the range of 0 to 255.

FUNCTION: Instructs SMFUTIL that records directed to specific DDNAMEs via the TYPETODD command are to be copied both to the specified DDNAME and to the standard output data set(s) (SYSUT2, SYSUT3, SYSUT2D, or SYSUT3D). If a parmlist is not supplied, all record types are copied to both the user defined targets and to the standard outputs (SYSUT2). If a parmlist is supplied, it consists of one or more integer specifications of specific record types that are to be copied to both locations. Single record types may be specified or a record range consisting of a starting and ending record type separated by a dash.

NOTE: COPYBOTH is useful for creating complete archive data sets of all or selected record types and special purpose data sets of specific record types. This is of great benefit in reducing the amount of data passing done by subsequent users of the data. For example the RMF record types (70-79) could be directed to a separate disk data set for processing by a daily performance report job and still be copied to the archive tapes for permanent retention. The parmlist gives you the ability to limit the duplication of record types on the standard outputs to only selected record types that must be in both places and still leave other types only in the subset data set (i.e. CICS 110 records on their on data set).

NOTE: COPYBOTH differs from COPYVALUEBOTH in that COPYBOTH pertains to records directed to specific DD names because of their specific record type. COPYVALUEBOTH pertains to records directed to specific DD names because of value specifications such as CICS applid's or SMF SID values.

R-CODE: Not affected.

6.2.38 COPYVALUEBOTH

FORMAT: COPYVALUEBOTH[(rrr[,rrr-rrr,rrr])]

INVERSE: NOCOPYVALUEBOTH

PARAM-LIST: OPTIONAL

rrr An integer record type specification in the range of 0 to 255.

FUNCTION: Instructs SMFUTIL that records directed to specific DDNAMEs via the **SIDSPLIT**, **DB2SPLIT**, **MQSPLIT** or the **CICSSPLIT** keyword are to be copied both to the specified DDNAME and to the standard output data set(s) (SYSUT2, SYSUT3, SYSUT2D, or SYSUT3D). If a parmlist is not supplied, all record types are copied to both the user defined targets and to the standard outputs (SYSUT2). If a parmlist is supplied, it consists of one or more integer specifications of specific record types that are to be copied to both locations. Single record types may be specified or a record range consisting of a starting and ending record type separated by a dash.

NOTE: COPYVALUEBOTH is useful for creating complete archive data sets of all record types and special purpose data sets of specific SMF system ID records or CICS records for particular applid's. This is of great benefit in reducing the amount of data passing done by subsequent users of the data. The parmlist gives you the ability to limit the duplication of record types on the standard outputs to only selected record types that must be in both places and still leave other types only in the subset data set (i.e. CICS 110 records on their on data sets by SID).

NOTE: For record types other than 110 (CICS) to be processed during a '**CICSSPLIT**' execution, additional keywords such as 'ALL' or 'INCLUDE' or other keyword that select other record types must be specified after the '**CICSSPLIT**' keyword to turn record types on because **CICSSPLIT** will turn off all record types except 110.

NOTE: COPYVALUEBOTH differs from COPYBOTH in that COPYVALUEBOTH pertains to records directed to specific DD names because of value specifications such as CICS applid's or SMF SID values. COPYBOTH pertains to records directed to specific DD names because of their specific record type.

R-CODE: Not affected.

6.2.39 CREATES0

FORMAT:	CREATES0
PARM-LIST:	NOT ALLOWED
FUNCTION:	Causes the “normal” output SYSUT2 and SYSUT2D (if present) data sets to be created even if no data is to be placed in them. Output data sets are normally not opened until they have output data. This parameter will cause the data set(s) to be open at program initialization. This parameter is not normally required because unique return codes define when the SYSUT2 and SYSUT3 data sets have data. Subsequent job steps should not attempt to process these data sets unless a return code indicates they contain data or CREATES0 must be specified to cause empty data sets to be produced.
NOTE:	CREATES0 may be overridden for a specific dynamically allocated dataset by specifying the DDNOCREATE keyword in the DDA block.
NOTE:	If a run has a return code of greater than 8 (such as a syntax error) no creation of empty datasets will be performed.
CAUTION:	If a run has no input data due to an ILIMIT(0) specification or an empty input dataset, any data set to be created as empty that contains a suffix or a date overlay in the data set name will have the current execution date used for that purpose.
R-CODE:	Not affected.

6.2.40 CREATEU0

FORMAT:	CREATEU0
PARM-LIST:	NOT ALLOWED
FUNCTION:	Causes the user defined output data set(s) specified in the TYPETODD command to be created even if no data is to be placed in them. Output data sets are normally not opened until they have output data. This parameter will cause the data set(s) to be open at program initialization. This parameter is normally required if subsequent job steps are to process data sets created even if they have no data. Subsequent job steps should not attempt to process these data sets unless a return code of 0 indicates they contain data or CREATEU0 must be specified to cause empty data sets to be produced. This is useful if more than one data set is being created as the return code would be set if any of the data sets had no data.
NOTE:	CREATEU0 may be overridden for a specific dynamically allocated dataset by specifying the DDNOCREATE keyword in the DDA block.
NOTE:	If a run has a return code of greater than 8 (such as a syntax error) no creation of empty datasets will be performed.
CAUTION:	If a run has no input data due to an ILIMIT(0) specification or an empty input dataset, any data set to be created as empty that contains a suffix or a date overlay in the data set name will have the current execution date used for that purpose.
R-CODE:	Not affected.

6.2.41 CURRENTMONTH

FORMAT: CURRENTMONTH

ALIAS: THISMONTH, THISMNTN, or CURRMNTN

PARAM-LIST: NOT ALLOWED

FUNCTION: Causes all available data for the current month to be selected.

EXAMPLE: If CURRENTMONTH had been specified in a run executed on Friday, December 27, 2002, the dates selected would have been Sunday, December 1, 2002 through the current date (12/27/2002).

NOTE: When CURRENTMONTH is specified, an automatic default specification of SHORTCPY is assumed. The GOTOEOF or NOSTOP keyword may be used, after the CURRENTMONTH keyword, to override this default.

R-CODE: 0 Data found for current month.
8 No data found for current month.

6.2.42 CURRENTWEEK

FORMAT: CURRENTWEEK

ALIAS: THISWEEK

PARAM-LIST: NOT ALLOWED

FUNCTION: Causes data to be selected from the preceding Saturday to the current date.

EXAMPLE: If CURRENTWEEK had been specified in a run executed on Friday, December 27, 2002, the dates selected would have been Saturday, December 21, 2002, through the current date (12/27/2002).

NOTE: When CURRENTWEEK is specified, an automatic default specification of SHORTCPY is assumed. The GOTOEOF or NOSTOP keyword may be used, after the CURRENTWEEK keyword, to override this default.

R-CODE: 0 Data found for current week.
4 No data found for current week.

6.2.43 DARETRY

FORMAT:	DARETRY([(count)[,time]])
PARM-LIST:	OPTIONAL
count	An integer in the range of 1 to 9999 that specifies the number of retries to be attempted for a data set.
time	An integer in the range of 1 to 9999 that specifies the number of seconds to wait between retries.
FUNCTION:	To instruct SMFUTIL that all DDA block allocations are to be retried if they fail allocation for unavailable dsname, volume, or unit. If the 'count' and/or 'time' are omitted, 'count' will default to 100 and 'time' to 15 seconds. When specified, if the allocation fails for one of the recoverable error codes, SMFUTIL will wait the time interval specified by 'time' and then retry the allocation. This will continue for 'count' times. If both DARETRY (global DDA retry) and the DDA block keyword RETRY are specified, the DDA block RETRY specifications (or defaults) will be taken.
R-CODE:	Not affected

6.2.44 DATASET

FORMAT:	DATA SET(dsname[,dsname,...])
dsname	A 1 to 44 character data set name or data set name prefix for which records are to be selected. If less than 44 characters are specified, all data sets that start with the specified string will be selected. To limit the selection to a specific data set that is less than 44 characters, it is necessary to pad the data set name to the right with at least one blank within the parenthesis. A wildcard specification of '*' (an asterisk) may be used in any position to specify that any value is to be accepted in that position.
PARM-LIST:	At least 1 data set name specification required.
FUNCTION:	To cause all records that contain data set name fields to be tested for a specific data set name or data set name prefix. Those that match will be selected. All record types which are defined as in the data set name subtype (see "MODIFICATION AND CUSTOMIZATION" on page .199 for more information) are turned on for selection. Only those records that match the data set specification are selected for output. The data set specification is length sensitive. If the data set specification is less than 44 characters in length, it will be used as a mask to select all data sets that begin with the specification. If a data set that is less than 44 characters is desired only, it must be padded to the right with at least one blank. Other records that do not have data set names embedded in them may be specifically selected and will bypass data set name validation and will be put to the output data set(s). Multiple DATA SET parameters may be coded.
R-CODE:	0 Data was written to output. 4 No data output. Data set name not found in any input record.

6.2.45 DATE

FORMAT: DATE(ccyyddd[:ccyyddd][,ccyyddd:ccyyddd])

ccyyddd Julian date(s).

PARM-LIST: At least one date required.

FUNCTION: To instruct SMFUTIL that only data on a particular date or within a particular date range is eligible to be selected. All input data records with dates outside the specified date or date ranges will be discarded. If only one date is specified, only data for that day will be eligible for selection. Multiple start:end pairs may be coded.

If the input data set (SYSUT1xx) is not provided in the JCL or via a DDA block definition and the SYSARCH archive database is available to the execution, SMFUTIL will attempt to locate the required data via the archives.

NOTE: When DATE is specified, an automatic default specification of SHORTCPY is assumed. The GOTOEOF or NOSTOP keyword may be used, after the DATE keyword, to override this default.

NOTE: DATE is mutually exclusive with the DTRANGE keyword.

R-CODE:

0	Data found for date (range).
4	No data found for date (range).

6.2.46 DATESCAN

FORMAT: DATESCAN([(minute-delta],[ALLWEEKDAYS)])

minute-delta The number of minutes of time span between records that triggers a DATESCAN report entry.

PARM-LIST: OPTIONAL

FUNCTION: Instructs SMFUTIL to inspect the input data stream for missing time spans of data. SMFUTIL will default to a time span of 5 minutes before it reports a gap in the input data. The user may optionally specify a time span of up to 1439 minutes. SMFUTIL will normally require the output data to be in sorted format in order to check for missing data. The user may specify the FORCE keyword to instruct SMFUTIL to continue checking for data gaps even if it detects out of sort data. See the section titled “SMF DATA ORDER CHECKING” on page 160 for further information on sorted data validation and tolerances.

By default, DATESCAN scans for missing date/time spans in all input data. Optionally, the keyword WEEKDAYS, placed after the minute delta parameter, may be used to indicate that only days during the week (i.e. Monday through Friday) are to be considered for scanning.

R-CODE:

0	If no missing data detected.
4	If missing data areas found or data is out of sorted order.

6.2.47 DB2SPLIT

FORMAT: DB2SPLIT(ssid-ddname[,ssid-ddname,...])

ssid A 1 to 4 character subsystem identifier (or prefix) for a DB2 region.

ddname An output file name the selected records are to be targeted to.

FUNCTION: Instructs SMFUTIL to split DB2 SMF records (types 100-102) by Subsystem Id (ssid) and place the data on the indicated DDNAME file. The 'VALUE' field offset is located by a record version recognition scan using the SMFDB2TB module for the associated records types (100-102). This location is assumed to contain the ssid for the DB2 started task that created the record. If the content of the value field in the record matches the specified 'ssid' string, the record will be written to file associated with the specified DDNAME. If the ssid string is directly followed by the '-' separator character, it will be assumed that the ssid specification is a prefix mask. Any record containing an ssid that matches the specification for it's length will be considered selected. If a blank follows the applid string (between it and the '-') it will force the ssid specification to be a full match for the ssid in each record. This syntax structure allows one specification to select multiple SID's for selection to one output file.

NOTE: If a DB2 record is found with an ssid not specified in a DB2SPLIT statement, it will be written to the standard output data sets. By default, type DB2 records written to user specified DDNAMEs will not be written to the standard output files. The keyword COPYVALUEBOTH may be specified to cause the selected 110 records to be written to both the user specified DDNAMEs and to the standard output data sets.

Unless specific record types have already been selected, either explicitly or implicitly, when a DB2SPLIT function is encountered, all record types will be turned off and only type 100 through 102 records will be turned on for selection. This can be negated by a specification of 'ALL' after the DB2SPLIT keyword.

R-CODE:

0	DB2 data split as requested.
8	No DB2 records found.

6.2.48 DCACHECKPOINT

FORMAT:	DCACHECKPOINT([dsnprefix][,unit][,volser][,priqty][,dataclas][,mgmtclas][,storclas])
dsnprefix	A 1 to 36 character data set name prefix to be used to allocate the DUMPCLEARALL Checkpoint data set. This will be suffixed with the string “.SYSsyid” where ‘syid’ is the SMF system identifier of the executing system. Note that the prefix must be a valid data set name structure.
unit	A 1 to 8 character DASD type unit name value to be used to allocate the DUMPCLEARALL checkpoint data set.
volser	A 6 character volume serial of the device to be used to receive the DUMPCLEARALL checkpoint data set.
priqty	The primary space quantity for the DUMPCLEARALL Checkpoint data set, in tracks. The valid range is 1 to 255. The default is 1.
dataclas	A defined SMS Data Class to be used for the allocation of this data set.
mgmtclas	A defined SMS Management Class to be used for the allocation of this data set.
storclas	A defined SMS Storage Class to be used for the allocation of this data set.
PARM-LIST:	REQUIRED but each position is optional.
FUNCTION:	Instructs SMFUTIL to override the DUMPCLEARALL Checkpoint data set allocation specifications in the SMFUDFLT module with the supplied values. Each specification is positional and each may be omitted by specifying the intervening comma.
NOTE:	Normally, the DUMPCLEARALL Checkpoint data set allocation specifications should be defined in the SMFUDFLT module during installation and not altered thereafter. The DCACHECKPOINT keyword is provided for execution time override in the event a specific volume or device unit name specified in the defaults is not available and for sites that wish not to alter the shipped defaults for standards purposes.
CAUTION:	IF one of the SMS class parameters is coded, ensure that SMFUTIL will be able to delete the data set at the end of a successful execution.
CAUTION:	Before specifying this keyword for the first time, make sure that a current DUMPCLEARALL execution has not ended abnormally leaving an existing DCA Checkpoint data set waiting to be restarted.
R-CODE:	Not affected.

6.2.49 DCBABENDEXIT

FORMAT:	DCBABENDEXIT
INVERSE:	NODCBABENDEXIT
PARM-LIST:	NOT ALLOWED
ABBREVIATIONS:	DCBABXIT and NODCBABXIT
FUNCTION:	Instructs SMFUTIL to set (DCBABENDEXIT) or not set (NODCBABENDEXIT) a DCB exit to suppress certain DCB related ABEND's that can be safely ignored. One example would be a block count mismatch in the trailer label that would otherwise cause a 237-0C ABEND.
R-CODE:	Not affected.

6.2.50 DCBRESET

FORMAT: DCBRESET
INVERSE: NODCBRESET
PARM-LIST: NOT ALLOWED
FUNCTION: Instructs SMFUTIL to reset the DCB characteristics of OLD and SHR data set when a DCB is not explicitly specified in the JCL. The reset of the LRECL/BLKSIZE/RECFM is identical to that done for new data sets with no DCB specified. This allows an existing data set to be change to an optimum DCB without deleting and reallocating the data set. As shipped, SMFUTIL defaults to DCBRESET in the SMFUDFLT module. This default may be changed if required. A temporary change for a single execution may be invoked by specifying the inverse form (NODCBRST). Note that existing data sets with a disposition of MOD will not be reset regardless of the setting of this parameter.
R-CODE: Not affected.

6.2.51 DDEXIT

FORMAT: DDEXIT(exit-name)
exit-name A 1 to 8 character name of a user supplied load module exit. The load module must be available in the LINKLST or on LPALIB, JOBLIB, or STEPLIB.
PARM-LIST: A load module name is required.
FUNCTION: Instructs SMFUTIL to take a user supplied exit routine for each record to be deleted as a result of the DELDUP function. Upon entry to a user exit module, register 1 points to the record to be deleted.
 Upon return to SMFUTIL register 15 is tested for a return code. The following return codes are valid:
 0 Proceed with deletion of record.
 4 Override the DELDUP decision to delete this record and retain it instead.
R-CODE: Not affected.

6.2.52 DDSIDKEEP

FORMAT: DDSIDKEEP(ddname-smfid[,ddname-smfid,...])

ddname A valid output DD name.

smfid An SMF System ID that is acceptable for output to this ddname

PARM-LIST: A value pair must be provided.

FUNCTION: Instructs SMFUTIL to allow a specific system ID to be written to the specified output DD name. During output of specific record to a specific DD, the DDSIDKEEP specifications are scanned. If the current output DDNAME is not found in any DDSIDKEEP specification, the record is output to the DD. This is considered an unfiltered DD name. If the DDNAME is found the search continues to the end of the DDSIDKEEP specification chain. If an appropriate DDNAME and SID specification combination are not found, the record placement to the DD is suppressed and processing continues normally with the next output DD.

NOTES: The DDSIDKEEP specification must come after any DDA block definitions of DD names to be specified in the DDSIDKEEP syntax.

The same DD namemay be matched with more than one smfid's in multiple DDSIDKEEP specifications, as required by the user.

The same smfid may be matched with any number of ddname's in multiple DDSIDKEEP specifications, as required by the user.

R-CODE: Not affected.

6.2.53 DDSIDSKIP

FORMAT: DDSIDSKIP(ddname-smfid[,ddname-smfid,...])

ddname A valid output DD name.

smfid An SMF System ID that is not acceptable for output to this ddname

PARM-LIST: A value pair must be provided.

FUNCTION: Instructs SMFUTIL to prevent a specific system ID from being written to the specified output DD name. During output of specific record to a specific DD, the DDSIDSKIP specifications are scanned. If the current output DDNAME is not found in any DDSIDSKIP specification, the record is output to the DD. This is considered an unfiltered DD name. If the DDNAME is found but no matching SID is ever found the record is output to the DD. If the DDNAME is found with a matching SID, the record placement to the DD is suppressed and processing continues with the next output DD.

NOTE: The DDSIDSKIP specification must come after any DDA block definitions of DD names to be specified in the DDSIDSKIP syntax.

The same DD namemay be matched with more than one smfid's in multiple DDSIDSKIP specifications, as required by the user.

The same smfid may be matched with any number of ddname's in multiple DDSIDSKIP specifications, as required by the user.

R-CODE: Not affected.

6.2.54 DDRETAIN

FORMAT: DDRETAIN(count)

count An integer in the range of 1 to 99.

PARM-LIST: A value must be provided.

FUNCTION: Instructs SMFUTIL to set the depth of the retained record list to the specified value during the DELDUP function. If DELDUP is not also specified, DDRETAIN has no effect. If DDRETAIN is not specified during a DELDUP function execution, the default value from the site defaults table will be used (10 as shipped).

Each record retained requires a 32K buffer and causes additional overhead for record comparison. Use caution in setting values for DDRETAIN in excess of 10 as a large region size and degraded performance will result. Smaller values should be used (i.e. 1 or 2) when executing DELDUP on large files to enhance performance in trade for reduced dependability of the DELDUP function if the input data is not perfectly sorted..

R-CODE: Not affected.

6.2.55 DECR

FORMAT: DECR(integer)

integer Number of days to decrement from current date.

PARM-LIST: REQUIRED

FUNCTION: Instructs SMFUTIL to copy data beginning with the specified number of days before the current date. A specification of DECR(0) copies only the current day's data. A specification of DECR(1) copies all of yesterday and all of today. The number of days is limited to 365.

NOTE: When DECR is specified, an automatic default specification of SHORTCPY is assumed. The GOTOEOF or NOSTOP keyword may be used, after the DECR keyword, to override this default.

R-CODE:

0	Data copied
4	No data found within date range.

6.2.56 DELDUP

FORMAT: DELDUP

PARM-LIST: NOT ALLOWED

FUNCTION: Instructs SMFUTIL to delete duplicate records. If input data is PERFECTLY sorted, any duplicate records found will be deleted. If input data is not sorted, DELDUP will be suspended.

If a '//DUPSNAP DD SYSOUT=A' DD card is present in the JCL during execution of a DELDUP function, all duplicate records deleted will be snapped to print on it when they are deleted.

If a '//DUPSAVE DD DSN=???' DD card for an output data set is present in the JCL during execution of a DELDUP function, all duplicate records deleted will be written on the data set when they are deleted.

NOTE: DELDUP may not find all duplicate records unless the input data is in perfect sorted order. Normally, SMF data will require sorting prior to executing SMFUTIL with DELDUP. The purpose of DELDUP is to eliminate duplicate records on sorted SMF data sets when the duplicates were originally caused by improper dumping procedures. If the data set has not been sorted already, consider using the DELDUPRV function instead. It's purpose is to delete duplicated blocks of data. See the section titled "SMF DATA ORDER CHECKING" on page 160 for further information on SMF sequence checking.

DELDUP is a high overhead function. It should not be used on a routine basis. The DDRETAIN keyword may be used to reduce the overhead if required.

R-CODE: Not affected

6.2.57 DELDUPRV

FORMAT: DELDUPRV

PARM-LIST: NOT ALLOWED

FUNCTION: Instructs SMFUTIL search for and delete duplicate blocks of data records. DELDUPRV differs from DELDUP in that when a record is found out of sort sequence (in reverse order) by more than the value specified (or defaulted) by SORTTOLERANCE, it is assumed that the new data is a duplicate of data already passed. SMFUTIL will skip the current and all incoming records until a record is encountered with a date/time stamp that is greater than the last record accepted (prior to the reverse order data block).

NOTE: DELDUPRV should not be used if type 2 and 3 records are present in the data and are selected for output or if the data was originally put together out of dump sequence order. DELDUPRV is provided to assist in cleaning up old tapes that had a MOD disposition during a system or copy job failure and thus contain duplicated blocks of data. If data was placed on the tapes out of the sequence it was dumped from the SMF data sets, DELDUPRV should not be used. The SORTTOLERANCE keyword should be specified and a relatively large tolerance should be specified (i.e. 15 minutes or more). Too small a sort tolerance may result in valid data being deleted from the output data set. The sort tolerance must be smaller than the shortest possible interval between SMF dump jobs. Too large a SORTTOLERANCE will result in duplicate blocks of data being bypassed.

R-CODE: Not affected

6.2.58 DELETERC

FORMAT: DELETERC(retcode)

retcode An integer value of 3 or 7.

PARM-LIST: A value must be provided.

FUNCTION: Instructs SMFUTIL as to the return code at or below which dataset deletions requested via the CATINPUT or GDGINPUT control keywords will be performed. This is the MAXIMUM return code that will allow deletions. Any return code above the value will suppress deletions. The acceptable values of 3 or 7 are intended to allow a run completion of 0 or 4 plus any split indication return code of 1, 2 or 3.

The default value is 7.

R-CODE: Not directly affected.

6.2.59 DETAIL

FORMAT: DETAIL

INVERSE: NODETAIL

PARM-LIST: NOT ALLOWED

FUNCTION: Instructs SMFUTIL to produce a detailed statistics report for each record type encountered. Each line of the output report will represent one record type. Data produced for each type will be:

1. Record Type Number.
2. Number of records input.
3. Number of records output.
4. Low time/date encountered on input.
5. High time/date encountered on input.
6. Low date/time encountered on output.
7. High date/time encountered on output.
8. Minimum record length on input.
9. Maximum record length on input.
10. Average record length on input.
11. Minimum record length on output.
12. Maximum record length on output.
13. Average record length on output.

NOTE: Average record lengths are calculated by dividing the sum of all lengths encountered for a record type by the total record count. If the accumulation of record lengths becomes too large, the calculation will be suspended and '*****' will be placed in the output field. The TERSE keyword may be used to limit the report to only those record types that were written to an output data set.

R-CODE: Not affected.

6.2.60 DISKONLY

FORMAT: DISKONLY

PARM-LIST: NOT ALLOWED.

FUNCTION: To instruct SMFUTIL that, during a search of the ARCHIVE database for input data set selection, only data sets that reside on disk are to be considered for selection. All data set marked as residing on tape devices will be bypassed.

R-CODE: Not affected.

6.2.61 DSPLIT (OBSOLETE)

FORMAT: DSPLIT

NOTE: This keyword is still available for compatibility with prior releases. It may be removed in a future release. It is strongly recommended that this keyword not be used to accomplish a desired split function. The dynamic allocation control block method should be used instead by specifying the split boundary via the DDAEND keyword.

PARM-LIST: NOT ALLOWED

FUNCTION: To instruct SMFUTIL to check for a change in the day-of-week of the output data directed to the output data sets. If a change is detected, SMFUTIL will cease placing data on the primary output data set (SYSUT2) and begin placing data (starting with the new day) on the alternate output data set (SYSUT3).

NOTE: See the section entitled "INPUT DATA SPLITTING" on page 166 for information about requirements for proper splitting results.

R-CODE: 0 No day change. All output on SYSUT2.
1 Change detected. SYSUT3 has data.

6.2.62 DSPLITDP (OBSOLETE)

FORMAT: DSPLITDP

NOTE: This keyword is still available for compatibility with prior releases. It may be removed in a future release. It is strongly recommended that this keyword not be used to accomplish a desired split function. The dynamic allocation control block method should be used instead by specifying the split boundary via the DDAEND keyword.

PARM-LIST: NOT ALLOWED

FUNCTION: To instruct SMFUTIL to check for a change in the day-of-week of the output data directed to the duplex data sets. If a change is detected, SMFUTIL will cease placing data on the primary duplex output data set (SYSUT2D) and begin placing data (starting with the new day) on the alternate duplex output data set (SYSUT3D).

NOTE: See the section entitled "INPUT DATA SPLITTING" on page 166 for information about requirements for proper splitting results.

R-CODE: 0 No day change. All duplex output on SYSUT2D.
1 Change detected. SYSUT3D has data.

6.2.63 DTRANGE

FORMAT:	DTRANGE(start:end[,start:end,...])
start	The date/time stamp value to begin copying data from.
end	The date/time stamp value to terminate the range to be copied.
PARM-LIST:	At least 1 start:end pair must be coded.
FUNCTION:	Instructs SMFUTIL to copy all data from the starting date/time value to the ending date time value. Multiple ranges may be coded and multiple DTRANGE keywords may be specified. The date/time value may be coded in the old or new syntax. The old syntax has the format of 'ccyydddhmmsssth' where:
'cc'	Has the value 19 to 29 (century value).
'yy'	Has the value 00 to 99 (year).
'ddd'	Has the value 001 to 366 (Julian day value).
'hh'	Has the value of 00 to 23 (hour value).
'mm'	Has the value of 00 to 59 (minute value).
'ss'	Has the value of 00 to 59 (seconds value).
'th'	Has the value of 00 to 99 (tenths and hundredths of a second).
The new syntax has the format of 'my/dm/ccyy-hhmmsssth' where:	
'my'	Has the value of 01 to 12 (month of year).
'dm'	Has the value of 01 to 31 (day of month).
'cc'	Has the value 19 to 29 (century value).
'yy'	Has the value 00 to 99 (year).
'hh'	Has the value of 00 to 23 (hour value).
'mm'	Has the value of 00 to 59 (minute value).
'ss'	Has the value of 00 to 59 (seconds value).
'th'	Has the value of 00 to 99 (tenths and hundredths of a second).
NOTE:	Each date/time string may be truncated to less than the required number of characters if desired. The range start positions not coded will default to zeros and the range end positions not coded will default to high values. If the input data set (SYSUT1xx) is not provided in the JCL or via a DDA block definition and the SYSARCH archive database is available to the execution, SMFUTIL will attempt to locate the required data via the archives.
NOTE:	DTRANGE is mutually exclusive with the DATE, EDATE, GDATE, and/or xTIME keywords.
EXAMPLE 1:	DTRANGE(200236008:200236107)
EXAMPLE 2:	DTRANGE(12/26/2002-08:12/27/2002-07)
	Either of the above formats would select December 26, 2002 from 08:00:00.00AM thru December 27, 2002 at 07:59:59.99AM.
R-CODE:	Not affected.

6.2.64 DTVALOFF

FORMAT: DTVALOFF

PARM-LIST: NOT ALLOWED

FUNCTION: To instruct SMFUTIL to turn off date and time validation on input records. All date and time functions are disabled. SMFUTIL will pass data from input to output without validating that the SMF header contains a valid date and time. This may be used to force SMFUTIL to copy data of questionable character.

NOTE: The date field in the input record must be in packed format. If it contains a non-packed data value, it will automatically be discarded whether DTVALOFF is coded or not.

R-CODE: Not affected.

6.2.65 DUMPCLEARALL

FORMAT: DUMPCLEARALL

ALIAS: DMPCLALL

PARM-LIST: NOT ALLOWED

FUNCTION: To instruct SMFUTIL to dump and clear all SMF data sets that are marked "DUMP REQUIRED". This is done by searching the SMF system control blocks for all MAN data sets that currently contain SMF data and allocating them as input data sets. The data sets are allocated in the order that they contain data (i.e. the data set containing the oldest data is allocated first followed by the next oldest and so on). An optional switch of the recording SMF data set will be done as requested by the MAN1LEAV setting and/or a user specified "ISMF" keyword. After any required switch is completed, the currently active SMF MAN data set will be ignored and all other MAN data sets marked "DUMP REQUIRED" will be allocated as input data sets (SYSUT1xx) in the order in which they contain data (oldest first). The allocations will be the only input data set(s) allowed for the execution. Once input (data copy) operations are complete, each of the allocated data sets will be cleared for reuse.

NOTE: DUMPCLEARALL is mutually exclusive with the CLEAR keyword and with any other input data sets specified.

The MAN1LEAV parameter (set to YES in the SMFUDFLT module as shipped) instructs SMFUTIL to leave the primary SMF data set as available (if possible) after DUMPCLEARALL processing is complete. This allows a subsequent "Z EOD" command to switch to the primary data set and the following IPL to begin recording on the primary without leaving data out of order in another alternate data set. During the IPL, because another MANx data set contains data, the IEFU29 exit will be triggered and a dump job submitted. Once the IPL is complete this dump job (again invoking DUMPCLEARALL with MAN1LEAVE) will cause a switch off of the primary data set.

The ISMF keyword may also be specified to request that always do at least one switch of the recording MAN data set, if possible. This ensures that all currently available data is dumped.

NOTE: If SUCLEAR=YES is defaulted or SUCLEAR specified at execution time, the MAN1LEAVE option is not required and should be turned off. SMF will not reuse the primary dataset until SMFUTIL has properly cleared it and notified SMF of it's availability.

R-CODE: Not directly affected.

6.2.66 EDATE

FORMAT: EDATE(dd/mm/ccyy:dd/mm/ccyy,[dd/mm/ccyy:dd/mm/ccyy,...])

dd/mm/ccyy Gregorian date(s) in European format.

PARAM-LIST: At least one date required.

FUNCTION: To instruct SMFUTIL that only data on a particular date or within a particular date range is eligible to be copied. All input data records with dates outside the specified date or range will be discarded. If only one date is specified, only data for that day will be eligible for copying. Multiple start:end date range pairs may be coded.

If the input data set (SYSUT1xx) is not provided in the JCL or via a DDA block definition and the SYSARCH archive database is available to the execution, SMFUTIL will attempt to locate the required data via the archives.

NOTE: When EDATE is specified, an automatic default specification of SHORTCPY is assumed. The GOTOEOF or NOSTOP keyword may be used, after the EDATE keyword, to override this default.

NOTE: EDATE is mutually exclusive with the DTRANGE keyword.

R-CODE: 0 Data found for date (range).
4 No data found for date (range).

6.2.67 EOF370

FORMAT: EOF370

PARAM-LIST: NOT ALLOWED

FUNCTION: Instructs SMFUTIL to look for VSAM end-of-file indication using the MVS/370 indicator instead of the MVSA/XA or MVS/ESA indicator. This should be used when SMF MAN data sets created by an MVS/370 system are being read.

NOTE: When the DUMPCLEARALL or MANXALOC function is executed on an MVS/370 system, and it causes the dynamic allocation of one or more MAN data sets from the SMCA control block in the system, "EOF370" is automatically turned on. It need not be specified by the user.

R-CODE: Not affected.

6.2.68 ESTAEON

FORMAT: ESTAEON

INVERSE: ESTAEOFF or NOESTAE

PARAM-LIST: NOT ALLOWED

FUNCTION: Instructs SMFUTIL to establish an ESTAE environment to trap unexpected ABENDs. Where possible, the ABEND will be overridden and the execution will continue.

NOTE: As shipped, the SMFUFLT module sets the ESTAE environment on by default. This should not be changed or overridden at execution time (via ESTAEOFF) unless directed to do so by Product Support personnel.

R-CODE: Not directly affected however if the ABEND retry routine determines that the ABEND threshold is reached (15 consecutive ABENDs), it will terminate the run with a return code of 12.

6.2.69 EXCLUDE

FORMAT: EXCLUDE(rectype[,rectype,...])
 EXCLUDE(rectype:rectype[,rectype])
 EXCLUDE(rectype(subtype)[,rectype])

rectype A 1 to 3 decimal number in the range of 0 to 255 that identifies a record type to be excluded for consideration in processing. A range may be specified by separating two numbers with a colon. One or more specific subtypes may be excluded for a single record type by immediately following the record type with one or more subtype specifications enclosed in parenthesis. The subtypes are applied to that record type only. All other subtypes of that record type will be selected. For subtypes to be useable, the subtype indicator for the record type must be on the record definition table.

PARM-LIST: At least one record type required.

FUNCTION: Instructs SMFUTIL to exclude specific record types from consideration for processing.

R-CODE: Not affected.

6.2.70 EXONLY

FORMAT: EXONLY(rec-type[,rec-type,...])

rec-type A 1 to 3 decimal number in the range of 0 to 255 that identifies a record type to be included for consideration in processing. A range may be specified by separating two numbers with a colon.

PARM-LIST: At least one record type required.

FUNCTION: Instructs SMFUTIL to include specific record types from consideration for processing. This function differs from the EXCLUDE function in that all record types are turned on for selection (even if they had been previously excluded implicitly or explicitly) and only the specified record types are turned off for selection.

R-CODE: Not affected.

6.2.71 EXIT

FORMAT: EXIT(exit-name[,exit-name,...])

exit-name A 1 to 8 character name of a user supplied load module exit. The load module must be available in the LINKLST or on LPALIB, JOBLIB, or STEPLIB.

PARM-LIST: At least one load module name required.

FUNCTION: Instructs SMFUTIL to take a user supplied exit for each record to be copied to the output data set. If more than one exit load module name is specified, the modules are called in the order given. If more names are required than will fit on one card, multiple EXIT control keywords may be used. See the section titled "USER EXITS" on page 215 for more information on implementing user exits.

R-CODE: 0 Load module located.
 8 Specified load module not found.

6.2.72 FASTMOD

FORMAT: FASTMOD
 INVERSE: NOFASTMOD
 PARM-LIST: NOT ALLOWED.
 FUNCTION: To instruct SMFUTIL to use the SYSCKPT data set to retain and use block id information for high-speed positioning of cartridge data sets that are to be extended (disposition of MOD).
 R-CODE: Not affected

6.2.73 FORCE

FORMAT: FORCE
 INVERSE: NOFORCE
 PARM-LIST: NOT ALLOWED.
 FUNCTION: To instruct SMFUTIL to continue execution of the DATESCAN function even if out of sequence data is detected greater than the SORTTOLERANCE value.
 R-CODE: Not affected

6.2.74 GDATE

FORMAT: GDATE(mm/dd/ccyy:mm/dd/ccyy[,mm/dd/ccyy:mm/dd/ccyy,...])
 mm/dd/ccyy Gregorian date(s).
 PARM-LIST: At least one date required.
 FUNCTION: To instruct SMFUTIL that only data on a particular date or within a particular date range is eligible to be copied. All input data records with dates outside the specified date or range will be discarded. If only one date is specified, only data for that day will be eligible for copying. Multiple start:end ranges may be coded.
 If the input data set (SYSUT1xx) is not provided in the JCL or via a DDA block definition and the SYSARCH archive database is available to the execution, SMFUTIL will attempt to locate the required data via the archives.
 NOTE: When GDATE is specified, an automatic default specification of SHORTCPY is assumed. The GOTOEOF or NOSTOP keyword may be used, after the GDATE keyword, to override this default.
 NOTE: GDATE is mutually exclusive with the DTRANGE keyword.
 R-CODE: 0 Data found for date (range).
 4 No data found for date (range).

6.2.75 GDGCKPTI

FORMAT:	GDGCKPTI(ckptgdg[(SKIPLAST,DELETE)],...)
ckptgdg	A 1 to 4 character CKPTGDG checkpoint record name. This corresponds to the name specified by a DDA CKPTGDG parameter in a previous run. Multiple ckptgdg specifications are supported (each with the optional sub-parameter).
SKIPLAST	An optional keyword to indicate not to process the last (current) generation indicated by the ckptgdg checkpoint record.
DELETE	An optional keyword to indicate generations input are to be deleted at successful completion of run.
FUNCTION:	To instruct SMFUTIL a DDA created CKPTGDG checkpoint record is to be interrogated and all generations of the indicated GDG index are to be dynamically allocated and processed as input data sets in ascending order. If SKIPLAST is specified, the last (current or (0)) generation will be omitted. If DELETE is specified, all GDG data sets that were allocated for the gdg index will be deleted at the end of the run. This deletion is done after all processing has completed. If the return code of the run is 8 or greater, the DELETE step is bypassed.
NOTE:	<p>The SYSCKPT data set is required to be present in the JCL or via the CKPTDB control keyword. The named CKPTGDG record must have previously been created with an SMFUTIL execution utilizing the same SYSCKPT data set and with a matching CKPTGDG specification in a DDA block structure for an output GDG data set.</p> <p>Use of GDGCKPTI causes the INPTGDG1/2 to be used to checkpoint the first input record and LASTGDG1 to be used to checkpoint the last input record. This allows the same SYSCKPT data set to support two different processes, one where the CKPTGDG checkpoint records are created and a second where the GDGCKPTI parameter utilizes the CKPTGDG records for input. The SYSCKPT data set must be unique to a paired execution as outlined above and not used for other runs. If multiple CKPTGDG specifications are contained in a single execution, only one subsequent execution utilizing a single GDGCKPTI specification is to be considered the linked process. All other executions utilizing the same SYSCKPT data set (and GDGCKPTI specifications matching the other CKPTGDG specifications) should specify the NOCKPTIN keyword to avoid conflicting with the input data check pointing in the primary linked process.</p> <p>After a successful execution using GDGCKPTI, all \$GDG checkpoint records defined are updated. The starting and ending points are set to the next absolute generation following the last one just processed. This means that a related execution utilizing the CKPTGDG with this same name is required prior to another GDGCKPTI execution.</p>
R-CODE:	<p>0 Data found for date (range).</p> <p>4 Missing GDG generation detected.</p>

6.2.76 GDGINPUT

FORMAT:	GDGINPUT(gdindex[(SKIPLAST],[DELETE],[limit],[latest],[BIDH]),...)
gdindex	A 1 to 35 character GDG index name to be input. Multiple gdindex specifications are supported (each with the optional parameter list).
SKIPLAST	An optional keyword to indicate not to process the last (current) generation of this GDG index.
DELETE	An optional keyword to indicate data sets found under this GDG index and used as input are to be deleted at successful completion of the run. The maximum return code allowing deletions is defined by the DELETERC default value.
limit	An optional integer value in the range of 1 to 255 to limit the number of GDG data sets that may be selected for this GDG index.
latest	An optional integer value in the range of 1 to 255 to limit the dataset selection to this number of the “latest” (newest) data sets in the list.
BIDH	If coded, the BIDH keyword indicates that, during an ARCHIVEINPUT execution, datasets under the specified index are to have BID values recorded on an hourly rather than a daily basis. The dataset(s) must contain a single days worth of data for this to be effective else SMFUTIL will revert to daily BID values for the dataset.
FUNCTION:	<p>To instruct SMFUTIL that the existing generation(s) of a GDG index are to be dynamically allocated and processed as input data sets in ascending order. If SKIPLAST is specified, the last (current or (0)) generation will be omitted. If DELETE is specified, all GDG data sets that were allocated for the gdindex will be deleted at the end of the run. This deletion is done after all processing has completed. If the return code of the run is 8 or greater, the DELETE function is bypassed.</p> <p>If the ‘limit’ sub-parameter is specified, no more than the specified number of data sets will be selected for input processing for this prefix. All data sets in the catalog above this limit will be ignored during the current execution. Note that you may specify the ‘DELETE’ keyword with a limit value because delete only applies to data sets actually selected for input. This means that data sets NOT selected for input because of a user specified limit will NOT be subject to deletion.</p> <p>If the ‘latest’ parameter is specified, only the last number of datasets specified will be considered for input. This means if there are 10 datasets under the prefix specified and a ‘latest’ value of 3 is specified, only the top (last) 3 datasets in the GDG list will be considered. Use the ‘latest’ parameter carefully, especially if the DELETE keyword is specified, as out of order data processing is likely.</p> <p>If ‘limit’ and ‘latest’ are both specified, ‘limit’ will only be applied to datasets eligible for selection after ‘latest’ has been applied. This effectively means that ‘limit’ can never be larger than ‘latest’.</p> <p>Judicious usage of ‘limit’ and ‘latest’ can be utilized to selectively process portions of a GDG list.</p>
NOTE:	If ‘DELETE’ is specified, only datasets actually selected for input processing will be eligible for DELETE processing. Datasets skipped because of SKIPLAST, ‘limit’ or ‘latest’ specifications will not be deleted.
R-CODE:	0 Data found for date (range). 4 Missing GDG generation detected.

6.2.77 GENDSND

FORMAT: GENDSND

NOTE: **OBSOLETE – Use dynamic allocation with SUFFIX keyword instead.**

PARM-LIST: NOT ALLOWED.

FUNCTION: To instruct SMFUTIL to generate a data set name suffix for the data set name specified for DDNAME SYSUT2 and SYSUT2D if present. The suffix will have the form ".Jccyyddd" where ccyyddd is the Julian date of the first record written to SYSUT2 (i.e. ".J2002110"). SYSUT2 must have a data set name prefix of 35 characters or less specified via the 'DSN=' parameter in the JCL. It may not be a temporary data set or a GDG (relative or specific). It must have a disposition of NEW and it must be on a tape device. Disk data sets are not allowed as the Format 1 DSCB is created by the operating system prior to program initialization. The altered data set name would not be found.

R-CODE: 0 Data set name suffix added.
 4 GDG data set name structure found on DD. Bypassed.
 8 Invalid SYSUT2 data set for GENDSND.

6.2.78 GENDSNM

FORMAT: GENDSNM

NOTE: **OBSOLETE – Use dynamic allocation with SUFFIX keyword instead.**

PARM-LIST: NOT ALLOWED.

FUNCTION: To instruct SMFUTIL to generate a data set name suffix for the data set name specified for DDNAME SYSUT2 and SYSUT2D if present. The suffix will have the form ".mmmcyy" where mmm is the first 3 characters of the month and ccyy is the year from the date of the first record written to SYSUT2. SYSUT2 must have a data set name prefix of 36 characters or less specified via the 'DSN=' parameter in the JCL. It may not be a temporary data set or a GDG (relative or specific). It must have a disposition of NEW and it must be on a tape device. Disk data sets are not allowed as the Format 1 DSCB is created by the operating system prior to program initialization. The altered data set name would not be found.

R-CODE: 0 Data set name suffix added.
 4 GDG data set name structure found on DD. Bypassed.
 8 Invalid SYSUT2 data set for GENDSNM.

6.2.79 GENDSNW

FORMAT: GENDSNW

NOTE: **OBSOLETE – Use dynamic allocation with SUFFIX keyword instead.**

PARM-LIST: NOT ALLOWED.

FUNCTION: To instruct SMFUTIL to generate a data set name suffix for the data set name specified for DDNAME SYSUT2 and SYSUT2D, if present. The suffix will have the form ".Wxxxccyy" where 'xx' is logical week of the year (i.e. 01 to 53) and ccyy is the year from the date of the first record written to SYSUT2. SYSUT2 must have a data set name prefix of 36 characters or less specified via the 'DSN=' parameter in the JCL. It may not be a temporary data set or a GDG (relative or specific). It must have a disposition of NEW and it must be on a tape device. Disk data sets are not allowed as the Format 1 DSCB is created by the operating system prior to program initialization. The altered data set name would not be found. The DDA block structure can be used to dynamically allocated a disk data set and add a suffix to it.

R-CODE: 0 Data set name suffix added.
 4 GDG data set name structure found on DD. Bypassed.
 8 Invalid SYSUT2 data set for GENDSNM.

6.2.80 GOTOEOF

FORMAT: GOTOEOF

ALIAS: NOSTOP

INVERSE: STOP

PARM-LIST: NOT ALLOWED

FUNCTION: To instruct SMFUTIL to process the entire input file until physical end-of-file is encountered. Normally, SMFUTIL will terminate early if the following conditions occur:

1. CHKOSORT is in effect.
2. A particular date/time range is being copied.
3. A date/time is encountered that is above the requested range.
4. The output data is in sorted order to this point.

Under these conditions, SMFUTIL will assume that the remainder of the input data is sorted, and all of the requested range has been copied. This prevents processing large amounts of data that are past the required date/time range. If the input data is suspect as to sorted ordering, the GOTOEOF keyword will force SMFUTIL to continue processing data even after the date/time range being requested is passed.

NOTE: Caution should be used in utilizing the GOTOEOF keyword as substantial amounts of data may be processed that is not required.

R-CODE: Not affected.

6.2.81 IDTRANGE

FORMAT:	IDTRANGE(ddn-start:end[,ddn-start:end,...])
ddn	A 1 to 8 character DDNAME that is to be limited in the range of date/time record values it will provide.
start	A 1 to 15 character date/time stamp value for the specified DDNAME to begin providing data. All data prior to this date/time will not be accepted for input processing from the specified DDNAME.
end	A 1 to 15 character date/time stamp value for the specified DDNAME to cease receiving data. All data after this date/time will not be accepted for input processing from the specified DDNAME.
PARM-LIST:	At least 1 ddn-start:end pair must be coded.
FUNCTION:	<p>Instructs SMFUTIL to allow only data from the starting date/time value to the ending date/time value to be received from the specified ddname. Multiple ddname/dtrange values may be coded within the parenthesis and multiple ODTRANGE keywords may be specified. Each DDNAME may be specified multiple times with a different range for each specification. The date/time value has the format of 'ccyydddhhmsssth' where:</p> <p>'cc' Has the value 19 to 29 (century value). 'yy' Has the value 00 to 99 (year). 'ddd' Has the value 001 to 366 (Julian day value). 'hh' Has the value of 00 to 23 (hour value). 'mm' Has the value of 00 to 59 (minute value). 'ss' Has the value of 00 to 59 (seconds value). 'th' Has the value of 00 to 99 (tenths and hundredths of a second).</p> <p>Each date/time string may be truncated to less than 15 characters if desired. The range start positions not coded will default to zeros and the range end positions not coded will default to high values.</p>
NOTE:	<p>Caution should be exercised in specifying multiple date/time ranges for the same DDNAME and mixing IDTRANGE with ITIMERANGE in the same execution. SMFUTIL does not validity check the specifications with each other. Each specification is considered independently.</p> <p>The IDTRANGE keyword is only a filter. It does not select any specific record types from the specified ddname. It only prevents data from being processed as input from the DDNAME unless it is within the specified date/time range.</p>
R-CODE:	Not affected.

6.2.82 INCLUDE

FORMAT:	INCLUDE(rectype[,rectype,...]) INCLUDE(rectype:rectype[,rectype]) INCLUDE(rectype(subtype)[,rectype])
rectype	A 1 to 3 decimal number in the range of 0 to 255 that identifies a record type to be included for consideration in processing. A range may be specified by separating two numbers with a colon. One or more specific subtypes may be selected for a single record type by immediately following the record type with one or more subtype specifications enclosed in a single set of parenthesis. The subtype are applied to that record type only. For subtype to be useable, the subtype indicator for the record type must be on the record definition table.
PARM-LIST:	At least one record type required.
FUNCTION:	To instruct SMFUTIL to select specific record types for processing.
R-CODE:	Not affected.

6.2.83 INCONLY

FORMAT:	INCONLY(rec-type[,rec-type,...])
rec-type	A 1 to 3 decimal number in the range of 0 to 255 that identifies a record type to be included for consideration in processing. A range may be specified by separating two numbers with a colon.
PARM-LIST:	At least one record type number required.
FUNCTION:	To instruct SMFUTIL to copy specific record types to the output data set(s). This function differs from the INCLUDE function in that all record types are turned off for selection (even if they had been previously implicitly or explicitly selected) and only the record types specified are turned on. This is useful for selecting only a specific record type for a SUBSET such as JOBNAME.
NOTE:	Multiply INCONLY parameters are self defeating as all record types are turned off each time the INCONLY parameter is encountered and only the record types within the parameter list are turned back on. In essence, only the last INCONLY parameter will be honored.
R-CODE:	Not affected.

6.2.84 INDTVOFF

FORMAT:	INDTVOFF
PARM-LIST:	NOT ALLOWED
FUNCTION:	Instructs SMFUTIL to bypass date/time block validation of input data sets selected via the archive. This is a special capability for former CA/JARS sites that converted their CA/JARS SCDS database to an SMFUTIL ARCHIVE. Normally, when reading a data set located via the archive, SMFUTIL ensures that only data within the ARCHIVE record start and end date/time stamps is used. This can cause a problem with ARCHIVE data originally created by CA/JARS as some of the SCDS records may not correctly reflect the actual end times of the data sets accurately. The INDTVOFF effectively tells SMFUTIL to read the entire CA/JARS created data set until the date/time range requested by the user is passed.
R-CODE:	Not affected.

6.2.85 INITEXIT

FORMAT:	INITEXIT(exit-name)
exit-name	A 1 to 8 character name of a user supplied load module exit. The load module must be available in the LINKLST or on LPALIB, JOBLIB, or STEPLIB.
PARM-LIST:	A load module name is required.
FUNCTION:	Instructs SMFUTIL to take a user supplied initialization exit routine prior to beginning the processing of data. See the section titled "USER EXITS" on page 215 for more information on the implementation and use of user exits. Only one initialization exit may be use. If more than one INITEXIT parameter is provided, the last one encountered will be used.
R-CODE:	Not affected.

6.2.86 INPUTDATADISTRIBUTION

FORMAT:	INPUTDATADISTRIBUTION[(type[,type])]
ALIAS:	IDIST
PARM-LIST:	OPTIONAL
type	A keyword of ALL, RECTYPE or BOTH indicating the type of percentage to be reported. RECTYPE is the default. It instructs SMFUTIL to calculate the percentage based on only the specific type of record sum for the 24 hour period. Each record type line will add to 100% for that 24 hour period. The optional report type ALL indicates the percentage is to be calculated base on the total sum of ALL record types for the 24 hour period. Note that to produce both reports, the keyword BOTH may be specified in the parameter list. For example, IDIST(BOTH) would request both reports.
FUNCTION:	To instruct SMFUTIL to produce a Data Distribution report for each input data set. This report shows the percentage of each record type encountered for each hour of the day out of the total of that record type or all record types for that data set in the 24 hour period. If more than one day's worth of data is processed, the percentage represents that hour for all days processed for the data set.
NOTE:	This is a moderately high overhead report request. It cannot be requested by default. It should not be requested when large volumes of data will be processed. An additional 25K of region will be required for each input data set processed. Requesting both reports does not significantly increase the overhead.
R-CODE:	Not affected.

6.2.87 INPUTDETAIL

FORMAT: INPUTDETAIL

ALIAS: IDetail

INVERSE: NOINPUTDETAIL

PARM-LIST: NOT ALLOWED

FUNCTION: Instructs SMFUTIL to produce a detailed statistics report for each input

data set. Each line of the output report will represent one record type. Data produced for each type will be:

1. Record Type Number.
2. Number of records this output data set.
3. Low date/time encountered this output data set.
4. High date/time encountered this output data set.
5. Minimum record length this output data set.
6. Maximum record length this output data set.
7. Average record length this output data set.

NOTE: Average record lengths are calculated by dividing the sum of all lengths encountered for a record type by the total record count. If the accumulation of record lengths becomes too large, the calculation will be suspended and '*****' will be placed in the output field.

R-CODE: Not affected.

6.2.88 INPUTLIMIT

FORMAT: INPUTLIMIT(number)

number An integer of 1 to 9 digits.

PARM-LIST: REQUIRED

FUNCTION: To instruct SMFUTIL to limit the input record count to a preset number. This is useful if only a small sample of input data is desired.

R-CODE: Not affected.

6.2.89 IOELIMIT

FORMAT: IOELIMIT(number)

number An integer of 1 to 6 digits.

PARM-LIST: REQUIRED

FUNCTION: To instruct SMFUTIL to limit the number of consecutive input data I/O errors that will be tolerated to the specified value. If this limit is reached, SMFUTIL will terminate. This serves to prevent flooding the Operator console with IOS0001 messages. If IOELIMIT is not specified at execution time, the IOELIMIT value specified at installation (default of 25) will be used.

NOTE: This value does not limit the total number of I/O errors during the run. It only limits the number of CONSECUTIVE input data I/O errors. It also has no effect on output data set I/O errors as these will result in the immediate termination of the run.

R-CODE: Not affected.

6.2.90 ISETNAME

FORMAT: ISETNAME(set-name)

set-name The set name to be assigned to archived input data.

PARM-LIST: REQUIRED

FUNCTION: To supply the ARCHIVE record identifying set name to be used to archive input data in lieu of 'NORMAL'. This allows special purpose archive sets to be created in the archive database. This keyword only has meaning if 'ARCHINPT' is also specified.

R-CODE: NO EFFECT.

6.2.91 ISMF

FORMAT: ISMF

PARM-LIST: NOT ALLOWED

FUNCTION: Instructs SMFUTIL to ensure at least one switch of the recording MAN data set is done during a DUMPCLEARALL processing run.

NOTE: ISMF will be ignored if MAN1LEAV is specified (or defaulted) and the primary data set is currently the only available data set to receive the switch. The MAN1LEAV specification overrides the request to force at least one switch. ISMF is also ignored if there are currently no data sets marked as available to accept recording when the switch is done. This would be the case when all data sets are marked "DUMP REQUIRED" except the active data set.

R-CODE: Not affected.

6.2.92 ISMFDATA

FORMAT: ISMFDATA

PARM-LIST: NOT ALLOWED

FUNCTION: Instructs SMFUTIL that it may perform one switch of the recording MAN data set if required during a DUMPCLEARALL processing run to ensure at least 1 MANx dataset is available for dumping.

NOTE: ISMFDATA works like ISMF except the switch is optional instead of being required. In addition, an ISMFDATA switch will only be done if there are no MANx datasets already marked "DUMP REQUIRED".

ISMFDATA is actually a safety net feature where DUMPCLEARALL is used but ISMF and MAN1LEAVE are not specified. It allows a DUMPCLEARALL run to continue by doing a switch and thus creating data to be dumped. Otherwise, the run would end with a return code of 12 because no data was available to be dumped.

R-CODE: Not affected.

6.2.93 ITIME

FORMAT: ITIME(s-time[:e-time])

x-time A 1 to 8 digit number in 'hhmmssst' format as a single string.

'hh'	Has a range of 0 to 23
'mm'	Has a range of 0 to 59
'ss'	Has a range of 0 to 59
'th'	Has a range of 0 to 99

PARM-LIST: At least start time must be specified.

FUNCTION: To instruct SMFUTIL to select only records within a specific time range to be copied to the output data set(s). The time range is inclusive. This means that only records within the range are eligible for selection. Records outside the range, no matter what the date, will be discarded. If only the start-time is specified, midnight will be assumed as the end time. If less than the full hhmmssst format is specified, the number will be assumed to be left justified with the exception that a single digit number will be padded to the left with a zero. The start time will be padded to the right with low values (0's) and the end time will be padded to the right with high values (i.e. 595999). The start time must be less than or equal to the end time.

EXAMPLE: ITIME(1:13) is the same as
 ITIME(01000000:13595999).

ITIME(16:2000) is the same as
 ITIME(16000000:20005999).

R-CODE:

0	Data copied.
4	Requested time range not found.

6.2.94 ITIMERANGE

FORMAT:	ITIMERANGE(ddn-[tstart][:tend][,ddn-tstart:tend,...])
ddn	A 1 to 8 character DDNAME that is to be limited in the range of record time values it will provide.
tstart	A 2 to 8 character time stamp value for the specified DDNAME to begin providing data. All data prior to this time, within each day, will not be ignored from the specified ddname.
end	A 2 to 8 character time stamp value for the specified DDNAME to cease providing data. All data after this time, within each day, will be ignored from the specified ddname.
PARM-LIST:	At least 1 ddn-tstart:tend pair must be coded.
FUNCTION:	<p>Instructs SMFUTIL to allow only data from the starting time value to the ending time value to be received from the specified ddname. Multiple ddname/trange values may be coded within the parenthesis and multiple ITIMERANGE keywords may be specified. Each DDNAME may be specified multiple times with a different range for each specification. The time value has the format of 'hhmmsssth' where:</p> <p>'hh' Has the value of 00 to 23 (hour value). 'mm' Has the value of 00 to 59 (minute value). 'ss' Has the value of 00 to 59 (seconds value). 'th' Has the value of 00 to 99 (tenths and hundredths of a second).</p> <p>Each time string may be truncated to less than 8 characters if desired but the minimum is 2 digits ('hh' specification). The range start positions not coded will default to zeros and the range end positions not coded will default to high values.</p>
NOTE:	Caution should be exercised in specifying multiple time ranges for the same DDNAME and mixing ITIMERANGE with IDTRANGE in the same execution. SMFUTIL does not validity check the specifications with each respect to other. Each specification is considered independently.
R-CODE:	Not affected.

6.2.95 IUCB

FORMAT:	IUCB(ucbadrs[,ucbadrs-ucbadrs])
ALIAS:	UCB
ucbadrs	A two to four character UCB address mask or UCB address range.
PARM-LIST:	At least one UCB mask must be coded.
FUNCTION:	To instruct SMFUTIL to include records that contain specific UCB addresses that match the specified mask.
NOTE:	IUCB automatically turns off all record types that SMFUTIL does not have UCB address location data on. The SMFUTLRT can be updated to contain the location of volume serial data for user type records.
NOTE:	Any record type selected for processing by other means (i.e. via an INCLUDE keyword) that does not contain volume serial information will be pass directly to output processing.
R-CODE:	Not directly affected. However, if a UCB mask is specified that causes no records to be selected a return code of 8 will be issued unless COPY00K is specified.

6.2.96 IVOLUME

FORMAT:	IVOLUME(volmask)
ALIAS:	IVOLSER
volmask	A one to six character volume serial mask.
PARM-LIST:	At least one volmask must be coded.
FUNCTION:	To instruct SMFUTIL to include records that contain volume serial numbers that match the specified mask.
NOTE:	IVOLSER automatically turns off all record types that SMFUTIL does not have volume serial location data on. The SMFUTLRT can be updated to contain the location of volume serial data for user type records.
NOTE:	Any record type selected for processing by other means (i.e. via an INCLUDE keyword) that does not contain volume serial information will be pass directly to output processing.
R-CODE:	Not directly affected. However, if a volume mask is specified that causes no records to be selected a return code of 8 will be issued unless COPY00K is specified.

6.2.97 IVSTART

FORMAT:	IVSTART(volnum M L)
volnum	Volume sequence number to start with. Range must be from 1 to n where n is the number of volumes in the data set.
M	Start with the volume in the middle (if odd number of volumes) or just past the middle (if even number of volumes) in the catalog set.
L	Start with last volume in catalog set.
PARM-LIST:	Either "M", "L" or a volume sequence number must be coded.
FUNCTION:	To instruct SMFUTIL to start with a volume (of a multiple volume input data set) other than the first volume. A parameter specification of "M" will instruct SMFUTIL to select the volume closest to the logical middle of the volume set as the starting volume. A parameter specification of "L" will cause SMFUTIL to select the last volume of the set as the starting volume. This is useful for copying data from the end of a data set.
NOTE:	If SYSUT1 is a set of concatenated data sets, IVSTART is utilized for the first data set only. Subsequent data sets concatenated to the first one will begin at volume 1 of the set if they are multiple volume.
R-CODE:	Not directly affected. However, if a date range is being copied and the first record on the volume mounted is past the end date of the date range, SMFUTIL will terminate with a return code of 4 for date range not found.

6.2.98 JOBNAME

FORMAT:	JOBNAME(job name[,job name,...])
job name	A one to eight character job name or job name prefix for which records are to be selected. If less than 8 characters are specified, all job names that start with the specified string will be selected. To limit the selection to a specific job name that is less than 8 characters, it is necessary to pad the job name to the right with blanks within the parenthesis. A wildcard specification of '*' (an asterisk) may be used in any position to specify that any value is to be accepted in that position.
PARM-LIST:	At least 1 job name specification required.
FUNCTION:	To cause all records for a specific job name (JOB or STC type) or whose job name begins with a specific prefix to be selected. All record types which have Job name capability (see "MODIFICATION AND CUSTOMIZATION" on page 199 for more information) are turned on for selection. Only those records that match the job name specification are selected for output. The job name specification is length sensitive. If the job name specification is less than 8 characters in length, it will be used as a mask to select all job names that begin with the specification. If a job name that is less than 8 characters is desired only, it must be padded to the right with blanks to specify a total length of 8 characters. Other records that do not have job names embedded in them may be specifically selected and will bypass job name validation and will be put to the output data set(s).
R-CODE:	0 Data was written to output. 4 No data output. Job name not found in any input record.

6.2.99 LARGEBLOCK

FORMAT: LARGEBLOCK
 INVERSE: NOLARGEBLOCK
 PARM-LIST: NOT ALLOWED.
 FUNCTION: To instruct SMFUTIL that the IBM Large Block Interface (LBI) is to be requested for all output datasets unless disabled individually with the DDANOLARGEBLOCK keyword. This specification has no meaning for input datasets as LBI is automatically supported for input datasets (if the LBI keyword is specified or LBI=YES is specified in the defaults module).
 R-CODE: Not affected

6.2.100 LBI

FORMAT: LBI
 INVERSE: NOLBI
 PARM-LIST: NOT ALLOWED.
 FUNCTION: To instruct SMFUTIL to enable the IBM Large Block Interface (LBI) interface. This allows SMFUTIL to automatically support input datasets that were created with large blocks and enables the usage of the LARGEBLK and DDALARGEBLOCK keywords for creating output datasets with large blocks.
 R-CODE: Not affected

6.2.101 LINESPERPAGE

FORMAT: LINESPERPAGE(number)
 PARM-LIST: REQUIRED
 number An integer value in the range of 30 to 999 indicating the number of print lines per page of output. The default value is 55 as shipped.
 FUNCTION: Instructs SMFUTIL to print the specified number of output lines on a page before doing a title break for line count.
 NOTE: A specification of '999' suppresses line counting completely allowing longer reports to be produced without title interruption.
 R-CODE: Not affected.

6.2.102 LISTALL

FORMAT: LISTALL

PARM-LIST: NOT ALLOWED

FUNCTION: Instructs SMFUTIL to a statistical report line for all record types, even those with a zero input count.

NOTE: This differs from the VERBOSE/NOTERSE keyword in that VERBOSE only affects record types with zero output counts.

When LISTALL is specified, the “DETAIL RECORD STATISTICS REPORT” will contain 256 (types 0 thru 255) lines at all times.

R-CODE: Not affected.

6.2.103 LISTLICENSE

FORMAT: LISTLICENSE

PARM-LIST: NOT ALLOWED

FUNCTION: Instructs SMFUTIL to print out the current license information it was installed with.

R-CODE: Not affected.

6.2.104 MANILEAVE

FORMAT: MANILEAVE

INVERSE: NOMANILEAVE

PARM-LIST: NOT ALLOWED.

FUNCTION: To instruct SMFUTIL that, if possible, the primary SMF data set is to be left as ‘AVAILABLE’ after DUMPCLEARALL has finished execution. This allows a “Z EOD” to be performed without data being orphaned in the an alternate SMF data set. In conjunction with the ISMF parameter, SMFUTIL may perform up to two SMF switch functions to ensure the primary data set is marked ‘DUMP REQUIRED’ prior to the start of the dump phase.

NOTE: SMFUTIL must be running authorized in order for this function to be available.

NOTE: If SUCLEAR=YES is defaulted or SUCLEAR specified at execution time, the MANILEAVE option is not required and should be turned off. SMF will not reuse the primary dataset until SMFUTIL has properly cleared it and notified SMF of it’s availability.

R-CODE: Not affected.

6.2.105 MANXALOC

- FORMAT:** MANXALOC
- PARM-LIST:** NOT ALLOWED
- FUNCTION:** To instruct SMFUTIL to search the SMF system control blocks for all 'SYS1.SMF' data set that currently contain SMF data and allocate them for input. The data set are allocated in the order that they contain data (i.e. the data set containing the oldest data is allocated first followed by the next oldest and so on). The currently active SMF data set will be allocated last (if 'ACTIVEOK' is selected or defaulted). The allocations will be the final input data set(s) for the execution. They will be allocated AFTER all other input data set specified for the run. This includes data set allocated explicitly, via JCL or DDA blocks, or implicitly, via the ARCHIVE search for selected dates.
- NOTE:** The positioning of the SMF data set allocations as the final input data set allows the user to get up to the minute SMF data logically concatenated to other selected input data. For example, if data from midnight to 6:00am were in the archive and the everything since 6:00am were still out in SMF, a run with the keywords 'TODAY MANXALOC' would produce all of the existing data for the current day, from midnight to the most current record available. The date would not have to be specified and the data would be presented in the order produced by SMF (assuming the data was dumped to the archive in the correct order).
- R-CODE:** Not directly affected but if NOACTIVE is specified and no other SMF data set contain data, no allocation will result, yielding a return code of 12 if no other input data set are specified.

6.2.106 MONTHCAL

- FORMAT:** MONTHCAL(cal-name)
- cal-name** A 1 to 8 character load module name for the user defined month calendar required or the character "0".
- PARM-LIST:** One and only one calendar load module name must be coded or a zero is coded to indicate no calendar is to be used.
- FUNCTION:** To instruct SMFUTIL to use the specified month calendar for this execution.
- NOTE:** The load module named must have been constructed via the SMFUMCAL macro and must be available to be loaded at execution time.
- R-CODE:** NO EFFECT.

6.2.107 MONTHPCAL

- FORMAT:** MONTHPCAL(cal-name)
- cal-name** A 1 to 8 character load module name for the user defined perpetual month calendar required or the digit "0".
- PARM-LIST:** One and only one perpetual calendar load module name must be coded or a zero is coded to indicate no calendar is to be used.
- FUNCTION:** To instruct SMFUTIL to use the specified month calendar for this execution.
- NOTE:** The load module named must have been constructed via the SMFUPCAL macro and must be available to be loaded at execution time.
- R-CODE:** Not affected.

6.2.108 MQSPLIT

FORMAT: MQSPLIT(name-ddname[,name-ddname,...])

name The one to four character MQ subsystem name for a started task.

ddname An output file name the selected records are to be targeted to.

PARM-LIST: At least one name-DDNAME pair required.

FUNCTION: Instructs SMFUTIL to copy record types 115 and 116 to selected outputs based upon their MQ SUBSYSTEM NAME values. If the content of the subsystem name value in the record matches the specified 'name' string, the record will be written to file associated with the specified DDNAME. If the name string is directly followed by the '-' separator character, it will be assumed that the name specification is a prefix mask. Any record containing a name value that matches the specification for its length will be considered selected. If a blank follows the name string (between it and the '-') it will force the name specification to be a full match for the name in each record. This syntax structure allows one specification to select multiple SID's for selection to one output file.

NOTE: If a record is found with an 'name' value not specified in a MQSPLIT statement, it will be written to the standard output data sets. By default, records selected to the target MQSPLIT DDNAMEs via their 'name' values will not be written to the standard output files. The keyword COPYVBTH may be specified to cause the selected records to be written to both the user specified DDNAMEs and to the standard output data sets.

R-CODE:

0	Data split as requested.
8	No records found to copy.

6.2.109 MSPLIT (OBSOLETE)

FORMAT: MSPLIT

NOTE: This keyword is still available for compatibility with prior releases. It may be removed in a future release. It is strongly recommended that this keyword not be used to accomplish a desired split function. The dynamic allocation control block method should be used instead by specifying the split boundary via the DDAEND keyword.

PARM-LIST: NOT ALLOWED

FUNCTION: To instruct SMFUTIL to check for a change in the month of the output data directed to the output data sets. If a change is detected, SMFUTIL will cease placing data on the primary output data set (SYSUT2) and begin placing data (starting with the first record of the new month) on the alternate output data set (SYSUT3).

NOTE: See the section entitled "INPUT DATA SPLITTING" on page 166 for information about requirements for proper splitting results.

R-CODE:

0	No day change. All output on SYSUT2.
3	Month Change detected. SYSUT3 has data.

6.2.110 MSPLITDP (OBSOLETE)

FORMAT: MSPLITDP

NOTE: This keyword is still available for compatibility with prior releases. It may be removed in a future release. It is strongly recommended that this keyword not be used to accomplish a desired split function. The dynamic allocation control block method should be used instead by specifying the split boundary via the DDAEND keyword.

PARM-LIST: NOT ALLOWED

FUNCTION: To instruct SMFUTIL to check for a change in the month of the output data directed to the duplex data sets. If a change is detected, SMFUTIL will cease placing data on the primary duplex output data set (SYSUT2D) and begin placing data (starting with the new day) on the alternate duplex output data set (SYSUT3D).

NOTE: See the section entitled “INPUT DATA SPLITTING” on page 166 for information about requirements for proper splitting results.

R-CODE: 0 No day change. All duplex output on SYSUT2D.
3 Month Change detected. SYSUT3D has data.

6.2.111 MULTICLEAR

FORMAT: MULTICLEAR

PARM-LIST: NOT ALLOWED.

INVERSE: NOMULTICLEAR

FUNCTION: To instruct SMFUTIL to perform clear processing for all MANx datasets in a DUMPCLEARALL execution simultaneously. This will result in greater overlap of I/O processing and return MANx datasets to the SMF system faster.

NOTE: Due to the asynchronous nature of the processes, messages issued during the clear processing may appear out of order as data sets may not finish clearing in the same order they were started.

R-CODE: Not affected.

6.2.112 NIORATIO

FORMAT:	NIORATIO(integer)
integer	An integer from 1 to 9 digits.
PARM-LIST:	REQUIRED
FUNCTION:	Overrides the NIORATIO value specified in the SMFUDFLT module and instructs SMFUTIL to maintain the specified maximum number of READ operations per NOTE macro during an ARCHINPUT operation. In other words, for each “NIORATIO” READ operations done there will be at least one NOTE macro issued.
NOTE:	<p>The lower the value specified the more accurate the BID values obtained via the NOTE macro will be but the longer the ARCHINPUT operation will take. Higher values result in faster ARCHINPUT operations but correspondingly less accurate BID values. The default value of 25 is a good compromise in most situations.</p> <p>As a safety precaution, each time SMFUTIL sees a date transition in the input data stream, a NOTE macro will be forced regardless of the current count.</p>
CAUTION:	Very low values of NIORATIO will result in extremely long run times for ARCHINPUT executions. Typically, values below the default of 25 will result in only marginally improved BID accuracy and are not cost justifiable.
R-CODE:	Not affected.

6.2.113 NOARCSID

FORMAT:	NOARCSID
INVERSE:	ARCHSID
PARM-LIST:	NOT ALLOWED
FUNCTION:	To instruct SMFUTIL that any requested SYSID specification masks are to be ignored when searching the archive database for records matching the date range(s) specified. All records in the database will be considered valid for selection if they contain data within a requested date range. This specification only has meaning when the archive is being used to locate requested input data. It should only be specified if ALL data in the archive data sets is fully commingled. If desired, NOARCSID may be made the default by specifying ‘ARCHSID=NO’ in the SMFUDFLT module and reassembling it.
R-CODE:	Not affected.

6.2.114 NOLOCATE

FORMAT: NOLOCATE
 INVERSE: LOCATE
 ALIAS: SKPCATCK
 PARM-LIST: NOT ALLOWED.
 FUNCTION: To instruct SMFUTIL that, during the validation of a candidate input tape data set located in the archive, it is NOT to verify that the data set is still cataloged in a system catalog. Normally, SMFUTIL will skip tape data sets that are no longer cataloged on the assumption that they have been scratched by the tape management system. In the event of system problems with the catalog that contains the SMF archive data set names, this may not be a valid test. SMFUTIL may well be required to extract the SMF data necessary to reconstruct the catalog. Skipping the catalog LOCATE will be required and the 'NOLOCATE' keyword accomplishes this.
 R-CODE: Not affected.

6.2.115 NOSYSUT1

FORMAT: NOSYSUT1
 PARM-LIST: NOT ALLOWED.
 FUNCTION: To instruct SMFUTIL that it is to ignore the DDNAME of "SYSUT1" and only consider a DD valid as input if it has at least one character appended to SYSUT1.
 R-CODE: Not affected.

6.2.116 NOTIFY

FORMAT: NOTIFY(userid[(F)],userid,...)
 userid A 1 to 8 character user id to be notified.
 PARM-LIST: At least 1 userid must be coded.
 FUNCTION: To instruct SMFUTIL to notify the specified user(s) when the SMFUTIL execution completes. If the optional "(F)" after a userid, that user will only be send notification if the run return code is 4 or greater.
 A special userid of "OPER" may be coded to cause notification to be sent to the system operator console. The "(F)" may also be included after "OPER".
 NOTE: In order to send a notification to a TSO user, SMFUTIL must be executing in an authorized environment. In addition the TSO user must be receiving MAIL. If the user specified is not logged on, they will receive the message at the next logon as long as NOMAIL is not specified.
 R-CODE: Not affected.

6.2.117 ODTRANGE

FORMAT:	ODTRANGE(ddn-start:end[,ddn-start:end,...])
ddn	A 1 to 8 character DDNAME that is to be limited in the range of date/time record values it will receive.
start	A 1 to 15 character date/time stamp value for the specified DDNAME to begin receiving data. All data prior to this date/time will not be placed to the specified ddname.
end	A 1 to 15 character date/time stamp value for the specified DDNAME to cease receiving data. All data after this date/time will not be placed to the specified ddname.
PARM-LIST:	At least 1 ddn-start:end pair must be coded.
FUNCTION:	<p>Instructs SMFUTIL to allow only data from the starting date/time value to the ending date/time value to be placed to the specified ddname. Multiple ddname/dtrange values may be coded within the parenthesis and multiple ODTRANGE keywords may be specified. Each DDNAME may be specified multiple times with a different range for each specification. The date/time value has the format of 'ccyydddhhmmssst' where:</p> <p>'cc' Has the value 19 to 29 (century value). 'yy' Has the value 00 to 99 (year). 'ddd' Has the value 001 to 366 (Julian day value). 'hh' Has the value of 00 to 23 (hour value). 'mm' Has the value of 00 to 59 (minute value). 'ss' Has the value of 00 to 59 (seconds value). 'th' Has the value of 00 to 99 (tenths and hundredths of a second).</p> <p>Each date/time string may be truncated to less than 15 characters if desired. The range start positions not coded will default to zeros and the range end positions not coded will default to high values.</p>
NOTE:	<p>Caution should be exercised in specifying multiple date/time ranges for the same DDNAME and mixing ODTRANGE with OTIMERANGE in the same execution. SMFUTIL does not validity check the specifications with each other. Each specification is considered independently.</p> <p>The ODTRANGE keyword is only a filter. It does not direct any specific record types to the specified ddname. It only prevents data from being written to the DDNAME unless it is within the specified date/time range. If the object DDNAME is user defined (i.e. not a standard output data set such as SYSUT2), the user must target specific record types to the DDNAME using another keyword such as TYPETODD or CICSSPLIT.</p>
R-CODE:	Not affected.

6.2.118 ORDERCHECK

FORMAT: ORDERCHECK
INVERSE: NOORDERCHECK
PARM-LIST: NOT ALLOWED
FUNCTION: Instructs SMFUTIL to ensure that the current input data set logically follows the previous runs input data set. The gap value is the maximum amount of seconds of the time gap between the final record of the previous run and the first record of the current run. If “gap-sec” is not specified via the “ORDERGAP” keyword, the default will be used. As shipped the default is ORDEROFF (no gap checking) and a gap value of 300 seconds (5 minutes) is preset. In this way the ORDERCHK keyword can turn on order checking without setting the gap value. The default values may be changed by altering the SMFUFLT module and reassembling. See the section titled “MODIFICATION AND CUSTOMIZATION” on page 199 for more information.
NOTE: The “SYSCKPT” data set must be present for the prior run and the current run for the ORDERCHK keyword to be effective. SMFUTIL preserves the date and time of the final record in checkpoint record “LASTREC1”. This record must be available to the next run or order checking is bypassed.
R-CODE: 0 - Input data set order accepted
 8 - Out of order data set sequence detected.

6.2.119 ORDERGAP

FORMAT: ORDERGAP[(gap-sec)]
gap-sec An integer value from 1 to 7 digits.
INVERSE: NOORDER or ORDEROFF
PARM-LIST: OPTIONAL
FUNCTION: Sets the gap value used by the ORDERCHK function. If ORDERGAP is specified in a run the ORDERCHK function will automatically be assumed, whether or not it is specified. If a “gap-sec” parameter is not specified the default will be assumed. As shipped the default is value is 300 seconds (5 minutes). The default value may be changed by altering the SMFUFLT module and reassembling. See “MODIFICATION AND CUSTOMIZATION” on page 199 for more information.
NOTE: The “SYSCKPT” data set must be present for the prior run and the current run for the ORDERCHK keyword to be effective. SMFUTIL preserves the date and time of the final record in checkpoint record “LASTREC1”. This record must be available to the next run or order checking is bypassed.
R-CODE: 0 - Input data set order accepted
 8 - Out of order data set sequence detected.

6.2.120 OSYNADOFF

FORMAT: OSYNADOFF

PARM-LIST: NOT ALLOWED.

FUNCTION: To instruct SMFUTIL to disable the SYNAD exits embedded in the output DCB's. These exits intercept I/O errors and allow SMFUTIL to continue to execute after such an error has occurred (in certain situations). This parameter is used for debugging purposes only. It should only be invoked at the request of WOVEN SOFTWARE support personnel.

R-CODE: Not affected.

6.2.121 OTIMERANGE

FORMAT: OTIMERANGE(ddn-[tstart][:tend][,ddn-tstart:tend,..])

ddn A 1 to 8 character DDNAME that is to be limited in the range of record time values it will receive.

tstart A 2 to 8 character time stamp value for the specified DDNAME to begin receiving data. All data prior to this time, within each day, will not be placed to the specified ddname.

end A 2 to 8 character time stamp value for the specified DDNAME to cease receiving data. All data after this time, within each day, will not be placed to the specified ddname.

PARM-LIST: At least 1 ddn-tstart:tend pair must be coded.

FUNCTION: Instructs SMFUTIL to allow only data from the starting time value to the ending time value to be placed to the specified ddname. Multiple ddname/trange values may be coded within the parenthesis and multiple OTIMERANGE keywords may be specified. Each DDNAME may be specified multiple times with a different range for each specification. The time value has the format of 'hhmmsssth' where:

'hh' Has the value of 00 to 23 (hour value).

'mm' Has the value of 00 to 59 (minute value).

'ss' Has the value of 00 to 59 (seconds value).

'th' Has the value of 00 to 99 (tenths and hundredths of a second).

Each time string may be truncated to less than 8 characters if desired but the minimum is 2 digits ('hh' specification). The range start positions not coded will default to zeros and the range end positions not coded will default to high values.

NOTE: Caution should be exercised in specifying multiple time ranges for the same DDNAME and mixing OTIMERANGE with ODTRANGE in the same execution. SMFUTIL does not validity check the specifications with each other. Each specification is considered independently.

The OTIMERANGE keyword is only a filter. It does not direct any specific record types to the specified ddname. It only prevents data from being written to the DDNAME unless it is within the specified time range. If the object DDNAME is user defined (i.e. not a standard output data set such as SYSUT2), the user must target specific record types to the DDNAME using another keyword such as TYPETODD or CICSSPLIT.

R-CODE: Not affected.

6.2.122 OUTPUTDETAIL

FORMAT:	OUTPUTDETAIL
ALIAS:	ODETAIL
INVERSE:	NOOUTPUTDETAIL
PARM-LIST:	NOT ALLOWED
FUNCTION:	Instructs SMFUTIL to produce a detailed statistics report for each output data set. Each line of the output report will represent one record type. Data produced for each type will be: <ol style="list-style-type: none"> 1. Record Type Number. 2. Number of records this output data set. 3. Low date/time encountered this output data set. 4. High date/time encountered this output data set. 5. Minimum record length this output data set. 6. Maximum record length this output data set. 7. Average record length this output data set.
NOTE:	Average record lengths are calculated by dividing the sum of all lengths encountered for a record type by the total record count. If the accumulation of record lengths becomes too large, the calculation will be suspended and '*****' will be placed in the output field.
R-CODE:	Not affected.

6.2.123 OUTPUTLIMIT

FORMAT:	OUTPUTLIMIT(integer)
integer	An integer from 1 to 9 digits.
PARM-LIST:	REQUIRED
FUNCTION:	Instructs SMFUTIL to limit the total output record count to a specific value. This is useful is only a small sample of a particular data type is required.
NOTE:	An OUTPUTLIMIT specification of 0 (zero) can be useful after an aborted execution that used MOD data sets with checkpoint protection. It would cause the output MOD data set checkpoint records to be acted upon and return the output data sets to their original state without actually adding any data to the outputs. This can be helpful in getting back to where you were before the aborted job occurred.
R-CODE:	Not affected.

6.2.124 OUTPUTDATADISTRIBUTION

FORMAT: OUTPUTDATADISTRIBUTION[(type)]

ALIAS: ODIST

PARM-LIST: OPTIONAL

type A keyword of ALL, RECTYPE or BOTH indicating the type of percentage to be reported. RECTYPE is the default. It instructs SMFUTIL to calculate the percentage based on only the specific type of record sum for the 24 hour period. Each record type line will add to 100% for that 24 hour period. The optional report type ALL indicates the percentage is to be calculated base on the total sum of ALL record types for the 24 hour period. Note that to produce both reports, the keyword BOTH may be specified in the parameter list. For example, ODIST(BOTH) would request both reports.

FUNCTION: To instruct SMFUTIL to produce a Data Distribution report for each output data set. This report shows the percentage of each record type output for each hour of the day out of the total of that record type or all record types for that data set in the 24 hour period. If more than one day's worth of data is processed, the percentage represents that hour for all days processed for the data set.

NOTE: This is a moderately high overhead report request. It cannot be requested by default. It should not be requested when large volumes of data will be processed. An additional 25K of region will be required for each input data set processed. Requesting both reports does not significantly increase the overhead.

R-CODE: Not affected.

6.2.125 POSTEXIT

FORMAT: POSTEXIT(exit-name)

exit-name A 1 to 8 character name of a user supplied load module exit for post processing records. The load module must be available in the LINKLST or on LPALIB, JOBLIB, or STEPLIB.

PARM-LIST: REQUIRED.

FUNCTION: Instructs SMFUTIL to take a user supplied post processing exit for each record after the record has completed SMFUTIL processing. Each record is pass to the POST exit after it has been written by SMFUTIL to the output data set(s) or discarded as not meeting filtering criteria (i.e. record type, system id, etc.). See the section titled "USER EXITS" on page 223 for more information on implementing user exits.

R-CODE:

0	Load module located.
8	Specified load module not found.

6.2.126 PREVIOUSMONTH

FORMAT: PREVIOUSMONTH
ALIAS: LASTMONTH
PARM-LIST: NOT ALLOWED.
FUNCTION: To instruct SMFUTIL to copy data for the previous month to the output data set.
EXAMPLE: If PREVIOUSMONTH had been specified in a run executed on Monday, December 30, 2002, the dates selected would have been Friday, November 1, 2002, through Sunday, November 30, 2002.
NOTE: When PREVIOUSMONTH is specified, an automatic default specification of SHORTCPY is assumed. The GOTOEOF or NOSTOP keyword may be used, after the PREVIOUSMONTH keyword, to override this default.
R-CODE: 0 Requested data copied.
 8 No data within dates for PREVIOUSMONTH found on input data set.

6.2.127 PREEXIT

FORMAT: PREEXIT(exit-name[,exit-name,...])
exit-name A 1 to 8 character name of a user supplied load module exit. The load module must be available in the LINKLST or on LPALIB, JOBLIB, or STEPLIB.
PARM-LIST: At least one load module name required.
FUNCTION: Instructs SMFUTIL to take a user supplied preprocessing exit for each record input. If more than one exit load module name is specified, the modules are called in the order given. If more names are required than will fit on one card, multiple PREEXIT control keywords may be used. See the section titled "USER EXITS" on page 215 for more information on implementing user exits.
NOTE: This exit point is after record construction completion but before any validation of the record by SMFUTIL. It is the user's responsibility to ensure that the record is valid for the exit's processing purposes. An ESTAE exit and/or SPIE exit is strongly recommended.
R-CODE: 0 Load module located.
 8 Specified load module not found.

6.2.128 PREVIOUSWEEK

FORMAT: PREVIOUSWEEK

ALIAS: LASTWEEK, PREVWEEK

PARAM-LIST: NOT ALLOWED.

FUNCTION: To instruct SMFUTIL to copy data for the previous week to the output data set. Data from the previous Saturday through last Friday is selected.

EXAMPLE: If PREVWEEK had been specified in a run executed on Monday, December 30, 2002, the dates selected would have been Saturday, December 21, 2002, through Friday, December 27, 2002.

NOTE: When PREVWEEK is specified, an automatic default specification of SHORTCPY is assumed. The GOTOEOF or NOSTOP keyword may be used, after the PREVWEEK keyword, to override this default.

R-CODE: 0 Requested data copied.
8 Dates for PREVWEEK not found on input data set.

6.2.129 PRINT

FORMAT: PRINT[(limit)]

limit An optional integer value from 1 to 9 digits.

PARAM-LIST: OPTIONAL.

FUNCTION: To instruct SMFUTIL to produce a printed listing of records selected for output. A '//SYSOUT DD SYSOUT=A' statement must be specified in the execution JCL to receive the output of the print function. The printed output will contain both a hex dump and a printable text listing of each record selected. Caution should be exercised in use of the PRINT function as a large amount of printed output may be generated if a large amount of records are selected for output unless the optional 'limit' parameter is used. If a limit value is specified, printing will be suspended after the specified number of records has been printed but normal processing will continue unless a print only execution was requested (no data movement in progress). If print only requested, SMFUTIL will terminate after the requested number of records have been printed. The SIMULATE function can be used to do a print only execution without creating an output data set. If PRINT is specified and the SYSUT2 DD statement is missing, a print only execution will be assumed and SIMULATE will be defaulted.

R-CODE: Not affected.

6.2.130 PROCESS

FORMAT: PROCESS(process-name)

process-name A 1 to 16 character process name to be associated with this execution of SMFUTIL.

INVERSE: NOPROCES

PARM-LIST: REQUIRED.

FUNCTION: Instructs SMFUTIL to verify that the correct checkpoint data set is in use. The process name and the executing system id (SMF id) are recorded in the \$PROCESS checkpoint record for future verification. If the \$PROCESS checkpoint record is already present in the checkpoint data set, it's contents must match the supplied name for the run to proceed.

If PROCESS=YES is specified in the SMFUDFLT table and the PROCESS keyword is not specified, the process name defaults to the content of the 16 byte TIOCSTEP field in the TIOT.

If PROCESS=NO is specified in the SMFUDFLT table, \$PROCESS is not verified unless the PROCESS keyword is specified at execution time with a process name.

If NOPROCES is specified at execution, it overrides the PROCESS=YES specification in the defaults table and suppresses checking of the \$PROCESS checkpoint record.

R-CODE: 0 \$PROCESS added or matched current name and system.
8 Existing \$PROCESS record did not match current name.

6.2.131 PROCSYS

FORMAT: PROCSYS(sighs-id)

sys-id A 1 to 4 character processing system id to override the system name used in conjunction with the PROCESS name for recording in the \$PROCESS checkpoint record.

PARM-LIST: REQUIRED.

FUNCTION: Instructs SMFUTIL to use the specified system id instead of the execution system id in recording and verifying the \$PROCESS checkpoint record. This allows a system independent process to be run on any available system and still use the related checkpoint data set.

NOTE: PROCSYS is ignored for a run specifying DUMPCLEARALL. The executing system SMF id will always be used. This is to prevent a DUMPCLEARALL checkpoint data set from one system from being inadvertently used on another system.

R-CODE: Not affected.

6.2.132 RECEXIT

FORMAT: RECEXIT(rrr-exit-name[,rrr-exit-name,...])

rrr A valid SMF record type number in the range of 1 to 255.

exit-name A 1 to 8 character name of a user supplied load module exit. The load module must be available in the LINKLST or on LPALIB, JOBLIB, or STEPLIB.

PARM-LIST: At least one record number and load module name combination required.

FUNCTION: Instructs SMFUTIL to take a user supplied exit routine for each record, of the type specified, that is chosen for output processing. If more than one exit load module name is specified for the same record type, the modules are called in the order given. If more record/exit names combinations are required than will fit on one card, multiple RECEXIT control keywords may be used. See the section titled "USER EXITS" on page 215 for more information on implementing user exits.

EXAMPLE: RECEXIT(30-SCAN30,26-SCAN26) would cause load module SCAN30 to be branched to each time a type 30 record was encountered and load module SCAN26 to be branched to each time a type 26 record was encountered.

R-CODE: 0 All exit load module located.
8 Specified load module not found.

6.2.133 REPORTFIXED

FORMAT: REPORTFIXED

INVERSE: REPORTDYNAMIC

PARM-LIST: NOT ALLOWED.

FUNCTION: Instructs SMFUTIL to use a fixed format report. All columns will remain in the same location and of the same width for any post processing of the report that may be required.

R-CODE: Not affected.

6.2.134 RVALEXIT

FORMAT: RVALEXIT(exit-name[,exit-name,...])

exit-name A 1 to 8 character name of a user supplied load module exit. The load module must be available in the LINKLST or on LPALIB, JOBLIB, or STEPLIB.

PARM-LIST: At least one record validation exit load module name required.

FUNCTION: Instructs SMFUTIL to take a user supplied exit routine for each record found prior to the record being validated by SMFUTIL. This exit point can be used to pre-validate and/or modify records before SMFUTIL accepts them for internal validation. Using this exit point, it is possible to create SMF records out of other types of variable length records. If more than one exit load module name is specified, the modules are called in the order given. If more exit names are required than will fit on one card, multiple RVALEXIT control keywords may be used. See the section titled "USER EXITS" on page 215 for more information on implementing user exits.

EXAMPLE: RVALEXIT(FIXUP) would cause load module FIXUP to be branched to each time a new record was constructed.

R-CODE: 0 All exit load module(s) located.
8 Specified load module not found.

6.2.135 SAT/SUN/MON/TUE/WED/THU/FRI

FORMAT: SAT
 SUN
 MON
 TUE
 WED
 THU
 FRI

PARM-LIST: NOT ALLOWED

FUNCTION: Instructs SMFUTIL to discard (filter out) data created on days other than the specified day(s) for processing. One or more days may be specified. If one or more days are specified in the same run with WEEKEND and/or WORKDAY the last specification encountered (i.e. one or more days or WORKDAY or WEEKEND) will be honored.

R-CODE: 0 Data for specified days selected
 8 No data found for specified days.

6.2.136 SEARCHARCHIVE

FORMAT: SEARCHARCHIVE

ALIAS: SEARCH

PARM-LIST: NOT ALLOWED

FUNCTION: To instruct SMFUTIL that the ARCHIVE database is to be searched for data to meet the date/time spans requested even though a user specified input data set was supplied (via JCL or DDA block). Note that any user supplied data set will be input first and any ARCHIVE supplied data set(s) input after the user supplied input.

NOTE: Use of the SEARCHARCHIVE keyword and a user supplied input data set may result in duplicated data and/or erroneous early termination of the execution because the user data set is being read out of order with the rest of the data. Note that the ARCHIVE supplied data set(s) will be allocated AFTER the user supplied data set. This means the user supplied data set(s) are input first. If the user supplied data set(s) contains data outside the scope of the supplied explicit or implicit DATE/TIME range, it will be skipped. The GOTOEOF keyword should be specified to prevent early termination of the run in the event that the user supplied data set(s) contain data after the requested date range. A post process SORT step may then be necessary to ensure that the data is in the desired sequence.

R-CODE: Not affected.

6.2.137 SETNAME

FORMAT:	SETNAME(set-name[,setname])
set-name	The set-name of archive records to be used to select archive data volumes required to satisfy the requested date/time ranges for this execution.
PARM-LIST:	REQUIRED
FUNCTION:	To supply the ARCHIVE record identifying set name to be used to retrieve data from the archive. When recording data to the archive database, SMFUTIL places a unique set name into each record in the archive. This set name is derived from the output file the data is written to. For output to SYSUT2 (or any split from it such as SYSUT3), the set name is "NORMAL" this is the default set for retrieval from the archive. For output to SYSUT2D (or any split from it such as SYSUT3D) the set name is "DUPLEX". If the "ARCHUSER" keyword is supplied while data is being written to user defined data sets (i.e. TYPETODD specifications) the first five characters of the user supplied DDNAME will be used as the set name to record data in the archive. See the section titled "ARCHIVING SMF DATA" on page 175 for more information on this subject. During data retrieval, SMFUTIL will default to a set name of "NORMAL" for data retrieval from the archive. The ARCHSET keyword can be used to cause SMFUTIL switch to the "DUPLEX" records or to consider only data sets created by specific user filtering requests. Only one set name may be in effect per execution. If specified more than once per execution, only the last specification will be used.
NOTE:	If more than one set name is specified for retrieval, output data may not be in sorted order and may also contain multiple copies of the same data depending on the content of the different sets. Caution should be exercised in the selection of multiple sets to ensure the correctness of the selected data. For example, if two different sets are selected that contain the same date/time range but different record types, the output data will be the correct record types but will not be in sorted order.
R-CODE:	Not affected.

6.2.138 SCRATCHRETRY

FORMAT:	SCRATCHRETRY(interval,count)
interval	An amount of time in seconds that is to be waited before retry. Range is 0 to 60.
Count	The number of times retry is to be attempted before giving up. Range is 1 to 60.
PARM-LIST:	REQUIRED
FUNCTION:	Instructs SMFUTIL to retry certain failed scratch attempts made on behalf of the DELETE parameter on the GDGINPUT or CATINPUT keywords.. To be retried, the status code returned from the failed scratch SVC must indicate certain conditions exist. For example, if an enqueue failed to gain control of the data set (i.e. it is in use by another address space) a retry can be attempted in the hope that the data set will become available. The maximum
NOTE:	To prevent any retries, set the 'interval' parameter to '0'.
R-CODE:	Not affected.

6.2.139 SCLRTHLD

FORMAT: SCLRTHLD(num)

num A percentage value from 0 to 100.

PARM-LIST: A single value is required.

FUNCTION: Instructs SMFUTIL at or above what percent full a MANx dataset should be cleared with the more efficient full clear method rather than SHORTCLEAR. During a DUMPCLEARALL execution SMFUTIL will automatically suppress a requested SHORTCLEAR in favor of a full clear for each MANx dataset that is at least this percent full.

R-CODE: Not affected.

6.2.140 SDB

FORMAT: SDB

INVERSE: NOSDB

PARM-LIST: NOT ALLOWED.

FUNCTION: To instruct SMFUTIL that SDB (System Determined Block size) is active in the operating system and SMFUTIL does not need to choose a BLKSIZE value if a '0' value is found in a JCL allocated data set or an omitted BLKSIZE value is found in a DDA block allocated data set.

NOTE: This parameter disables BLKSIZE optimization by SMFUTIL. The installation must ensure that a valid and appropriate BLKSIZE value will be supplied by the operating system for ALL new output data sets produced by SMFUTIL. An alternative would be to leave NOSDB in place and specify BLKSIZE(0) for each DDA allocated output data set that SMS is going to choose a BLKSIZE value for. This will enable SMS Extended Sequential data set creation for specific data sets but still leave SMFUTIL BLKSIZE optimization active for other output data sets. See "MODIFICATION AND CUSTOMIZATION" on page 199 for more information on changing the default value of SDB.

R-CODE: Not affected.

6.2.141 SHORTCLEAR

FORMAT: SHORTCLEAR

ALIAS: SHORTCLR

INVERSE: NOSHORTCLEAR

PARM-LIST: NOT ALLOWED.

FUNCTION: To instruct SMFUTIL to use a VSAM UPDATE process instead of a SEQUENTIAL overwrite process to clear MANx data sets.

NOTES: When SHORTCLEAR is requested, instead of sequentially overwriting the entire SMF MANx data set with clearing blocks, SMFUTIL uses a Control Interval UPDATE process with RBA addressing to overlay data blocks with clear blocks. VSAM Improved Control Interval access (ICI) processing is used. During normal DUMPCLEARALL processing, only control intervals that physically contained SMF data input to the run will be updated. During a standard CLEAR process, all control intervals of the SMF MANx data set being cleared will be updated to a clear status.

The SHORTCLEAR request will automatically be suppressed by the SCLRTHLD threshold for MANx datasets full or nearly full. The full clearing process is faster when the majority of the MANx dataset requires clearing. Local testing may be necessary to determine what SCLRTHLD percentage results in the lowest overall run time for clearing.

Test have shown that SHORTCLEAR will usually result in lower wall clock times for any given clearing process (both DUMPCLEARALL and CLEAR functions) but an increase of the total CPU time required, the Start I/O rate, and the total EXCP's issued will be seen. Substantial improvements will be seen in DUMPCLEARALL runs that do SMF MANx switching where at least one MANx data set being dumped and cleared contains a relatively small amount of data. This is normally the case when SMFUTIL performs an additional switch to get off of the primary data set (i.e. MAN1LEAV is specified).

SHORTCLEAR is enabled by default, as shipped. The installation must decide whether to leave it turned on or not.

RESTRICTIONS: SHORTCLEAR cannot be used to clear an un-initialized SMF MANx data set (i.e. one that has just been defined via IDCAMS and never used by SMF). This is because the VSAM UPDATE process cannot be used to "LOAD" an empty data set. It is only valid to replace records within an existing full data set. If you wish to initialize new SMF MANx data sets with SMFUTIL prior to placing them into production use, specify:

"CLEAR NOSHORTCLEAR"

as control parameters in the SYSIN stream.

R-CODE: Not affected.

6.2.142 SHORTCOPY

FORMAT: SHORTCOPY
ALIAS: SHORTCOPY
INVERSE: NOSHORT or GOTOEOF
PARAM-LIST: NOT ALLOWED.
FUNCTION: To instruct SMFUTIL to terminate the copy operation at the end of the DATE/TIME range even if the input data is not sorted.
NOTE: If SHORTCPY is used (or defaulted in the SMFUDFLT module) and the first record on the input data set is past the requested date/time specification, SMFUTIL will terminate with no data copied. This quick termination feature can be used in conjunction with the IVSTART specification to step backwards through a multiple volume data set to find the volume that contains the requested data without spanning many reels of tape. Caution should be exercised in using this technique when the required time frame may be split across two volumes.
R-CODE: Not affected.

6.2.143 SIDSPLIT

FORMAT: SIDSPLIT(sid-ddname[,sid-ddname,...])
sid The SMF identifier for a system.
ddname An output file name the selected records are to be targeted to.
PARAM-LIST: At least one sid-DDNAME pair required.
FUNCTION: Instructs SMFUTIL to copy records to selected outputs based upon their SMF SYSTEM ID. If the content of the SMF system id value in the record matches the specified 'sid' string, the record will be written to file associated with the specified DDNAME. If the sid string is directly followed by the '-' separator character, it will be assumed that the sid specification is a prefix mask. Any record containing an sid that matches the specification for its length will be considered selected. If a blank follows the sid string (between it and the '-') it will force the sid specification to be a full match for the sid in each record. This syntax structure allows one specification to select multiple SID's for selection to one output file.
NOTE: If a record is found with an 'sid' value not specified in a SIDSPLIT statement, it will be written to the standard output data sets. By default, records selected to the target SIDSPLIT DDNAMEs via their 'sid' values will not be written to the standard output files. The keyword COPYVBTH may be specified to cause the selected records to be written to both the user specified DDNAMEs and to the standard output data sets.
R-CODE:

- 0 Data split as requested.
- 8 No records found to copy.

6.2.144 SIMULATE

FORMAT: SIMULATE

PARM-LIST: NOT ALLOWED.

FUNCTION: To instruct SMFUTIL that this is a simulation run. No data will be copied and the output data sets are not used (or required in the JCL). This is useful for scanning an input data set for date ranges, time spans, record counts, Userid's, Job names, etc.

R-CODE: Not affected.

6.2.145 SKIPHDT

FORMAT: SKIPHDT

PARM-LIST: NOT ALLOWED.

INVERSE: NOSKIPHI

FUNCTION: To instruct SMFUTIL to skip as invalid (ignore and delete) input records who's date value is greater (higher) than the current system date. This was primarily designed as a way to suppress Y2K testing records from the main SMF production data stream. It has been retained because some OEM products still produce such records in some installations and they can cause may problems if left in the stream.

R-CODE: Not affected.

6.2.146 SKIPOLD

FORMAT: SKIPOLD

PARM-LIST: NOT ALLOWED.

FUNCTION: To instruct SMFUTIL that it is to bypass input until the date/time values in the input record exceed the date/time value of the last record processed in the previous execution. This keyword depends on the availability of the SYSCKPT data set. See the section entitled "BYPASSING PREVIOUSLY PROCESSED DATA" on page 163.

R-CODE: Not affected.

6.2.147 SMFIDCHECK

FORMAT:	SMFIDCHECK[(rc)]
INVERSE:	NOSMFIDCHECK
rc	The return code to be set if a mismatched SMF ID is encountered. Valid values are 0, 4, 8, or ABEND. A specification of ABEND will terminate the run immediately when a mismatch is found. All other values will allow the run to continue normally with the specified return code being issued as the minimum return code when the run completes.
PARM-LIST:	Optional. Default value is 4.
FUNCTION:	To instruct SMFUTIL to check for consistency of the SMF ID's in the data stream and what return code to set if an SMF ID is encountered in an SMF record that is different from the SMF ID from the first record input or that retained in the \$PROCESS checkpoint record, if the checkpoint dataset is present. This is intended to verify that a particular CKPTDB checkpoint PDS data set is only used to process data from a particular system and/or data is not contaminated with invalid system id's from OEM records.
NOTE:	A checkpoint data set must be present for SMFIDCHECK to be functional for checking the \$PROCESS keyword.
NOTE:	During a DUMPCLEARALL execution, SMFIDCHECK will detect and report the first mismatch in the SMF ID in the data stream but the requested return code to be set will be ignored as data from the MANx datasets is assumed to be valid at all times in order for correct dumping and clearing to occur.
R-CODE:	Affected as directed by user when a mismatched ID is found except unaffected for DUMPCLEARALL.

6.2.148 SORTTOLERANCE

FORMAT:	SORTTOLERANCE[(delta[,units])]
ALIAS:	SORTTOL
delta	The number of minutes of ordering variance to tolerate. Valid range is 0 to 999.
units	A keyword indicating the unit value of the delta. Valid units are MINUTES, SECONDS, and HUNDREDTHS. These can be abbreviated to MIN, SEC, and HUN. If units values is omitted, MINUTES will be defaulted.
PARM-LIST:	At lease 'delta' value is required.
FUNCTION:	To instruct SMFUTIL to tolerate a certain amount of variance between the time stamp in SMF records before flagging the set out of sort sequence. The default is 15 minutes.
NOTE:	Caution should be exercised in specifying "0" (or very low values of hundredths of a second) as the sort tolerance. SMF data can routinely be out of true sort by fractions of a second. In addition, very small values (less than 1 second) can adversely impact the functioning of DATESCAN and DELDUPRV.
R-CODE:	Not affected.

6.2.149 SPIEOF

FORMAT: SPIEOF

PARM-LIST: NOT ALLOWED.

FUNCTION: To instruct SMFUTIL to turn off the error correcting program interrupt exit. This parameter is used for debugging purposes only. It should only be invoked at the request of support personnel.

R-CODE: Not affected.

6.2.150 SPINBACKUP

FORMAT: SPINBACKUP(numblks)

numblks A integer value of 2 to 99 indicating the number of data blocks to be backed up after high speed spin finds a target block.

PARM-LIST: REQUIRED

FUNCTION: To instruct SMFUTIL how many blocks are to be reprocessed after high speed spin completes it's target block search. The default value of 3 set by the SPINBKUP parameter of the SMFUDFLT module is usually sufficient for most data. Larger values may occasionally be needed if out of order OEM record types routinely cause the spin down process to skip required data.

R-CODE: Not affected.

6.2.151 SRBUFF

FORMAT: SRBUFF(numbuff)

numbuff A integer value of 0 or 2 to 99 indicating the number of buffers to be supplied for sequential read operations on SYSUT1xx files. A specification of "0" indicates dynamic buffering where the system determines the optimum buffering level.

PARM-LIST: REQUIRED

FUNCTION: To instruct SMFUTIL to use the indicated number of buffers during sequential read operations.

R-CODE: Not affected.

6.2.152 SUBSET

FORMAT: SUBSET(subset[,subset,...])

subset The defined name of a class of SMF records.

PARAM-LIST: At least one SUBSET name required.

FUNCTION: To instruct SMFUTIL to copy all records associated with a previously defined subset name. This is a way of logically grouping records together that have certain traits in common. SMFUTIL comes with the following subset definitions pre-defined:

SYS	System oriented records.
MACHINE	Hardware error oriented records.
AUXSTOR	Aux storage records.
RMF	IBM RMF Records.
CMF	Boole & Babbage CMF Records.
DSNAME	Records containing data set names.
JOB	Job oriented records.
JES	JESx oriented records.
USER	IBM defined user record numbers (128-255)
IMS	IBM defined IMS records.
TSO	Time Sharing Option oriented records.
USID	Resource user oriented records.
SECURITY	RACF oriented records.
IO	Input/Output data set oriented records.
DATA	Data set access and activity records.
VSAM	Activity records for VSAM files.

NOTE: These pre-defined subset names are located in CSECT SMFSSTBL. A source code module (by the same name) is located in the INSTLIB data set. The user may modify the existing definitions and/or add new ones. The existing definition names should not be deleted as other control keywords in SMFUTIL use some of the definitions for classifying records. See the section titled “DEFINING SUBSETS” on page 208 for further details on the modification procedure.

R-CODE: Not affected.

6.2.153 SUBTYPE

FORMAT:	SUBTYPE(s1[,s2,s3,...,sn])
s1,s2,..	A record subtype value (0-999999).
PARAM-LIST:	At least one record subtype value is required.
FUNCTION:	<p>To instruct SMFUTIL to accept only specific record subtype values for output. All records that are considered as valid for record subtype selection will have their internal subtypes compared against the list specified. If a match is found, the record will be selected. If no match is found, the record will be discarded. All records that are selected for input (explicitly via INCLUDE, INONLY, SUBSET, or implicitly via EXCLUDE or EXONLY) but are not valid for subtype selection will be considered valid and will be selected.</p> <p>The SUBTYPE keyword and any specification of record subtype values within a TYPETODD keyword are mutually exclusive. They may not both be used within the same execution of SMFUTIL.</p>
EXAMPLE:	SUBTYPE(4,5)
NOTE:	<p>SUBTYPE is designed to be as a filter when only one record type is being considered. The SUBTYPE specification should only be used in conjunction with a specific INCLUDE specification so as to select the specific record type to be used for subtype selection (i.e. record type 30). If SUBTYPE is used when large amounts of record types are being selected, the content of the SYSUTxx output data set(s) may be unpredictable as not all subtype values are applicable to all records that have subtype capability. If subtype subsets are desired during processing of large amounts of input records, use the TYPETODD specification in conjunction with COPYBOTH to place selected record type/subtype combinations to a subset data set.</p>
EXAMPLE:	INCLUDE(30) SUBTYPE(5)
	This control statement would select only the type 5 subtypes of the type 30 records.
R-CODE:	Not affected.

6.2.154 SUBUFF

FORMAT:	SUBUFF(numbuff)
numbuff	A integer value of 0 or 2 to 99 indicating the number of buffers to be supplied for sequential write operations on user defined output files. A specification of "0" indicates dynamic buffering where the system determines the optimum buffering level.
PARAM-LIST:	REQUIRED
FUNCTION:	To instruct SMFUTIL to use the indicated number of buffers during sequential write operations to user defined files (other than the normal SYSUT2? or SYSUT3? files).
R-CODE:	Not affected.

6.2.155 SUCLEAR

FORMAT: SUCLEAR
 INVERSE: IBMCLEAR or NOSUCLEAR
 PARM-LIST: NOT ALLOWED.
 FUNCTION: To instruct SMFUTIL to use the SAFE CLEAR method for clearing MANx datasets. This prevents SMF from reusing partially cleared MANx datasets in the event of a system outage during a CLEAR process.
 R-CODE: Not affected.

6.2.156 SWBUFF

FORMAT: SWBUFF(numbuff)
 numbuff A integer value of 0 or 2 to 99 indicating the number of buffers to be supplied for sequential write operations on SYSUT2? and SYSUT3? files. A specification of “0” indicates dynamic buffering where the system determines the optimum buffering level.
 PARM-LIST: REQUIRED
 FUNCTION: To instruct SMFUTIL to use the indicated number of buffers during sequential write operations to the normal output data sets.
 R-CODE: Not affected.

6.2.157 SYMBOLS

FORMAT: SYMBOLS
 INVERSE: NOSYMBOLS
 PARM-LIST: NOT ALLOWED.
 FUNCTION: To instruct SMFUTIL to use (SYMBOLS) or disallow use (NOSYMBOLS) of the system symbolic substitution utility ASASYMBM. If either keyword is used to override the setting of the “SYMBOLS=” parameter in the SMFUDFLT module it should be on a line by itself as the first control statement in the input stream. As shipped, symbolic substitution is disabled (SYMBOLS=NO).
 R-CODE: Not affected.

6.2.158 SYNADOFF

FORMAT: SYNADOFF
 PARM-LIST: NOT ALLOWED.
 FUNCTION: To instruct SMFUTIL to disable the SYNAD exits embedded in the input DCB's. These exits intercept I/O errors and allow SMFUTIL to continue to execute after such an error has occurred. This parameter is used for debugging purposes only. It should only be invoked at the request of WOVEN SOFTWARE support personnel.
 R-CODE: Not affected.

6.2.159 SYSID

FORMAT: SYSID(smfid[,smfid,...])

smfid One to four character System SMFID or SMFID prefix or mask.

PARM-LIST: At least 1 smfid required.

FUNCTION: To instruct SMFUTIL to select only records produced by the specified system or systems. If less than four characters are specified the id will be used as a mask and all id's encountered with the prefix mask will be selected. If a specified id of less than four characters is desired, the id should be right padded with blanks to a length of four characters in total. If any position contains an asterisk (*) that position will be treated as a and any character will be accepted in that position. For example a specification of '*083' would match 'A083' and 'B083' equally but not 'C090'.

NOTE: If the first 'smfid' specified is &SID, it will be replaced by the executing system's ID. This allows selection of records produced by the executing system only without knowing the system's ID beforehand.

If the first 'smfid' specified is &DID, it will be replaced by the system ID contained in the first input record. This allows input to be limited to a single system without knowing the system ID beforehand.

R-CODE: 0 -Data copied.
4 -No records found for system(s).

6.2.160 SYSTYPE

FORMAT: SYSTYPE(ttt)

ttt A three character operating system identifier.

PARM-LIST: REQUIRED.

FUNCTION: To instruct SMFUTIL to select records produced by a particular operating system type. Supported type definitions are SP2, SP3, SP4 SP5, ESA, MVS, MXA, VS1, VS2, and SVS.

R-CODE: 0 Data copied.
8 No data found for system type.

6.2.161 TAPEONLY

FORMAT: TAPEONLY

PARM-LIST: NOT ALLOWED.

FUNCTION: To instruct SMFUTIL that, during a search of the ARCHIVE database for input data set selection, only data sets that reside on tape are to be considered for selection. All data set marked as residing on disk devices will be bypassed.

R-CODE: Not affected.

6.2.162 TERMEXIT

FORMAT:	TERMEXIT(exit-name)
exit-name	A 1 to 8 character name of a user supplied load module exit. The load module must be available in the LINKLST or on LPALIB, JOBLIB, or STEPLIB.
PARM-LIST:	A load module name is required.
FUNCTION:	Instructs SMFUTIL to take a user supplied termination exit routine after all data has been processed and prior to terminating SMFUTIL. See the section titled "USER EXITS" on page 215 for more information on the implementation and use of user exits. If the TERMEXIT keyword is specified more than once, the last one encountered will be used.
R-CODE:	Not affected.

6.2.163 TERSE

FORMAT:	TERSE
INVERSE:	VERBOSE (or NOTERSE)
PARM-LIST:	NOT ALLOWED.
FUNCTION:	To instruct SMFUTIL to produce output listing information for only records types actually selected for output processing. Input record types with an output count of zero will not be listed.
NOTE:	TERSE and VERBOSE are moot (ignored) if the LISTALL keyword is specified as all record types from 0 to 255 will be listed regardless of input or output record count.
R-CODE:	Not affected.

6.2.164 TEXTSCAN

FORMAT:	TEXTSCAN(mask-string[,mask-string,...])
mask-string	A 1 to 44 character string of text to be searched for in each selected record. A masking character of an '*' may be included in the string to indicate any character in that position is acceptable. Note that blanks are acceptable within the string or at the end of the string.
PARM-LIST:	At least one string of one or more characters is required.
FUNCTION:	Instructs SMFUTIL to search all selected records for their entire length for matches to the specified string mask(s). If a match is found the record continues as selected. If a match is not found, the record is discarded. If more than one mask string is specified, a record matching to any mask string will cause the record to continue as selected.
NOTE:	<p>The TEXTSCAN function is an extremely high overhead operation. To optimize performance, avoid including record types that you are sure will not contain any matches. For example, if you are searching for a data set qualifier such as 'SYS1.' you should also specify SUBSET(DSNAME) so that only record types that contain data set names are considered. You should also INCLUDE any user record types that may also contain data set names and EXCLUDE any record types you are not concerned with.</p> <p>For performance reasons, avoid using a masking character ('*') as the first character of the string. It serves no purpose and slows down the search logic.</p>
R-CODE:	Not affected.

6.2.165 THSUFFIX

FORMAT: THSUFFIX

PARM-LIST: NOT ALLOWED.

FUNCTION: To instruct SMFUTIL to produce an additional suffix on a time suffix. This addition has the format “.Hth” where ‘th’ is the tenths and hundredths of a second value from the time value used to generate the suffix. This is useful to ensure unique suffixes when very short spans of data may be processed consecutively. The full suffix value will be “.Thhmmss.Hth”. The data set name value used must be short enough to allow the extended suffix to be appended.

R-CODE: Not affected.

6.2.166 TIME

FORMAT: TIME(s-time[:e-time])

x-time A 1 to 8 digit number in ‘hhmmsssth’ format as a single string.

‘hh’	Has a range of 0 to 23
‘mm’	Has a range of 0 to 59
‘ss’	Has a range of 0 to 59
‘th’	Has a range of 0 to 99

PARM-LIST: At least start time must be specified.

FUNCTION: To instruct SMFUTIL to set a start:end time basis. Only a single start:end time pair may be specified. The specified start time relates only to the first eligible date selected and the end time relates only to the last eligible date. The start time will be padded to the right with low values (0’s) and the end time will be padded to the right with high values (i.e. 595999). The time range is normal. This means that records from the start time on the first eligible date to the end time on the last eligible date will be available for selection. All records for dates in between the start and end date will be eligible for processing. If only the start-time is specified, records will be eligible from the start time on the first eligible date through midnight for the last eligible date. If less than the full hhmmsssth format is specified, the number will be assumed to be left justified with the exception that a single digit number will be padded to the left with a zero. Note that while the start time may be after the end time, caution should be exercised when only one day is eligible for selection. This situation would prevent any data from being copied.

EXAMPLE: TIME(1:13) is the same as
TIME(01000000:13595999).

NOTE: The TIME parameter should only be used in conjunction with the DATE parameter. If a DATE range is not specified, the TIME parameter will be ineffective, as the default first eligible date is 00001 and the last eligible date is 99365. All input data will fall between these dates so all data will be selected regardless of the TIME specification. To select only a specific time range from each day regardless of the dates of the input data, use the ITIME or XTIME specification.

R-CODE:

0	Data copied.
4	No records found.

6.2.167 TODAY

FORMAT: TODAY

PARM-LIST: NOT ALLOWED.

FUNCTION: To instruct SMFUTIL to select only records for the current date.

EXAMPLE: If SMFUTIL were executed with a specification of TODAY on Friday, December 27, 2002, only records available for that date would be copied.

NOTE: When TODAY is specified, an automatic default specification of SHORTCPY is assumed. The GOTOEOF or NOSTOP keyword may be used, after the TODAY keyword, to override this default.

R-CODE: 0 Data copied.
8 No records found for current date.

6.2.168 TSOUSER

FORMAT: TSOUSER(userid[,userid,...])

userid A 1 to 8 character user identification string mask. If less than 8 characters are specified, all users that start with the specified string will be selected. To limit the selection to a specific userid that is less than 8 characters, it is necessary to pad the userid to the right with blanks within the parenthesis. A wildcard character specification of '*' (an asterisk) may be used in any position to specify that any value is to be accepted in that position.

PARM-LIST: At least 1 userid required.

FUNCTION: To restrict record selection to SMF records that pertain to the specified TSO resource user. Records from the TSO subset with the selected userid(s) will be selected. Only record types with userid capability are screened. Other record types that are otherwise selected will bypass userid validation and be output.

EXAMPLE: TSOUSER(MYID) would select all user id's that begin with "MYID".
TSOUSER(MYID) would select only records for which the user id was "MYID".
TSOUSER(**ID) would select only records for which the user id has the characters 'ID' in the third and fourth positions. This is equivalent to TSOUSER(**ID****).

NOTE: See the section titled "USERID RECOGNITION" on page 151.

R-CODE: 0 Data copied.
4 No records found for TSO user.

6.2.169 TYPETODD

FORMAT:	TYPETODD(reclist-ddname[,reclist-ddname])
reclist	A list of 1 or more record types or ranges, in the range of 0 to 255, separated by commas. Each specification may be a discreet type (i.e. 15), may be a range (i.e. 70:79) or may specify one or more subtypes in parenthesis (i.e. 30(4,5). The list is terminated with a dash (-) which is followed by the DD name.
ddname	A valid DDNAME that matches a DD statement specified in the execution JCL or previously defined via a dynamic allocation DDA block.
PARM-LIST:	At least one record type to DDNAME association required.
FUNCTION:	<p>To instruct SMFUTIL to direct certain record types and/or certain subtypes of certain record types to a user defined DD name. One or more record types and/or subtypes may be associated with each DD name. If a range of record types is specified (i.e. rrr:rrr) neither record specification may be followed by a subtype specification. The same DDNAME may be specified more than once in the parameter list or in additional TYPETODD statements. The same record type or record type within a range may be directed to more than one DD name. Both types of syntax may be intermixed within one TYPETODD specification statement. See the section entitled "SUBTYPE SELECTION CRITERIA" on page 151. for more information on selecting records by subtype. The same DDNAME should not be the target for the same record type more than once and should not be the target of the same record type both with and without a subtype specification.</p> <p>The SUBTYPE keyword and any specification of record subtype values within a TYPETODD keyword are mutually exclusive. They may not both be used within the same execution of SMFUTIL.</p>
EXAMPLE1:	TYPETODD(4:5-FOURFIVE,30-TYPE30,128:255-USER)
EXAMPLE2:	TYPETODD(30-TYPE30,30,70:79-PDBLOAD) TYPETODD(230-PDBLOAD,70:79,240-CMF)
EXAMPLE3:	TYPETODD(30-T30ALL,30(5)-T30ST5) TYPETODD(30(1,2,3,4)-T301TO4)
NOTE:	The DD names specified must not begin with SYSUTxx, SYSCKPT, SYSPRINT, SYSIN, or SYSOUT. The DCB's will be constructed and will have the same validation and automatic DCB specifications as are supplied for the SYSUTxx DD statements. If checkpoint processing is in effect (SYSCKPT DD statement present), any data set pointed to by a user specified DDNAME that has a disposition of 'MOD' and resides on a tape device, will be checkpoint protected against data duplication.
R-CODE:	Not affected.

6.2.170 USER1

FORMAT: USER1(exit-name)

exit-name A 1 to 8 character name of a user supplied IFASMFD P USER1 type load module exit. The load module must be available in the LINKLST or on LPALIB, JOBLIB, or STEPLIB.

PARM-LIST: Exit load module name required.

FUNCTION: Instructs SMFUTIL to take a user supplied exit after each record is read from the input data set but before it is qualified for selection. The exit is supplied a 3 word parameter list address in register 1 as defined for an IFASMFD P USER1 exit. The first word of the list points to a 3 word work area for use by the exit. The second word of the parm list points to the record to be processed. The third word points to the input DDNAME that the record was read from. A return code of 0 causes normal processing to continue. A return code of 4 causes the record to be skipped (dropped from output) and the run continues. A return code value greater than 4 causes the exit not be called again but processing continues.

R-CODE: 0 Load module located.
8 Specified load module not found.

6.2.171 USER2

FORMAT: USER2(exit-name)

exit-name A 1 to 8 character name of a user supplied IFASMFD P USER2 type load module exit. The load module must be available in the LINKLST or on LPALIB, JOBLIB, or STEPLIB.

PARM-LIST: Exit load module name required.

FUNCTION: Instructs SMFUTIL to take a user supplied exit for each record to be written to an output data set. The exit is supplied a 3 word parameter list address in register 1 as defined for an IFASMFD P USER2 exit. The first word of the list points to a 3 word work area for use by the exit. The second word of the parm list points to the record to be processed. The third word points to the output DDNAME that the record is to be written to. A return code of 0 causes normal processing to continue. A return code of 4 causes the record to be skipped (dropped from output) and the run continues. A return code value greater than 4 causes the exit not be called again but processing continues.

NOTE: Because SMFUTIL supports multiple output data sets, it is possible that the supplied exit will be called multiple times for the same record.

R-CODE: 0 Load module located.
8 Specified load module not found.

6.2.172 USER3

FORMAT: USER3(exit-name)

exit-name A 1 to 8 character name of a user supplied IFASMFDP USER3 type load module exit. The load module must be available in the LINKLST or on LPALIB, JOBLIB, or STEPLIB.

PARM-LIST: Exit load module name required.

FUNCTION: Instructs SMFUTIL to take a user supplied exit after all records are written to an output data set and the data set has been closed. The exit is supplied a 3 word parameter list address in register 1 as defined for an IFASMFDP USER3 exit. The first word of the list points to a 3 word work area for use by the exit. The second word of the parm list points to closed DCB for the data set just completed. The third word points to the output DDNAME of the data set just completed and closed. A return code of 0 causes normal processing to continue. A return code value greater than 0 causes the exit not be called again but processing continues.

R-CODE:

0	Load module located.
8	Specified load module not found.

6.2.173 USERID

FORMAT: USERID(userid[,userid,...])

userid A 1 to 8 character user identification string mask. If less than 8 characters are specified, all users that start with the specified string will be selected. To limit the selection to a specific userid that is less than 8 characters, it is necessary to pad the userid to the right with blanks within the parenthesis. A wildcard character specification of '*' (an asterisk) may be used in any position to specify that any value is to be accepted in that position.

PARM-LIST: At least 1 userid required.

FUNCTION: To restrict record selection to SMF records that pertain to the specified resource user. All record types with userid capability are screened. Other record types that are selected utilizing other control keywords will bypass userid validation and be output.

EXAMPLE: USERID(MYID) would select all user id's that begin with "MYID".

USERID(MYID) would select only records for which the user id was "MYID".

USERID(**ID) would select only records for which the user id contained the characters 'ID' in the third and fourth positions. This is equivalent to USERID(**ID****).

NOTE: See the section titled "USERID RECOGNITION" on page 151.

R-CODE:

0	Data copied.
4	No records found for user.

6.2.174 _VALUEONLY

FORMAT: VALUEONLY

PARM-LIST: NOT ALLOWED

FUNCTION: To instruct SMFUTIL that records selected via a value comparison test (i.e. SIDSPLIT, DB2SPLIT, and CICSSPLIT) are the only records to be copied. Records not meeting the required value comparison test are to be discarded. As long as no other record types are selected for processing, SYSUT2 is not required and will not be used if supplied.

NOTE: If other record types are enabled (INCLUDE'd) in the same run as the value split parameter, SYSUT2 will be required to receive the other record types. The VALUEONLY specification will serve to discard the value split record types that did not meet the value test. For example, a type 110 record that does not contain an APPLID selected via CICSSPLIT parameter will be discarded. If VALUEONLY were not specified, this record would have normally have been written to SYSUT2.

R-CODE: Not affected.

6.2.175 _VCBUFF

FORMAT: VCBUFF(numbuff)

numbuff A integer value of 2 to 999 indicating the number of buffers to be supplied to VSAM for CLEAR functions.

PARM-LIST: REQUIRED

FUNCTION: To instruct SMFUTIL to use the indicated number of buffers during VSAM operations during CLEAR functions.

R-CODE: Not affected.

6.2.176 _VRBUFF

FORMAT: VRBUFF(numbuff)

numbuff A integer value of 2 to 999 indicating the number of buffers to be supplied to VSAM for read operations.

PARM-LIST: REQUIRED

FUNCTION: To instruct SMFUTIL to use the indicated number of buffers during VSAM read operations.

R-CODE: Not affected.

6.2.177 VSAMMOD

FORMAT: VSAMMOD[(ddname1,ddname2,...)]

ddnamex A 1 to 8 character DDNAME that identifies a VSAM output data set that is to be extended (MOD'ed) instead of over written.

PARM-LIST: OPTIONAL

FUNCTION: To instruct SMFUTIL on the disposition of output VSAM data sets. If VSAMMOD is specified without a parameter list, ALL output VSAM files will be extended. If a parameter list is specified, it must contain ddnames of output VSAM files to be extended. If a user DDNAME is specified (any DDNAME other than SYSUTxx SMFUTIL standard output ddnames), it must have been previously defined via a TYPETODD keyword.

R-CODE: 0 VSAM files extended as requested.
8 Specified DDNAME not defined.

6.2.178 VSINGLE

FORMAT: VSINGLE

INVERSE: VMULTI

PARM-LIST: NOT ALLOWED

FUNCTION: To instruct SMFUTIL that during value matching split request processing (SIDSPLIT, **CICSSPLIT** or DB2SPLIT), records are to be copied only to the first DDNAME that has matching criteria. All others are to be skipped. This allows specific request to be placed first and more generic (masking) request to be placed later. In essence, a "catch all" specification could be specified last.

EXAMPLE: VSINGLE
CICSSPLIT(ABCDPROD-CICSPROD)
CICSSPLIT(*-CICSOTHR)

Records produced by the CICS with APPLID ABCDPROD would be copied to DDNAME CICSPROD only with all other CICS APPLID's going to CICSOTHR. Note that if COPYVBTH were specified, all CICS records would also go to the standard outputs (SYSUT2/SYSUT2D). If COPYVBTH were not specified, NO CICS records would go to the standard outputs.

R-CODE: Not affected

6.2.179 VWBUFF

FORMAT: VWBUFF(numbuff)

numbuff A integer value of 2 to 999 indicating the number of buffers to be supplied to VSAM for write operations.

PARM-LIST: REQUIRED

FUNCTION: To instruct SMFUTIL to use the indicated number of buffers during VSAM write operations.

R-CODE: Not affected.

6.2.180 WEEKEND

FORMAT: WEEKEND

PARM-LIST: NOT ALLOWED

FUNCTION: To instruct SMFUTIL to select only records that were created on a weekend day (Saturday or Sunday) for processing. If both WEEKEND AND WORKDAY are specified, the last one encountered will be honored.

R-CODE: 0 Weekend days selected.
8 No weekend days found.

6.2.181 WEEKPCAL

FORMAT: WEEKPCAL(cal-name)

cal-name A 1 to 8 character load module name for the user defined perpetual week calendar required or the character "0".

PARM-LIST: One and only one perpetual calendar load module name must be coded or a zero is coded to indicate no calendar is to be used.

FUNCTION: To instruct SMFUTIL to use the specified week calendar for this execution.

NOTE: The load module named must have been constructed via the SMFUPCAL macro and must be available to be loaded at execution time.

R-CODE: NO EFFECT.

6.2.182 WEEKSTART

FORMAT: WEEKSTART(day)

day Specification of the name of a day that defines the first day of the week for CURRENTWEEK and PREVWEEK date calculations.

PARM-LIST: REQUIRED

FUNCTION: To instruct SMFUTIL to begin a logical week on the day specified. WEEKSTART is used in conjunction with the CURRENTWEEK and PREVWEEK keywords and must precede either of them in the input stream. The specification of 'day' must be from the following list:

SAT, SUN, MON, TUE, WED, THU, FRI

CURRENTWEEK or PREVWEEK will calculate the date of the beginning of the requested week and begin to select data on that date. CURRENTWEEK will to the current date. PREVWEEK will continue to the date preceding the start of the current week.

R-CODE: 0 WEEKSTART day accepted.
8 Invalid 'day' specified.

6.2.183 _WORKDAY

FORMAT: WORKDAY
 PARM-LIST: NOT ALLOWED
 FUNCTION: To instruct SMFUTIL to select only records that were created on a normal work day (Monday through Friday) for processing. If both WEEKEND AND WORKDAY are specified, the last one encountered will be honored.
 R-CODE: 0 Work days selected.
 8 No work days found.

6.2.184 _WSPLIT (OBSOLETE)

FORMAT: WSPLIT[(day)]
NOTE: **This keyword is still available for compatibility with prior releases. It may be removed in a future release. It is strongly recommended that this keyword not be used to accomplish a desired split function. The dynamic allocation control block method should be used instead by specifying the split boundary via the DDAEND keyword.**
 day Day specification to split data on.
 PARM-LIST: OPTIONAL
 FUNCTION: To instruct SMFUTIL to split the input data on a week boundary. The default to begin a new week is Saturday (SAT). If, after records have been written to SYSUT2, a record is encountered that was written on the day specified, it and all following records will be written to SYSUT3. Valid day specifications are SAT, SUN, MON, TUE, WED, THU, and FRI.
 NOTE: See the section entitled "INPUT DATA SPLITTING" on page 166 for information about requirements for proper splitting results.
 R-CODE: 0 No change found.
 2 Week Change found. SYSUT3 has data.

6.2.185 WSPLITDP (OBSOLETE)

FORMAT: WSPLITDP[(day)]

NOTE: This keyword is still available for compatibility with prior releases. It may be removed in a future release. It is strongly recommended that this keyword not be used to accomplish a desired split function. The dynamic allocation control block method should be used instead by specifying the split boundary via the DDAEND keyword.

day Day specification to split data on.

PARM-LIST: OPTIONAL

FUNCTION: To instruct SMFUTIL to split the data output to the duplex data sets on a week boundary. The default to begin a new week is Saturday (SAT). If, after records have been written to SYSUT2D, a record is encountered that was written on the day specified, it and all following records will be written to SYSUT3D. Valid day specifications are SAT, SUN, MON, TUE, WED, THU, FRI, and SAT.

NOTE: See the section entitled “INPUT DATA SPLITTING” on page 166 for information about requirements for proper splitting results.

R-CODE: 0 No change found.
2 Week Change found. SYSUT3D has data.

6.2.186 XACCOUNT

FORMAT: XACCOUNT(acctcode[,acctcode,...])

acctcode A 1 to 8 character accounting string mask. If less than 8 characters are specified, all accounts that start with the specified string will be excluded. To limit the selection to a specific account that is less than 8 characters, it is necessary to pad the string to the right with blanks within the parenthesis. A wildcard specification of ‘*’ (an asterisk) may be used in any position to specify that any value is to be matched in that position.

PARM-LIST: At least 1 acctcode specification required.

FUNCTION: To restrict record selection to SMF records that do not contain the specified account code(s). Records from the ACCOUNT subset with the specified account(s) will be excluded. Only record types with account capability are screened. Other record types that are otherwise selected will bypass account validation and be included in the output data.

EXAMPLE: XACCOUNT(AC01) would exclude all records with account's that begin with “AC01”.
XACCOUNT(AC01) would exclude only records for which the account code was “AC01”.

XACCOUNT(**01) would exclude only records for which the account code has the characters ‘01’ in the third and forth positions. This is equivalent to XACCOUNT(**ID***).

NOTE: Remember that not all records that support an account code field will contain one and some may have a truncated portion of the account field from the job card (if more than 4 characters was specified as an account code). See the section titled “ACCOUNT CODE RECOGNITION” on page 153.

R-CODE: 0 Data for Account copied.
4 All data excluded.

6.2.187 XBGUSER

FORMAT: XBGUSER(userid[,userid,...])

userid A 1 to 8 character user identification string mask. If less than 8 characters are specified, all users that start with the specified string will be excluded. To limit the consideration to a specific userid that is less than 8 characters, it is necessary to pad the userid name to the right with blanks within the parenthesis. A wildcard specification of '*' (an asterisk) may be used in any position to specify that any value is to be matched in that position.

PARM-LIST: At least 1 userid required.

FUNCTION: To restrict record selection to SMF records that pertain to batch resource user(s) other than those specified. Records from the BATCH subset with the selected userid(s) will be excluded. Only record types with userid capability are screened. Other record types that are otherwise selected will bypass userid validation and be processed.

EXAMPLE: XBGUSER(MYID) would exclude all user id's that begin with "MYID".
 XBGUSER(MYID) would exclude only records for which the user id was "MYID".
 XBGUSER(**ID) would exclude only records for which the user id has the characters 'ID' in the third and fourth positions. This is equivalent to XBGUSER(**ID****).

NOTE: See the section titled "USERID RECOGNITION" on page 151.

R-CODE: 0 Data for Batch user copied.
 4 All data excluded.

6.2.188 XDASET

FORMAT: XDASET(dsname[,dsname,...])

dsname A 1 to 44 character data set name or data set name prefix for which records are to be excluded. If less than 44 characters are specified, all data sets that start with the specified string will be excluded. To limit the selection to a specific data set that is less than 44 characters, it is necessary to pad the data set name to the right with at least one blank within the parenthesis. A wildcard specification of '*' (an asterisk) may be used in any position to specify that any value is to be accepted in that position.

PARM-LIST: At least 1 data set name specification required.

FUNCTION: To cause all records that contain data set name fields to be tested for a specific data set name or data set name prefix. Those that match will be excluded. All record types which are defined as in the DSNAMES subtype (see "MODIFICATION AND CUSTOMIZATION" on page 199 for more information) are turned on for selection. Only those records that match the data set specification are excluded from output. The data set specification is length sensitive. If the data set specification is less than 44 characters in length, it will be used as a mask to select all data sets that begin with the specification. If a data set that is less than 44 characters is desired only, it must be padded to the right with at least one blank. Other records that do not have data set names embedded in them may be specifically selected and will bypass data set name validation and will be put to the output data set(s). Multiple XDASET parameters may be specified.

R-CODE: 0 Data was written to output.
 4 No data output. All records excluded.

6.2.189 XERROFF

FORMAT: XERROFF
 ALIAS: NOXERROR
 INVERSE: XERROR
 PARM-LIST: NOT ALLOWED.
 FUNCTION: To instruct SMFUTIL to not continue after an I/O error has occurred on the input data set. A diagnostic message is printed and execution terminates. This is useful for determining the cause of excessive I/O errors on a volume that should be good.
 R-CODE: Not affected.

6.2.190 XJOBNAME

FORMAT: XJOBNAME(job-name[,job-name,...])
 job-name A one to eight character job-name or job-name prefix for which records are to be excluded. If less than 8 characters are specified, all job names that start with the specified string will be excluded. To limit the selection to a specific job name that is less than 8 characters, it is necessary to pad the job name to the right with blanks within the parenthesis. A wildcard specification of '*' (an asterisk) may be used in any position to specify that any value is to be matched in that position.
 PARM-LIST: At least 1 job name specification required.
 FUNCTION: To cause all records for a specific job name (JOB or STC type) or whose job name begins with a specific prefix to be excluded. All record types which have Job name capability (see "MODIFICATION AND CUSTOMIZATION" on page 199 for more information) are turned on for selection. Only those records that match the job name specification are excluded. The job name specification is length sensitive. If the job name specification is less than 8 characters in length, it will be used as a mask to exclude all job names that begin with the specification. If a job name that is less than 8 characters is desired only, it must be padded to the right with blanks to specify a total length of 8 characters. Other records that do not have job names embedded in them may be specifically selected and will bypass job name validation and will be put to the output data set(s).
 R-CODE: 0 Data was written to output.
 4 No data output. All data records excluded.

6.2.191 XSYSID

FORMAT: XSYSID(smfid[,smfid,...])

smfid One to four character System SMFID or SMFID prefix or mask.

PARM-LIST: At least 1 smfid required.

FUNCTION: To instruct SMFUTIL to exclude records produced by the specified system or systems. If less than four characters are specified the id will be used as a mask and all id's encountered with the prefix mask will be excluded. If a specified id of less than four characters is desired, the id should be right padded with blanks to a length of four characters in total. If any position contains an asterisk (*) that position will be treated as a wildcard character and any character will be accepted in that position. For example a specification of '*083' would match 'A083' and 'B083' equally but not 'C090'.

R-CODE: 0 -Data copied.
4 -All data records excluded.

6.2.192 XSYSTYPE

FORMAT: XSYSTYPE(opsysid)

opsysid Three character operating system identifier.

PARM-LIST: REQUIRED.

FUNCTION: To instruct SMFUTIL to exclude records produced by a particular operating system type. Supported type definitions are SP2, SP3, SP4 SP5, ESA, MVS, MXA, VS1, VS2, and SVS.

R-CODE: 0 Data copied.
8 All data excluded.

6.2.193 XTIME

FORMAT: XTIME(start-time[:end-time])

x-time A 1 to 8 digit number in hhmmssst format.

PARM-LIST: At least start time must be specified.

FUNCTION: To instruct SMFUTIL to select only records outside of a specific time range to be processed. The time range is exclusive. This means that only records outside the range are eligible for selection. Records inside the range, no matter what the date, will be discarded. If only the start-time is specified, midnight will be assumed as the end time. If less than the full hhmmssst format is specified, the number will be assumed to be left justified with the exception that a single digit number will be padded to the left with a zero. The start time must be less than or equal to the end time.

EXAMPLE: XTIME(1:12) is the same as
XTIME(01000000:12000000).

R-CODE: 0 Data copied.
4 No records found outside requested time range.

6.2.194 XTSOUSER

FORMAT: XTSOUSER(userid[,userid,...])

userid A 1 to 8 character user identification string mask. If less than 8 characters are specified, all users that start with the specified string will be excluded. To limit the selection to a specific userid that is less than 8 characters, it is necessary to pad the userid to the right with blanks within the parenthesis. A wildcard character specification of '*' (an asterisk) may be used in any position to specify that any value is to be matched in that position.

PARM-LIST: At least 1 userid required.

FUNCTION: To restrict record selection to SMF records that pertain to the TSO resource users other than those specified. Records from the TSO subset with the specified userid(s) will be excluded. Only record types with userid capability are screened. Other record types that are otherwise selected will bypass userid validation and be output.

EXAMPLE: XTSOUSER(MYID) would exclude all user id's that begin with "MYID".
 XTSOUSER(MYID) would exclude only records for which the user id was "MYID".
 XTSOUSER(**ID) would exclude select only records for which the user id has the characters 'ID' in the third and fourth positions. This is equivalent to XTSOUSER(**ID****).

NOTE: See the section titled "USERID RECOGNITION" on page 151.

R-CODE: 0 Data copied.
 8 All data records excluded.

6.2.195 XUCB

FORMAT: XUCB(ucbadrs[,ucbadrs-ucbadrs])

ucbadrs A two to four character UCB address mask or UCB address range.

PARM-LIST: At least one UCB mask must be coded.

FUNCTION: To instruct SMFUTIL to exclude records that contain specific UCB addresses that match the specified mask.

NOTE: IUCB automatically turns off all record types that SMFUTIL does not have UCB address location data on. The SMFUTLRT can be updated to contain the location of volume serial data for user type records.

NOTE: Any record type selected for processing by other means (i.e. via an INCLUDE keyword) that does not contain volume serial information will be pass directly to output processing.

R-CODE: Not directly affected. However, if a UCB mask is specified that causes all records to be excluded, a return code of 8 will be issued unless COPY00K is specified.

6.2.196 XUSERID

FORMAT: XUSERID(userid[,userid,...])

userid A 1 to 8 character user identification string mask. If less than 8 characters are specified, all users that start with the specified string will be excluded. To limit the consideration to a specific userid that is less than 8 characters, it is necessary to pad the userid to the right with blanks within the parenthesis. A wildcard character specification of '*' (an asterisk) may be used in any position to specify that any value is to be matched in that position.

PARM-LIST: At least 1 userid required.

FUNCTION: To restrict record selection to SMF records that contain userid specifications other than those specified. All record types with userid capability are screened. Other record types that are selected utilizing other control keywords will bypass userid validation and be output.

EXAMPLE: XUSERID(MYID) would exclude all user id's that begin with "MYID".
 XUSERID(MYID) would exclude only records for which the user id was "MYID".
 XUSERID(**ID) would exclude only records for which the user id contained the characters 'ID' in the third and fourth positions. This is equivalent to XUSERID(**ID****).

NOTE: See the section titled "USERID RECOGNITION" on page 151.

R-CODE: 0 Data copied.
 8 All data excluded.

6.2.197 XVOLUME

FORMAT: XVOLUME(volmask)

ALIAS: XVOLSER

volmask A one to six character volume serial mask.

PARM-LIST: At least one volmask must be coded.

FUNCTION: To instruct SMFUTIL to exclude records that contain volume serial numbers that match the specified mask.

NOTE: XVOLSER automatically turns off all record types that SMFUTIL does not have volume serial location data on. The SMFUTLRT can be updated to contain the location of volume serial data for user type records.

NOTE: Any record type selected for processing by other means (i.e. via an INCLUDE keyword) that does not contain volume serial information will be pass directly to output processing.

R-CODE: Not directly affected. However, if a volume mask is specified that causes all records to be excluded, a return code of 8 will be issued unless COPY00K is specified.

6.2.198 YEARFIRST

FORMAT: YEARFIRST

ALIAS: YFIRST

PARM-LIST: NOT ALLOWED.

FUNCTION: To instruct SMFUTIL to place the year value first on data set name suffixes generated for BOTH monthly and weekly format. This will allow catalog listings of SMF archive data sets to be in chronological order. Note that DAY suffix structures are unaffected.

EXAMPLE: A weekly suffix of '.W332002' would be changed to '.Y2002W33' and a monthly suffix of '.OCT2002' would be changed to '.Y2002OCT' if YEARFIRST were specified.

R-CODE: Not affected.

6.2.199 YEARFIRSTM

FORMAT: YEARFIRSTM

ALIAS: YFIRSTM

PARM-LIST: NOT ALLOWED.

FUNCTION: To instruct SMFUTIL to place the year value first on data set name suffixes generated for monthly format. This will allow catalog listings of SMF archive data sets to be in chronological order. Note that DAY and WEEK type suffix structures are unaffected.

EXAMPLE: A monthly suffix of '.OCT2002' would be changed to '.Y2002OCT' if YEARFIRSTM were specified.

R-CODE: Not affected.

6.2.200 YEARFIRSTW

FORMAT: YEARFIRSTW

ALIAS: YFIRSTW

PARM-LIST: NOT ALLOWED.

FUNCTION: To instruct SMFUTIL to place the year value first on data set name suffixes generated for weekly format. This will allow catalog listings of SMF archive data sets to be in chronological order. Note that DAY and MONTH suffix structures are unaffected.

EXAMPLE: A weekly suffix of '.W332002' would be changed to '.Y2002W33' if YEARFIRSTW were specified.

R-CODE: Not affected.

6.2.201 YESTERDAY

FORMAT: YESTERDAY

ALIAS: YESTERDY, YESTRDAY

PARM-LIST: NOT ALLOWED.

FUNCTION: To instruct SMFUTIL to select only records for the previous day.

EXAMPLE: If SMFUTIL were executed with a specification of YESTERDAY on Friday, December 27, 2002, only records available for Thursday, December 26, 2002 would be copied.

NOTE: When YESTRDAY is specified, an automatic default specification of SHORTCPY is assumed. The GOTOEOF or NOSTOP keyword may be used, after the YESTRDAY keyword, to override this default.

R-CODE: 0 Data copied.
8 No records found for previous day.

6.3 CONTROL KEYWORDS - DDA BLOCK COMMANDS

Input and output SMFUTIL data sets may be dynamically allocated via a DDA block. It serves to identify the structure and usage format of a data set. DDA BLOCK commands have the same syntactical format as NORMAL commands. They are restricted in that they may only be specified between 'DDASTART' and 'DDAEND' block definition statements. No NORMAL keywords may be specified within the block boundaries.

6.3.1 AEXPIRE

FORMAT: AEXPIRE(####)

PARM-LIST: Required.

A one to four digit number of days or a 7 digit specific Julian date (i.e. 2003001).

FUNCTION: To allow the user to specify the length of time the archive record associated with this data set is to be retained in the archive. The AEXPIRE parameter in the SMFUDFLT module is set to 0 as shipped which indicates permanent retention. This may be permanently changed to a 1 to 4 digit number of days value. A 7 digit Julian date string may not be specified in the SMFUDFLT module.

During subsequent searches of the archive for input data, any record that has had the expiration date pass will not be considered for input.

EXAMPLE AEXPIRE(730)

R-CODE: Not affected

6.3.2 ASETNAME

FORMAT: ASETNAME(name)

PARM-LIST: Required.

name A one to eight character name to be assigned to this data in the SMFUTIL archive.

FUNCTION: To allow the user to override the regular set name that would be used (i.e. 'NORMAL' for SYSUT2, 'DUPLEX' for SYSUT2D, or the first 5 characters of the DDNAME for TYPETODD data sets). The name specified may not contain any blanks but may contain special characters if the site wishes. This name will be used via the SETNAME parameter in subsequent data extraction runs to select the data to be extracted.

ARCHIVE and/or ARCHUSER must also be specified for the run or the ASETNAME value will be ignored as no archiving is taking place.

EXAMPLE ASETNAME(SPECIAL)

R-CODE: Not affected

6.3.3 DDABIDHR

FORMAT: DDABIDHR

PARM-LIST: Not Allowed.

FUNCTION: To instruct SMFUTIL that the associated dataset is to have Block ID values (BIDs) on an hourly basis. Each BID slot will represent one hour of data.

NOTE: The associated dataset must not have more than a single day of data and all the data must be from the same date in ascending order. If more than a single date is detected, DDABIDHR will be suppressed and a per day value will be kept.

DDABIDHR is valid for input and output datasets but has no effect unless archiving is actually done for the dataset.

R-CODE: Not affected

6.3.4 BIDSTART

FORMAT: BIDSTART(number)

PARM-LIST: Required.

number An 8 digit hex string specifying an absolute block that this input data set is to be positioned to prior to the start of input.

FUNCTION: To allow the user to perform high speed positioning of an input cartridge data set when the required input data (date/time range) is well into the data set. The value specified is used in a POINT macro to cause the device to skip to the specified block and input processing begins there. If the value specified is greater than the number of blocks in the data set, an I/O error occurs.

NOTE: This parameter has no effect on output data sets.

This is a crude way to skip over data that is not required. An alternative is to use the ARCHIVE database to select required input data. The ARCHIVE database automatically maintains up to 31 block ID values per volume (one for each day of data or one for each hour of data on a daily dataset) of input data and can automatically do high-speed positioning to the nearest day on which input is to begin.

EXAMPLE: BIDSTART(01001CA4)

R-CODE: Not affected

6.3.5 BLKSIZE

FORMAT:	BLKSIZE(number)
PARM-LIST:	Required.
number	An integer number specifying the block size value for this data set. The maximum value is 32760.
FUNCTION:	To allow the user to override the default block size value chosen by SMFUTIL. This value may also be specified in the DCB keyword as a positional parameter.
NOTE:	If the BLKSIZE keyword is omitted and the SDB parameter in the SMFUDFLT module is set to 'NO', SMFUTIL will choose an appropriate BLKSIZE value based on the supplied (or defaulted) LRECL and RECFM values in concert with the device type. If BLKSIZE is specified with a value of '0', SMFUTIL will honor it regardless of the setting of SDB in order to support Extended Sequential Data set creation by SMS.
R-CODE:	Not affected

6.3.6 BUFNO

FORMAT:	BUFNO(number)
PARM-LIST:	Required.
number	An integer number specifying the number of I/O buffers to be assigned to this data set.
FUNCTION:	To allow the user to override the SRBUFF, SWBUFF, or VRBUFF specifications for a single data set. The BUFNO maximum that may be specified is 250 for VSAM input and output data sets and 99 for sequential input and output data sets.
	EXAMPLE BUFNO(20)
R-CODE:	Not affected

6.3.7 CKPTGDG

FORMAT:	CKPTGDG(name[,start])
PARM-LIST:	Required.
name	A one to four character checkpoint name suffix valid for use as the last four characters of a PDS member name. The characters must be alpha, '#', '\$' or numeric digits 0 thru 9. This name will be used in later executions via the GDGCKPTI parameter to interrogate the checkpoint record and allocate input GDG data sets.
start	An optional starting GDG absolute generation number to be placed into the checkpoint record. This defines the starting generation for subsequent GDGCKPTI input requests. It is typically the oldest (earliest) version in the catalog.
FUNCTION:	<p>To cause SMFUTIL to checkpoint an output GDG data set to a user named checkpoint record. Each time the CKPTGDG execution is performed and a new generation is added to the GDG index, the checkpoint record is updated to reflect the presence of the new generation. When the GDGCKPTI execution is run, the checkpoint record is updated to reflect that the GDG generations represented have been input. The CKPTGDG and GDGCKPTI executions act as linked processes with one or more CKPTGDG runs being interspersed with a single GDGCKPTI run.</p> <p>The 'start' specification may be optionally used to set and/or reset the starting absolute generation number in the checkpoint record to the desired value. This will be used by GDGCKPTI as the starting point for input.</p>
R-CODE:	Not affected

6.3.8 CKPTLAST

FORMAT:	CKPTLAST(ckptsuffix)
PARM-LIST:	Required.
ckptsuffix	A one to four character checkpoint name suffix valid for use as the last four characters of a PDS member name. The characters must be alpha, '#', '\$' or numeric digits 0 thru 9.
FUNCTION:	To instruct SMFUTIL to use the supplied name to checkpoint the last record output to this dataset. This checkpoint record will be used in place of the LASTREC1 checkpoint for this dataset exclusively. Upon successful termination, SMFUTIL will store information about the last record written to this output dataset under this checkpoint name. The next time the job runs, this checkpoint record information will be used to detect split boundary crossing for this dataset in lieu of the LASTREC1 checkpoint record. This enables SMFUTIL to properly detect boundary crossings for datasets with less frequent record occurrences where the general purpose LASTREC1 checkpoint record might cause these boundaries to be missed.
NOTES:	<p>Each CKPTLAST name value must be unique. The same name may not be specified for more than one DDA block set.</p> <p>The value supplied by the user is used as a suffix for the standard name '\$LDT'. The resulting member name is used to store the checkpoint record in the checkpoint PDS.</p> <p>The SYSCKPT checkpoint PDS is required for the CKPTLAST specification to be functional. Therefore it must be present prior to SMFUTIL encountering the CKPTLAST keyword in a DDA block. It must be in the execution JCL or defined prior to the DDA block via the CKPTDB keyword.</p> <p>The CKPTLAST specification is ignored for input datasets and should not be coded.</p>
R-CODE:	Not directly affected

6.3.9 DATACLASS

ALIAS:	DATACLAS
FORMAT:	DATACLASS(class)
PARM-LIST:	REQUIRED
class	A valid 1 to 8 character SMS Data Class identifier for this data set allocation.
FUNCTION:	To provide SMS with the Data Class for this data set.
NOTE:	SMS must be active for the DATACLASS keyword to be used. If it is not, dynamic allocation will fail.
R-CODE:	Not affected

6.3.10 DCB

FORMAT:	DCB(recfm,lrecl,blksize,trtch)
PARM-LIST:	Required but each position optional. A comma must be specified for each omitted parameter.
recfm	The record format to be assigned to this data set. Valid values are “VBS”, “VB”, “VS” and “V”.
lrecl	The logical record length to be assigned to this data set. The maximum value is 32767.
blksize	The block size to be assigned to this data set. The maximum value is 32760.
trtch	A keyword value of COMP or NOCOMP to indicate if IDRC compression is to be used for this data set.
FUNCTION:	To supply the DCB characteristics to be used for the data set being allocated. For input data sets “DCB” may be omitted as SMFUTIL will handle any input data set even if the label for the data set is missing or incorrect. For output data sets it is strongly recommended that the lrecl, blksize, and recfm DCB positions be omitted and SMFUTIL allowed to optimize the output data sets DCB characteristics for the device and volume chosen. Any of the four positions may be omitted. The separating comma must be used to denote position. SMFUTIL will chose default value for omitted positions based on any values supplied in the other positions.
NOTES:	<p>If the ‘blksize’ value is omitted and the SDB parameter in the SMFUDFLT module is set to ‘NO’, SMFUTIL will choose an appropriate BLKSIZE value based on the supplied (or defaulted) LRECL and RECFM values in concert with the device type. If ‘0’ is specified, SMFUTIL will honor it regardless of the setting of SDB in order to support Extended Sequential Data set creation by SMS.</p> <p>If COMP is specified, the output device must support the IDRC compaction/compression feature. If it does not or if compaction/compression is disabled on the device controller, the OPEN of the data set will fail with an ABEND S413-40 (MSG IEC145I with return code 40). TRTCH alone may be specified while allowing SMFUTIL to chose all other DCB parameters:</p> <p>EXAMPLE DCB(,,,COMP)</p>
R-CODE:	Not affected

6.3.11 DDALARGEBLOCK

FORMAT:	DDALARGEBLOCK
INVERSE:	DDANOLARGEBLOCK
PARM-LIST:	NOT ALLOWED.
FUNCTION:	<p>To instruct SMFUTIL that the IBM Large Block Interface (LBI) is to be requested for this output dataset. This specification has no meaning for input datasets as LBI is automatically supported.</p> <p>A specification of DDANOLARGEBLOCK will negate a global specification of LARGEBLK=YES (in the defaults module) or a prior LARGEBLOCK keyword.</p>
R-CODE:	Not affected

6.3.12 DDAEND

FORMAT: DDAEND[(SPLIT(t,i))]

PARM-LIST: OPTIONAL.

SPLIT(t,i) An optional keyword indicating the output data set is to be split on a selected boundary.

t A keyword of “D” for day split, “W” for week split, “M” for month split, or “U” for a User Calendar split.

i An integer specification that informs SMFUTIL how many splits to prepare for. Maximum value is 34.

FUNCTION: To terminate a DDA block structure and to tell SMFUTIL when to allocate the data set. Note that a SUFFIX(Dx) specification is mutually exclusive with the “DDAEND(ALLOCATE)” specification. “DEFER” must be specified (or defaulted) to allow the allocation to be deferred until data is to be placed to it and a data date thus be available for suffix generation.

If SPLIT is specified, it instructs SMFUTIL to split an output data set on the boundary ‘t’ and to allow for ‘i’ splits to occur. Valid values for ‘t’ are “D” for a day boundary, “W” for a week boundary and “M” for a month boundary.

R-CODE: 0 No split in output data found.
 1 Requested Day split found.
 2 Requested Week split found.
 3 Requested Month split found or User Calendar split found.

6.3.13 DDASTART

FORMAT: DDASTART

PARM-LIST: NOT ALLOWED.

FUNCTION: To instruct SMFUTIL that a set of DDA type block commands follow.

R-CODE: Not affected.

6.3.14 DDACOPY0RC

FORMAT: DDACOPY0RC(rc)

PARM-LIST: Optional

rc A value of 0, 4, 8 or 12 to indicate the minimum return code to be issued if the associated data set receives no output data during the run. If no ‘rc’ value is supplied a default of ‘4’ is used.

FUNCTION: To indicate to SMFUTIL what return code is to be set if a specific output data set receives no output data.

R-CODE: The specified value is set as minimum if the data set receives no output.

6.3.15 DDACREATE

FORMAT: DDACREATE

PARM-LIST: NOT ALLOWED.

FUNCTION: To instruct SMFUTIL that this DDA data set is to be created even if no output data is placed to it.

NOTE: If a run has a return code of greater than 8 (such as a syntax error) no creation of empty datasets will be performed.

CAUTION: If a run has no input data due to an ILIMIT(0) specification or an empty input dataset, any data set to be created as empty that contains a suffix or a date overlay in the data set name will have the current execution date used for that purpose.

NOTE Ignored for input data sets.

R-CODE: Not affected.

6.3.16 DDANOCREATE

FORMAT: DDANOCREATE

PARM-LIST: NOT ALLOWED.

FUNCTION: To instruct SMFUTIL that if it receives no output data, this DDA data set is **NOT** to be created even if CREATEU0 or CREATES0 (as appropriate) is specified.

NOTE Ignored for input data sets.

R-CODE: Not affected.

6.3.17 DDAUCAL

FORMAT: DDAUCAL(cal-name)

cal-name A 1 to 8 character load module name of a user defined calendar to be used for any “U” type split associated with the DDA block.

PARM-LIST: One and only one calendar load module name must be coded.

FUNCTION: To instruct SMFUTIL to use the specified calendar for this execution for this DDA block.

NOTE: The load module named must have been constructed via the SMFUMCAL macro and must be available to be loaded at execution time.

R-CODE: NO EFFECT.

6.3.18 DDNAME

ALIAS:	DDN
FORMAT:	DDNAME(ddn)
PARM-LIST:	REQUIRED
ddn	A valid 1 to 8 character DDNAME to be assigned to this allocation.
FUNCTION:	To provide the DDNAME to be assigned to this allocation. A DDNAME structure of SYSUT1xx will be interpreted as an input data set. All others will be interpreted as output data sets.
R-CODE:	Not affected

6.3.19 DISP

FORMAT:	DISP(initial,normal,abnormal)
PARM-LIST:	Required
initial	A value denoting the initial disposition of the data set. Valid values are “NEW”, “OLD”, “SHR”, or “MOD”.
normal	A value denoting the disposition of the data set if the job step completes normally or if the data set is deallocated. Valid values are “KEEP”, “CATLG”, “DELETE”, or “UNCATLG”.
abnormal	A value denoting the disposition of the data set if the job step terminated abnormally (ABENDS) prior to deallocation of the data set. Valid values are “KEEP”, “CATLG”, “DELETE”, or “UNCATLG”.
FUNCTION:	To supply the data set disposition. If omitted, input data sets (SYSUT1xx) will default to DISP(SHR,KEEP,KEEP) and output data sets will default to DISP(NEW,CATLG,DELETE).
R-CODE:	Not affected

6.3.20 DSNNAME

ALIAS:	DSN
FORMAT:	DSNAME(dsname)
PARM-LIST:	REQUIRED
dsname	A valid 1 to 44 character data set name to be allocated.
FUNCTION:	To provide the data set name to be allocated. The data set name must be a fully qualified data set name or a relative GDG structure (i.e. SYSX.SMF.DATA(+1) is a relative GDG) or a PDS data set name with a member specification. If a data set name suffix is to be attached, the data set name length specified may not be greater than 35 and may not be a GDG or PDS member structure.
NOTE:	<p>If the data set name supplied contains the character string '&SID', it will be replaced by the SMF system identifier of the system on which the job is executing.</p> <p>If DDNAME for this DDA block is an output data set (i.e. other than SYSUT1xx) and the data set name supplied contains the character string '&DID', it will be replaced by the SMF system identifier contained in the first record of processed. This is useful for creating unique data set names to contain SMF data for each system being dumped and still maintain a single set of control cards and JCL procedures.</p>
R-CODE:	Not affected

6.3.21 DUMMY

FORMAT:	DUMMY
PARM-LIST:	NOT ALLOWED
FUNCTION:	To instruct SMFUTIL that a DUMMY (NULLFILE) data set is to be allocated for this output data set.
NOTE:	DUMMY is only valid for output data set definitions.
R-CODE:	Not affected

6.3.22 EXPDT

FORMAT:	EXPDT(expire-date)
PARM-LIST:	REQUIRED
expire-date	A valid 5 digit expiration date in the form of 'yyddd' or a valid 7 digit long form expiration date in the form of 'yyyddd'.
FUNCTION:	To provide the expiration date for the data set being allocated. EXPDT may be specified for both input and output data set.
R-CODE:	Not affected

6.3.23 LABEL

FORMAT:	LABEL(number[,type])
PARM-LIST:	REQUIRED
number	A valid label number for this data set on a magnetic tape volume. The maximum value is 9999.
type	The type of label processing to be performed on the volume.
FUNCTION:	To data set position on the tape and label type processing to be performed for the volume containing the data set. Valid label type values are “SL”, “NL”, “BLP”, “SUL”, “LTM”, “AL”, “AUL”, or “NSL”.
R-CODE:	Not affected

6.3.24 LIKE

FORMAT:	LIKE(dsname)
PARM-LIST:	REQUIRED
dsname	A valid 1 to 44 character data set name to be used by SMS to extract DCB characteristics for this data set allocation.
FUNCTION:	To provide SMS with an example data set that has the same DCB characteristics that are to be assigned to this SMF output data set. The example data set should be one with DCB attributes that are appropriate for SMF data (i.e. RECFM=V/VB/VBS). This will prevent warning messages when SMFUTIL overrides invalid DCB characteristics from the example data set.
NOTE:	SMS must be active for the LIKE keyword to be used. If it is not, dynamic allocation will fail.
R-CODE:	Not affected

6.3.25 LRECL

FORMAT:	LRECL(number)
PARM-LIST:	Required.
number	An integer number specifying the logical record length value for this data set. The maximum value is 32767.
FUNCTION:	To allow the user to override the default logical record length value chosen by SMFUTIL. This value may also be specified in the DCB keyword as a positional parameter.
R-CODE:	Not affected

6.3.26 MANAGEMENTCLASS

ALIAS: MGMTCLAS
 FORMAT: MANAGEMENTCLASS(class)
 PARM-LIST: REQUIRED
 class A valid 1 to 8 character SMS management class identifier for this data set allocation.
 FUNCTION: To provide SMS with the Management Class for this data set.
 NOTE: SMS must be active for the MANAGEMENTCLASS keyword to be used. If it is not, dynamic allocation will fail.
 R-CODE: Not affected

6.3.27 MODEL

FORMAT: MODEL(dsname)
 PARM-LIST: REQUIRED
 dsname A valid 1 to 44 character data set name to be used as a DCB model for a GDG data set allocation.
 FUNCTION: To provide the model data set name for a GDG allocation. If omitted when a relative GDG data set name is specified, MODEL will default to the GDGMODEL parameter in the SMFUDFLT module. As shipped, this is set to 'SYS1.MODEL'. The model data set should be one with no DCB attributes specified in the format 1 DSCB. This will prevent warning messages when SMFUTIL overrides invalid DCB characteristics from the model.
 R-CODE: Not affected

6.3.28 RETPD

FORMAT: RETPD(number)
 PARM-LIST: REQUIRED
 number A integer in the range of 0 to 9999 that specifies the number of days a data set is to be retained.
 FUNCTION: To provide the retention period for a NEW or MOD data set.
 R-CODE: Not affected

6.3.29 RECFM

FORMAT:	RECFM(recfm)
PARM-LIST:	Required.
recfm	The record format to be assigned to this data set. Valid values are “VBS”, “VB”, “VS” and “V”.
FUNCTION:	To allow the user to override the default record format value chosen by SMFUTIL. This value may also be specified in the DCB keyword as a positional parameter.
NOTE:	All SMF data sets must have variable length records so only the “V” type specifications are supported.
R-CODE:	Not affected

6.3.30 RETRY

FORMAT:	RETRY([(cnt)[,time]])
PARM-LIST:	OPTIONAL
cnt	An integer in the range of 1 to 9999 that specifies the number of retries to be attempted for this data set.
time	An integer in the range of 1 to 9999 that specifies the number of seconds to wait between retries.
FUNCTION:	To instruct SMFUTIL that this DDA block allocation is to be retried if it fails allocation for unavailable dsname, volume, or unit. If the ‘cnt’ and/or ‘time’ are omitted, ‘cnt’ will default to 100 and ‘time’ to 15 seconds. When specified, if the allocation fails for one of the recoverable error codes, SMFUTIL will wait the time interval specified by ‘time’ and then retry the allocation. This will continue for ‘cnt’ times. If both DARETRY (global DDA retry) and RETRY are specified, the RETRY specifications (or defaults) will be taken.
R-CODE:	Not affected

6.3.31 SPACE

FORMAT:	SPACE(type[,pri-qty,sec-qty,RLSE])
type	The type of space allocation to be used. Only "TRKS" and "CYLS" may be specified for track or cylinder allocation. If the SPACE parameter is specified and this positional parameter is omitted, "CYLS" will be defaulted.
pri-qty	The primary space quantity to allocate. The maximum that may be specified is 9999. If omitted, for "CYLS" type allocation, "5" will be defaulted for the primary quantity. For "TRKS" type allocation "15" will be defaulted.
sec-qty	The secondary allocation space quantity to allocate for secondary extents. The maximum that may be specified is 9999. If omitted, no secondary extents will be allocated.
RLSE	If coded, causes the unused space to be released when the data set is closed and deallocated.
PARM-LIST:	REQUIRED
FUNCTION:	To specify the amount of space to be allocated for a new disk data set. If completely omitted, a default value of "CYLS,5,5,RLSE" will be used for disk type unit specification. The SPACE parameter is mutually exclusive with a UNIT specification of any type of TAPE device.
NOTE:	"TRKS" may be specified as "TRK", "TRACKS", or simply "T". "CYLS" may be specified as "CYL", "CYLINDER", or simply "C". "RLSE" may be specified as "RELEASE", "REL", or "R".
R-CODE:	Not affected

6.3.32 SPLITDISP

FORMAT:	SPLITDISP(initial,normal,abnormal)
PARM-LIST:	Required
initial	A value denoting the initial disposition of any split data set. Valid values are “NEW”, “OLD”, “SHR”, or “MOD”.
normal	A value denoting the disposition of any split data set if the job step completes normally or if the data set is deallocated. Valid values are “KEEP”, “CATLG”, “DELETE”, or “UNCATLG”.
abnormal	A value denoting the disposition of any split data set if the job step terminated abnormally (ABENDS) prior to deallocation of the data set. Valid values are “KEEP”, “CATLG”, “DELETE”, or “UNCATLG”.
FUNCTION:	To supply the split data set disposition. If omitted, split data sets will default to DISP(NEW,CATLG,DELETE). This parameter allows for the pre-allocation of split data sets in anticipation of a split function. It can be used for disk data sets that may require large allocations to ensure that sufficient space will be available when the split takes place by pre-allocating the data set and specifying: ‘SPLITDISP(OLD,KEEP,KEEP)’.
NOTE:	The specification of SPILTDISP is only useful if the SUFFIX keyword is also specified and the correct data set name structure is anticipated in the pre-allocated data set. SPLITDISP cannot be used for relative GDG specifications as they cannot be pre-allocated.
R-CODE:	Not affected

6.3.33 STORAGECLASS

ALIAS:	STORCLAS
FORMAT:	STORAGECLASS(class)
PARM-LIST:	REQUIRED
class	A valid 1 to 8 character SMS Storage Class identifier for this data set allocation.
FUNCTION:	To provide SMS with the Storage Class for this data set.
NOTE:	SMS must be active for the STORAGECLASS keyword to be used. If it is not, dynamic allocation will fail.
R-CODE:	Not affected

6.3.34 SUFFIX

FORMAT: SUFFIX(xx)

PARM-LIST: Required

xx A two character keyword that defines what type of suffix to generate and what date/time value is used to generate it.

FUNCTION: To instruct SMFUTIL that the data set name supplied is to have a suffix appended to it prior to allocation. The type of suffix to be generated is indicated by a two character specification. The first character of this specification indicates the source of the date/time stamp used in the suffix generation as follows:

C - Current machine date

P - A previous machine date/time calculated from current machine date/time adjusted backwards, by the suffix type specification.

D - Data record date/time value (from first SMF data record to be output to data set).

The second character defines the format of the suffix to be generated as follows:

T - Time .Jccyyddd.Thhmmss

X - Extended Time .Jccyyddd.Thhmmss.Hth

D - Day .Jccyyddd

W - Week .Wwwccyy

M - Month .mmmccyy

Y - Year .Yccyy

F - Full .Yccyy.Dvvdd

S - Week start .Yccyy.Wvvdd

P - Month start .Yccyy.Mvv

U - User Calendar suffix As defined by User Calendar

In the above definitions, the capital letters shown are as they appear in the final suffix. Lower case letters are place holders as follows:

cc - Integer century value (i.e. 20)

yy - Integer year of century value (i.e. 01)

ddd - Integer julian day of year (i.e. 001 for Jan 1)

hh - Integer hour of day (range 00 to 23)

mm - Integer minute of hour (range 00 to 59)

ss - Integer second of minute (range 00 to 59)

th - Integer tenths and hundredths of a second of minute (range 00 to 99)

vv - Integer month value (range 01 to 12)

dd - Integer day value (range 01 to 31)

The preceding rules yield the following valid values for 'xx':

DT	-	Data date time/date
DX	-	Data date time/date extended
DD	-	Data date day
DW	-	Data date week
DM	-	Data date month
DY	-	Data date year
DF	-	Data date full date stamp
DS	-	Data date week start
DP	-	Data date month start
DU	-	Data date mapped to User Calendar entry
CT	-	Current date/time
CX	-	Current date/time extended
CD	-	Current day
CW	-	Current week
CM	-	Current month
CY	-	Current year
CS	-	Current week start
CP	-	Current month start
CF	-	Current date full date stamp
CT	-	Current time
CX	-	Current time extended format
CU	-	Current date mapped to User Calendar entry
PD	-	Previous day
PW	-	Previous week
PM	-	Previous month
PY	-	Previous year
PS	-	Previous week start
PP	-	Previous month start

Note that there is no allowance for PF (Previous Full) or PT (Previous Time) as these definitions are meaningless. (i.e. Previous Full for what date or Previous Time from when?)

Output data sets generated with the time suffix format can be recalled for input in a subsequent run via the "CATINPUT" keyword. The SUFFIX keyword is mutually exclusive with GDG data set name structures and temporary data set name structures. See the section titles "DSNAME SUFFIX GENERATION" on page 167 for more information of the generation of data set name suffixes.

Output data set definitions may use either "D" (data) or "C" (current) suffix types. There is no provision for use of "P" for output datasets. Note that to use the "D" (data) type, "DEFER" must be specified or defaulted on the DDAEND parameter in order for the data record to be available to generated the desired suffix.

“C” and “P” type suffix requests may be used for input data sets to request that the data set name be qualified before use. “D” may not be used for input as no data record is available to provide the required date.

R-CODE: Not affected

6.3.35 TRTCH

FORMAT: TRTCH(trtch)

PARM-LIST: Required.

trtch Valid values are “COMP”, “NOCOMP”, or “DEFAULT”.

FUNCTION: To allow the user to specify if compression is to be used for a specific output data. This value may also be specified in the DCB keyword as a positional parameter.

NOTE: If “DEFAULT” is specified, the actual usage of compression will be based on the default defined in the system for the device in use.

R-CODE: Not affected

6.3.36 UNIT

FORMAT: UNIT(device-name)

PARM-LIST: Required

device-name A 1 to 8 character generic or esoteric device name.

FUNCTION: To supply the unit name for the allocation. Cataloged input data sets do not require the UNIT keyword. The UNIT value may be generic device type (i.e. 3380) or a esoteric device name (i.e. SYSDA) or a specific device address (i.e. 481).

R-CODE: Not affected

6.3.37 VOLCOUNT

FORMAT: VOLCOUNT(number)

PARM-LIST: REQUIRED

number The number of output volumes that an output data set is expected to generate. The maximum value is 255.

FUNCTION: To instruct SMFUTIL that during dynamic allocation, the volume count specified is to be used. Note that output NEW and MOD data set with a UNIT specification of any type of tape device automatically get a default VOLCOUNT value of 255. The VOLCOUNT keyword would only be required for MOD'ing to cataloged data set without the UNIT parameter specified.

R-CODE: Not affected

6.3.38 VOLUME

FORMAT:	VOLUME(vser[,vser,vser])
PARM-LIST:	REQUIRED
vser	A valid 1 to 6 character volume serial that an input data set resides on or that an output data set is to be placed to.
FUNCTION:	To provide the volume serial for a data set. More than one volser may be specified. If more are required than can fit on one line, multiple VOLUME keywords may be specified. It is strongly recommended that the VOLUME command not be used unless absolutely necessary. Input and output data sets should be accessed through the catalog. Note that the VOLUME keyword is mutually exclusive with the definition of a VSAM data set for input.
R-CODE:	Not affected

6.3.39 VOLNUM

FORMAT:	VOLNUM(number)
PARM-LIST:	REQUIRED
number	The volume of a multiple volume set that input or output is to be started on. The maximum value is 255.
FUNCTION:	To identify the volume within a multiple volume data set that processing is to begin with. The VOLNUM keyword should only be used with input data sets or non-cataloged output "MOD" data sets. SMFUTIL will automatically determine the volume to start "MOD" processing on for cataloged "MOD" output data sets.
R-CODE:	Not affected

6.4 CONTROL STATEMENT EXAMPLES

It is important to understand the working relationship of the INCLUDE, EXCLUDE, SUBSET, JOBNAME, TSOUSER, BGUSER, and USERID keywords. SMFUTIL begins operation with all record types eligible for copying. When any of the above control keywords are encountered for the first time, the ALL flag is turned off and individual flags (for each record type) are turned on to select the record or off to exclude it. All subsequent encounters with the above keywords merely turn on or off the individual record type flags. The following examples will help to illustrate the interaction of these keywords:

EXAMPLE 1.

SUBSET(JOB)

ACTION: The ALL flag is turned off and record types with pertaining to batch jobs are turned on.

EXAMPLE 2.

JOBNAME(job-name) EXCLUDE(0-255) INCLUDE(30)

ACTION: The ALL flag is turned off and record types with job name capability are turned on. All record types are then turned off and only type 30 records are turned on. This will result in only type 30 records being selected for the specified job-name. The same result could be obtained by specifying:

JOBNAME(job-name) INONLY(30)

EXAMPLE 3.

JOBNAME(job-name) EXCLUDE(4,5)

ACTION: The ALL flag is turned off and record types with job name capability are turned on. Type 4 and 5 records are then turned off. This will result in all record types for jobs being selected for the specified job name except types 4 and 5.

EXAMPLE 4.

EXCLUDE(4,5) JOBNAME(job-name)

ACTION: This would probably be in error. The ALL flag is turned off by the EXCLUDE and all record types except 4 and 5 are turned on. Record types with job name capability are then turned on again by JOBNAME (they were already on) and this would include types 4 and 5 being turned on again even though they had been turned off by the EXCLUDE. This will result in all record types that have job name capability being selected for the specified job name only. All other record types would be copied without job name validation.

EXAMPLE 5.

DETAIL INCLUDE(4-5,30,128-255)

ACTION: The detail record report is requested. Record types 4, 5, 70 through 79, and 128 through 255 are selected for processing. All other record types will not be selected. A colon may be substituted for the - in the above syntax (i.e. 128:255 is equivalent to 128-255).

EXAMPLE 6.

DETAIL TYPETODD(70:79-RMFFILE) INCLUDE(0-255) COPYBOTH

ACTION: The detail record report is requested. Record types 70 through 79 are directed to a data set defined on DDNAME RMFFILE. All record types are selected on again (all but 70 through 79 were turned off by TYPETODD). COPYBOTH is selected to indicate that records directed to a specific user DDNAME (70 through 79) are also to be copied to the standard output data set(s).

EXAMPLE 7.

CLEAR

ACTION: The input SYSUT1 SMF MANx data set will be cleared of data. No other parameters are allowed because no data movement will be performed. All required data must have been copied in a previous step.

EXAMPLE 8.

DATE(2002090:2002091)

ACTION: The date range of 03/30/2002 through 03/31/2002 is selected. Only records containing a date in this range will be eligible for selection.

EXAMPLE 9.

GDATE(03/30/2002:03/31/2002)

ACTION: The date range of 03/30/2002 through 03/31/2002 is selected. Only records containing a date in this range will be eligible for selection.

7. OPERATIONAL CONSIDERATIONS

Operationally, SMFUTIL is simple to use. There are some characteristics and limitations within SMF that need to be understood to effectively utilize all of the power of SMFUTIL. This section deals with these areas and explains how to effectively utilize SMFUTIL to overcome the limitations of the SMF reporting system.

Normally, SMF is archived to tape data sets. These can run to many volumes for each data set. It is critical that the volume count specification be supplied in the DD statements 'VOL=' parameter for each output data set that may create more than 5 output volumes. This will prevent 837-08 ABENDs. See the example execution JCL for the way to specify the volume count parameter. Note that DDA type dynamically allocated data sets automatically have 255 set in as the volume count.

To be selected for processing, a record must meet all selection criteria specified by the user. The exceptions to this rule are for job name and userid. Normally, when a keyword is used specifying a userid or job name, SMFUTIL automatically turns on all records types that are valid for userid or job name consideration. The user may select other record types explicitly via the INCLUDE keyword or implicitly via the SUBSET keyword. In this situation, SMFUTIL assumes that the user wants the additional records, and, not being able to screen them for the specified userid or job name, SMFUTIL will pass them to output.

7.1 LARGE BLOCK INTERFACE SUPPORT

At Version 8 and up, SMFUTIL supports the IBM Large Block Interface (LBI) specification. In the defaults module the LBI parameter **enables** the interface and the LARGEBLK keyword controls if output LBI datasets are to be created by default.

Basic support for the LBI interface is controlled by the LBI keyword in the SMFUDFLT module. As long as the LBI=YES default specification is retained as shipped (or the LBI keyword is specified at the beginning of the input control set) input datasets are automatically supported for LBI block type input. SMFUTIL will be able to automatically detect and read any dataset containing SMF data, be it a standard or LBI type, without the user having to make any special provisions. Note this is not true prior to version 8. Previous versions of SMFUTIL will not be able to read datasets created with LBI type large blocks.

As shipped, SMFUTIL will output standard type blocked datasets even when LBI is enabled via the LBI defaults keyword or LBI keyword in an execution because the LARGEBLK keyword is set to NO. A site may instruct SMFUTIL to use the LBI output for all or an individual output dataset. To override and create LBI dataset on all output datasets, change the LARGEBLK=NO specification to LARGEBLK=YES in the SMFUDFLT defaults module. **This is normally not recommended.** To override and create LBI datasets on all output datasets from a particular execution of SMFUTIL the user can specify the LARGEBLOCK keyword at the beginning of the control input set.

An individual DDA block dynamic allocation request can be set to output an LBI type dataset by specifying the DDALARGEBLOCK keyword within the DDA block.

Caution should be used when creating output datasets utilizing the LBI option as other programs that read SMF data may not be able to read the LBI type blocks. LBI is most appropriate for large scale long term SMFUTIL ARCHIVE storage such as monthly archives on 3590 type tape devices where only SMFUTIL is expected to extract SMF data from these archives.

When the LBI interface is requested for an output dataset, any block size specification request is ignored and the LBI block size in the DCBE is set to the largest physical block the device will accept as reported by the DEVTYPE macro. This is currently 64K for 3480 and 3490 devices and 256K for 3590 devices.

7.2 SMFUTIL APF AUTHORIZATION

SMFUTIL is link-edited with an authorization code of 1. This allows it to run in an authorized state if it is executed from an authorized library. The advantage of running SMFUTIL in an authorized environment is: 1) During a CLEAR step, SMF is informed that the cleared data set is again available via a cross memory post from SMFUTIL; and 2) Dynamically allocated data sets that require tape mounts will not give the operator a cancel option ('REPLY NO'). They will appear to the operator to be normal mount messages. In addition, SMFUTIL dynamic allocations will be able to wait for volumes, data sets, and/or devices just as JCL allocations do. If a DDA required volume or data set is currently not available, a non-authorized SMFUTIL execution will terminate with a dynamic allocation failure.

Note that most environments will require that a CLEAR be done authorized. This is because the system is not able to 'discover' the cleared data set on it's own. The cross memory post from SMFUTIL is essential to return the cleared data set for use.

7.3 SMFUTIL EXECUTION SERIALIZATION

The user can request that SMFUTIL serialize the execution of batch job via an execution parameter. The syntax is as follows;

PARM='SYNC([LOCAL|GLOBAL],name,[WAIT|TERM(r)]'

Where:

LOCAL	A local "SYSTEM" ENQ is issued to serialize the request within the LPAR.
GLOBAL	A global "SYSTEMS" ENQ is issued to serialize the request across the SYSPLEX.
name	A 1 to 8 character unique resource task name to identify this task request. Each job containing the same name will be serialized. Only one job will be allowed to run at a time.
WAIT	The job is to wait until the resource defined by 'name' is available and then begin execution.
TERM	The job is to terminate immediately if the resource defined by 'name' is already in use by another job.
r	The return code to issued if the job is terminated because the resource is not available. Only '0', '4', '8', and '12' are allowed.

For the GLOBAL request to function, the local SYSTEMS ENQ must be propagated around the ring to all systems via GRS. The ENQ QNAME is 'SMFUTIL' and the RNAME is 'SMFUSYNC.name' where name is the task name supplied by the user. The RNAME length is 17 bytes.

Note that GLOBAL and LOCAL should not be intermixed for the same task name. The same type (GLOBAL or LOCAL) should be used for every instance that a specific task name is used.

A typical use of the SYNC parameter is to prevent multiple copies of the same job or started task from running at the same time. Since the synchronization takes place before any initialization or processing is done by SMFUTIL other types of lockouts by dynamic allocation are prevented.

EXAMPLE 1: PARM='SYNC(LOCAL,MNTHCOPY,TERM(8))'

This job is defined as performing a task called MNTHCOPY on the local system. If another job is already running with this same task name the new copy of the job will be terminated with a return code of 8.

EXAMPLE 2: PARM='SYNC(LOCAL,DUMPCALL,WAIT)'

This job is defined as performing a task called DUMPCALL. This could be the started task that performs the DUMPCLEARALL process. SMFUTIL and the IEFU29 exit distributed with the product normally protect against multiple starts of the dump process but this only happens during the initialization phase of SMFUTIL. If a delay were encountered during processing, such as a tape unavailable for mounting, it is possible that an additional SMF MANx dataset could fill up causing an additional switch to a new MANx dataset. This would cause a new dump task to be started. It is important that this new task be allowed to run at some point as the currently executing task would not know to go back and dump/clear the additional dataset because the current task had already completed scanning the MANx datasets. The SYNC parameter shown above would cause the second task to wait, without holding any resources, until the first task had completed and then the second task would execute normally.

7.4 DYNAMIC ALLOCATION OF SYSPRINT OUTPUT DATASET

SMFUTIL requires the SYSPRINT dataset be available for each execution. If not found in the execution JCL, SMFUTIL will attempt to dynamically allocate it. The SMFUDFLT keyword SYSPDSN normally specifies '*' indicating the output is to be sent to a JESx spool dataset allocated to the default message class. This can be change to a single character output class if desired.

The user can request that SMFUTIL direct the SYSPRINT output to a disk dataset. There are three ways to inform SMFUTIL of the dataset information to be used to dynamically allocate a dataset to receive the SYSPRINT output. The first is to specify it in the SMFUDFLT module via the SYSPDSN, SYSPVOL, SYSPUNIT, SYSPDISP, SYSPMCLS, SYSPDCLS, and SYSPSCLS parameters. **This is not recommended as it affects all executions of SMFUTIL that do not contain a SYSPRINT DD card in the JCL.** The SMFUDFLT module should be left with the shipped default of **SYSPDSN='*'** or another single character defining an output class of your choice to ensure SYSPRINT is normally allocated to a spooled output dataset.

The second way to define the output dataset is via an execution parameter. The syntax is as follows;

```
EXEC PGM=SMFUTIL,PARM='SYSPRINT(dsn,[disp],[sfx],[volser],[unit])'
```

Where:

dsn	A valid dataset name to be allocated or a 1 character SYSOUT class. This may be a relative GDG structured dsname, if desired. If a single character is specified, it will used as a SYSOUT class for a JESx spool dataset.
disp	A disposition of NEW, MOD, or OLD. This may be abbreviated N, M or O. If a valid disp value is not specified, NEW will be assumed.
sfx	A suffix indicator of D, DT or DTX. Ignored when a relative GDG data set name is specified. See below for a description of the possible suffix types and their formats.
volser	A 1 to 6 character volume serial number to be used.
unit	A 1 to 8 character unit name to be used for the allocation.

NOTE: SYSPRINT may be abbreviated SYSP or SP.

EXAMPLE: PARM='SYSPRINT(SYSX.DUMPALL.SYS%SID.LIST(+1))'

The executing system SMF identification is used to complete the data set name and the dataset will be allocated to the unit name specified in the SYSPUNIT value of the SMFUDFLT module. Note the use of the SMFUTIL specific symbolic '%SID' in the string. This unique syntax prevents the system from "stripping down" the execution parameter which would be the case if the normal '&SID' parameter were used. The result is the same. The string '%SID' is replaced with the executing system SMF System Identification prior to allocation of the dataset.

Three types of suffixes are allowed on the SYSPRINT dsname. They are as follows:

```
D   = Date only           Format ".Jyyyyddd"
DT  = Date/Time          Format ".Jyyyyddd.Thhmmss"
DTX = Date/Time Extended Format ".Jyyyyddd.Thhmmss.Hth"
```

Certain restrictions apply when a suffix is requested. First, any suffix will be ignored when a relative GDG type dataset name is detected. In addition, the disposition of a suffixed dataset must be NEW. Certain length limits apply to the base data set name specification when a suffix is applied to ensure the overall data set name is 44 characters or less. For 'D' suffixes, the data set name must 35 characters or less. For 'DT', it must be 27 or less. For 'DTX' it must 23 characters or less.

It is the users responsibility that the dataset name requested is unique, either by utilizing a relative GDG or by requesting an adequate suffix be affixed to the dataset. The granularity of the suffix must be fine enough to ensure uniqueness for the frequency of the jobs execution (i.e. don't use 'D' for a job run multiple times per day).

The third way to define the SYSPRINT data set name information is with a SYSPARM DD statement in the JCL. SYSPARM defines a sequential dataset with a logical record length of 80 bytes. It format is as follows:

```
//SYSPARM DD DSN=USER.PARMLIB(PARMSYSP),DISP=SHR
```


Where the member **PARMSYSP** contains one or more of the following statements starting in position 1:

```

SYSPDSN=dsname.string.for.sysprint
SYSPDISP=NEW (or MOD or OLD)
SYSPSUFFIX=D (or DT or DTX)
SYSPVOL=volser
SYSPUNIT=unitname
SYSPMCLS=mgmtclass
SYSPDCLS=dataclass
SYSPSCLS=storclass

```

An omitted statement leaves the existing default value unchanged. If a statement is specified with a null field (blanks after the '=' sign, the parameter will be nullified. For example a statement of 'SYSPVOL= ' would cause the volume string to be set to blanks and any eligible volume on the defined unit will be considered for the allocation.

The SYSPARM method has several advantages over the execute parm field. It can specify the SMS class definitions and it can nullify parameters where the execution parm field cannot. Since the exec parm field is limited to 100 characters, using the SYSPARM avoids any future problems with parm usage. In addition, in most shops, changing PARMLIB members is usually much easier than changing production JCL jobs.

Note that SMFUTIL GDGCONTROL does not apply to this dataset if a GDG is used. Normal GDG processing will occur. The use of the SYNC parameter to synchronize runs is recommended to prevent lockouts due to the same GDG base index being used on multiple jobs or started tasks.

7.5 EXECUTION REGION SIZE REQUIREMENTS

SMFUTIL can utilize a large amount of virtual storage when copying data to multiple datasets and performing many complex operations and data splits. The majority of this storage will be “above the line” as all input and output I/O buffers are in 31 bit addressed storage. This greatly relieves the burden on the private area below the line. The best solution for a region specification is to code REGION=0M on the EXEC statement and simply let SMFUTIL obtain what it needs. If this is not allowed in your environment you may calculate a region as follows.

For a normal copy operation, SMFUTIL will operate safely in about 2048K (2 megabytes), depending on the block size of input and output data sets. Some functions obtain memory dynamically and may require substantially more region size. The operating system will obtain output buffers for each output data set. The amount required for each depends on the block size and the number of buffers to be used (BUFNO in the DCB parameter for non-VSAM data sets) and on the VSAM buffer specifications used (VRBUFF/VWBUFF/VCBUFF). Non-VSAM defaults to the value defined by the SRBUFF, SWBUFF, and SUBUFF definitions in the SMFUDFLT module. The VSAM defaults are specified in the SMFUDFLT module via keywords VRBUFF, VWBUFF, and VCBUFF. For a TYPETODD keyword specified ddname, add 32K times the value for SUBUFF for each sequential output DDNAME in the JCL. For normal output data sets, add 32K times the value for SWBUFF. For sequential input, add 32K times the number specified for SRBUFF for each input data set. Remember that the system does not obtain the buffers until the data set is opened. This means that a run with a xSPLIT and/or xSPLITDP function will require a larger region when the split occurs. Make sure not to lower the region size to a value based on a ‘no split boundary found’ execution. In addition, if the DELDUP function is to be used, add 320K. Likewise, add 24K if the DATESCAN function is used.

The following formula is complex but will work:

$$\begin{aligned} \text{REGION} = & (\#VSAMIN \times \text{VRBUFF} \times 32) + (\#VSAMOUT \times \text{VWBUFF} \times 32) \\ & + (\#SEQIN \times \text{SRBUFF} \times \text{LBIBLSZ}) + (\#SEQOUT \times \text{SWBUFF} \times \text{LBIBLSZ}) \\ & + (32 \times \text{DDRETAIN}) \text{ [if DELDUP used]} \\ & + 24 \text{ [if DATESCAN used]} \\ & + 756 \text{ [base region size]} \end{aligned}$$

The value LBIBLSZ is the block size (in K) that will be used for Large Block Interface datasets. If LBI is not used, substitute 32 for this value. If LBI is used, the value will be 64 for 3480 and 3490 devices and 256 for 3590 devices.

If the VxBUFF or SxBUFF keywords parameters are used to specify large buffering values, a substantially larger region may be required. Test executions will be required to determine the region size needed. The SYSLOG DD output will indicate the various totals of dynamic GETMAIN areas that were used.

A normal region size for a production MANx dump step that contains a split on two different boundaries (SYSUT2, 2D, 3, and 3D output data sets) would require 3600K as shipped. This region requirement can be reduced by reducing the buffering supplied to the VSAM and QSAM processing but the trade off in degraded performance and increased run time will be significant.

Note that for consistency, a region size of 4096K can be used as a safety valve. For any general purpose JCL procedure that is established in a PROCLIB for users to use in retrieving SMF data, a large region should be specified.

7.6 SYMBOLIC SUBSTITUTION OF SYSTEM SYMBOLS

If the SYMBOLS parameter of the SMFUDFLT module is set to “YES” or if the user supplies the SYMBOLS keyword as the first control statement of the execution, automatic substitution of defined system symbolic terms will be performed via the ASASYMBM utility provided it is available in the system. Control statements will be printed in the normal way. When substitutions are made the altered text will also be printed. All symbolic terms used must conform to IBM requirements and must be defined in the system. Undefined symbolic terms will result in syntax errors.

For example, if the system symbolic “&SYSNAME.” were defined as “SYSA” in the system, the string:

DSNAME(SYS1.&SYSNAME..DATA)

would result in a substituted string of

`DSNAME(SYS1.SYSA.DATA)`

Note that symbolic substitutions must yield valid SMFUTIL keywords and structures.

Caution must be exercised when mixing system symbolic terms and SMFUTIL defined symbolic terms within the same execution. For example, the string `&DID` is a valid SMFUTIL symbolic for defining the SMF data system ID. This may or may not be a defined system symbolic in your environment.

7.7 INSTALLATION DEFAULTS OVERRIDE AT EXECUTION

During installation, a set of execution defaults are defined via the SMFUDFLT source module. See the section titled “MODIFICATION AND CUSTOMIZATION” on page 199 for more information. While most of these default values may be overridden via SYSIN control statements, not all values may be altered and there may be times that a different set of default options may be desirable. SMFUTIL execution defaults may be overridden via a PARMLIB data set member containing default setting information. These values are input during initialization and will replace the SMFUDFLT installed values prior to the execution defaults being listed on the SYSPRINT data set. The verb statements are identical to the parameters used in the SMFUDFLT module. Verb statements begin in column 1, are followed by an equal sign (=) and which is followed by the default value desired. For example:

```
AEXPIRE=90
ARCHIVDB=SMFUTIL.ARCHIVE
```

The values are coded just as they are in the SMFUDFLT module except text strings (such as data set names) are never surrounded by quote marks. No other text may be on the same line as the value text. Comment lines may be inserted by placing an `*` in column 1.

7.7.1 SPECIFYING THE PARMLIB DATA SET VIA SMFUDFLT

The “PARMLIB” parameter of the SMFUDFLT module can be used to specify a parmlib data set that is to contain members for overriding the defaults information and/or the license code information.

7.7.2 SPECIFYING THE PARMLIB VIA JCL

The PARMLIB data set may also be specified in the JCL as follows:

```
//PARMLIB DD DSN=parmlib.dataset,DISP=SHR
```

This specification will override the PARMLIB keyword in the SMFUDFLT module and any PARMLIB specification in the “PARM=” field of the EXEC statement. As shipped, SMFUDFLT contains blanks for the PARMLIB. This situation will cause SMFUTIL to rely on the license information in the load module license table and the defaults information in the SMFUDFLT module assembled at installation time.

7.7.3 SPECIFYING THE MEMBER NAMES VIA SMFUDFLT

The DFLTMBR and LCNSMBR parameters of the SMFUDFLT may be used to specify permanent member names to be used for locating defaults and license information. As shipped, these parameters are blank meaning no member will be read.

7.7.4 SPECIFYING THE MEMBER NAMES VIA “PARM=”

Because the PARMLIB, DFLTMBR and LCNSMBR default parameters are utilized prior to the SYSIN control stream being processed, they may not be overridden by control keywords but they may be overridden by the “PARM=” field of the EXEC statement. The syntax is:

```
...EXEC PGM=SMFUTIL,PARM='PARMLIB=plib.dsnt,LCNSMBR=mbrname,DFLTMBR=mbrname'
```

These specifications will supercede the SMFUDFLT specifications in the load module.

7.7.5 SPECIFYING THE PARMLIB AND MEMBER VIA JCL

The PARMLIB data set may also be specified in the JCL with a member name as follows:

```
//PARMLIB DD DSN=parmlib.data.set(mbrname),DISP=SHR
```

This specification will override the PARMLIB keyword in the SMFUDFLT module and any PARMLIB specification in the “PARM=” field of the EXEC statement. The supplied member may contain BOTH the license information and defaults information. Each must have a special section header of [LICENSE] or [DEFAULTS] as follows:

```
*
* BEGIN DEFAULTS SECTION
[DEFAULTS]
ARCHIVDB=SMFUTIL.ARCHIVE
WKSTART=MON
*
* BEGIN LICENSE SECTION
[LICENSE]
CODE=12345567812345678
DATE=2004395
PRODUCT=SMFUTIL
VER=800
CPUSER#1=12349672
.
. (remainder of license information goes here)
.
* END OF LICENSE DATA
```

Note that selected default items maybe overridden as desired but the entire license must be defined if a member or a [LICENSE] section of a member is supplied.

Note that the PARMLIB DD statement may be an in stream JCL input dataset as follows:

```
//PARMLIB DD *
[DEFAULTS]
WKSTART=SUN
/*
```

If both a LICENSE and DEFAULTS section are specified they must be preceded by the appropriate header as described above. The DEFAULTS section may contain only the specific items to be overridden but the LICENSE section, if present, must be complete. The entire license, as supplied by ASPG, must be specified.

7.7.6 DEFAULT OVERRIDE EXCLUSIONS

NOTE: The following default verbs may not be overridden via the PARMLIB specification list for the reason given:

- PARMLIB** – Cannot be self defining. The PARMLIB is already being read.
- DFLTMBR** – The member is already being used.
- LCNSMBR** – The member is already being used.
- SUBSYSID** – The SSCT name may only be defined once and never changed.
- LOGLEVEL** – Safety reasons. Higher values of LOGLEVEL will cause large listings.

Inclusion of any of these keywords in the [DEFAULTS] override section will result in an SMFU246 error message and a termination of the execution.

These parameters may be overridden via the EXEC PARM field from their original definitions in the SMFUDFLT defaults source member assembled at INSTALL time.

7.7.7 LICENSE PARMLIB AND MEMBER VALIDATION

Whatever method is used to supply the PARMLIB data set name and license member specifications to SMFUTIL, they are used to locate license information that overrides what is already present in the license table in the load module. The entire license (all items from the report) must be included in the parmlib member. In the event the PARMLIB and/or MEMBER specified are not valid or available to the execution, an error message is issued and processing of the license override is bypassed. Normal execution continues as if no override had been requested. If the load module license table contains a valid and current license it will be honored. If not, execution will be aborted.

7.8 SYMBOLIC SUBSTITUTION FOR DEFAULT VALUES

As coded in the SMFUDFLT table, default items may not contain embedded system symbolic definitions however defaults supplied via the PARMLIB member process may contain symbolic definitions. If the system symbolic substitution module ASASYMBM is available and the SYMBOLS parameter is set to YES (via SMFUDFLT or a previous default member line of "SYMBOLS=YES"), symbolic substitution will be performed on default parameters as they are read in from the member. This allows you to use system defined symbols in text items such as data set names or unit definitions. Note that you may specify the "SYMBOLS=" verb multiple times to turn symbolic substitution on and off as needed during the processing of the entire defaults member. The final setting of the SYMBOLS verb will determine if symbolic substitution will be performed on the input control keywords specified on SYSIN.

7.9 RECORD SUBTYPES

7.9.1 SELECTING RECORDS BY RECORD SUBTYPE

Some record types have multiple subtypes produced under one record type number. An example of this is record type 30. There are two ways for SMFUTIL to select records based on the subtype value found within the record. The SUBTYPE keyword specification is a fast and easy way to select on a few subtypes of a specific record type that was previously specified via an INCLUDE or INONLY keyword specification. A control statement stream to select only subtype 5 of the type 30 records for Userid SYSXYZ would look like this:

```
USERID(SYSXYZ) INONLY(30) SUBTYPE(5)
```

Note that the INONLY keyword is used instead of an INCLUDE keyword. This is because it follows the USERID keyword which has already turned on only record types that are valid for selection via USERID (i.e. types 4, 5, 30, etc.). The INONLY will now turn off all types except type 30. The SUBTYPE keyword specification will cause only subtype 5 of the selected type 30 records to continue with processing. All other subtypes of type 30 records for this userid will be discarded.

As an alternate method, the TYPETODD keyword allows subtype filters to be placed on specific record types. In this way subtypes of specific record types can be placed to a subset data set while large amounts of data is passed to the archive files. For example, to place all subtype 5's of all type 30 records to a data set pointed to by a DDNAME of T30ST5 you could specify:

```
TYPETODD(30(5)-T30ST5) INCLUDE(0-255) COPYBOTH
```

The TYPETODD initially turns off all but the type 30 records for selection. The INCLUDE statement turns all records back on. The specification of COPYBOTH will cause all type 30 records to be selected for output to the primary SYSUTxx output data sets and the TYPETODD statement will select subtype 5 of all type 30 records to go to the specified data set. Please note that if COPYBOTH were omitted no type 30 records would be selected for output to the primary output data sets.

Please note that the two methods of specifying subtype values for selection are mutually exclusive and may not be mixed in one execution of SMFUTIL.

7.9.2 SUBTYPE SELECTION CRITERIA

To be considered valid for selection on a subtype specification, the record type must be marked as subtype compatible in the record account table CSECT (SMFUTLRT). See the section entitled "MODIFYING THE RECORD ACCOUNTING TABLE" on page 209 for details on adding non-IBM record types that have subtype capability to the table.

Most IBM records, that have subtype capability, contain a two byte binary number in position 22 (hex 16). All known IBM record types that have subtypes are so marked in the table. Other record types produced by user code or other vendor products will have to be added. In some instances the subtype specification may be a single character code instead of a two byte binary number. To allow for this situation, you may specify to SMFUTIL in the record account table module that a particular record's subtype value is only one byte long and/or is at a location other than the standard IBM location of decimal 22. The SMFUTIL processing routines will only accept a decimal number for a subtype specification. To specify a character value as a subtype for selection, substitute the character's decimal value. As an example, for a record subtype of 'A' specify a SUBTYPE(193) keyword. The same is true for the specifications in the TYPETODD subtype field.

7.10 USERID RECOGNITION

7.10.1 SOURCE OF USERID

Under a standard (unmodified) IBM OS/VS operating system of any type, the userid is not automatically placed in the SMF record. The standard SMF header for records pertaining to a particular user has an 8 byte field reserved for the userid (SMFxUIF). This field is taken from the JMRUSEID field in the common SMF exit parameter list. It is up to the installation to fill in the JMRUSEID field area in any of the SMF exits. This can cause a problem, because this 8 byte area is often already used for other purposes. This field may be used by some non-IBM program products to pass control block addresses or other information between SMF exits. Most software vendors have software updated versions that correct this situation. Before overlaying this area with a userid, it would be advisable to check with the vendor of any software that has SMF exits installed on your system to insure that their exits are not utilizing the field.

7.10.2 SECURITY SYSTEMS AND USERID'S

Security packages such as RACF from IBM and ACF2 or TOP SECRET from Computer Associates will fill in the userid field for you. If you are utilizing the ACF2 security system from Computer Associates (formerly SKK and then UCCEL), for security authorization, you will not need to code any SMF exit to fill in the userid. ACF2 will do it for you. There is an initialization parameter called STAMPSMF in the ACF2 Field Definition Record (FDR) or the GSO OPTS record (depending on what ACF2 release you are on) that can be set to cause the ACF2 userid to be placed in the SMF records. The default for this option is "NO" because of the difficulties outlined above. If you have CA7 from Computer Associates installed, be sure to put on both zaps as outlined above to ensure that the ACF2 records have their RDRDATE properly restored or some ACF2 reports will have zeros for date of job on reader. Other security packages may have similar options. Check with your vendor to make sure.

7.10.3 TYPE 30 RECORDS AND USERID'S

Special consideration must be given to type 30 records because they contain the userid in an offset identification section, not in the header section as the older style records do. This identification section contains the normal userid field that is filled in from the common exit parameter list as described above. It also contains a userid field that is completed by RACF if it is in the system. Most security packages fill in this RACF userid field properly without modification.

7.10.4 USERID MASKING

All of the control keywords that can specify a userid string operate in two ways. First, if a userid string of 8 characters is specified, that this the only userid that will be selected on. Second, if a userid of less than 8 characters is specified, the userid string acts as a mask and all Userid's that begin with the mask will be selected on. This means that to select only records for a userid that is less than 8 characters long, it is necessary to pad the userid in the specification to the right with blanks. A specification of an '*' (an asterisk) in any position in the userid string will result in a wild card specification. Any character will be accepted in that position.

7.10.5 USERID CHECKING AND OTHER RECORD TYPES

When a control keyword is encountered that selects records pertaining to a certain userid or userid mask, SMFUTIL automatically turns on selection for the record types that pertain to the type of selection requested (Batch, TSO, etc.). The records that are turned on all have a characteristic that they may contain a userid string. Other records produced by SMF do not contain userid strings and, thus, it would not be logical to attempt to select or discard these records based on userid. SMFUTIL, however, does allow these other record types to be specifically selected by other control keywords (INCLUDE, SUBSET, etc.) during the same run that is screening the userid type records for the specified userid(s). The other records specifically selected will not be checked for userid, but will pass directly to output, providing they meet other selection criteria (i.e. DATE, TIME, etc.).

7.11 JOBNAME RECOGNITION

Several certain record types contain job name information. Types 4, 5, and 6 are typical records that contain the job name in the header information (in a fixed location). Type 30 contains the job name in an offset location that may vary in specific local from release to release.

7.11.1 TYPE 30 RECORDS AND JOB NAMES

Special consideration must be given to type 30 records because they contain the job name in an offset identification section, not in the header section as the older style records do. The system fills in this field automatically. SMFUTIL differentiates between type 30 record job name location and other record types automatically. The job name location in the 30 record is located indirectly via a "triplet" that contains an offset from the beginning of the record to the identification section. The actual location of the identification section may be changed from release to release but the offset to the triplet does not change. This allows SMFUTIL to dependable locate the job name despite changing system levels of even mixed system data.

7.11.2 JOBNAME MASKING

All of the control keywords that can specify a job name string operate in two ways. First, if a job name string of 8 characters is specified, that this the only job name that will be selected on. Second, if a job name of less than 8 characters is specified, the job name string acts as a mask and all job names that begin with the mask will be selected on. This means that to select only records for a job name that is less than 8 characters long, it is necessary to pad the job name in the specification with blanks to the right. A specification of an '*' (an asterisk) in any position in the job name string will result in a wildcard character specification. Any character will be accepted in that position.

7.11.3 JOBNAME CHECKING AND OTHER RECORD TYPES

When a control keyword is encountered that selects records pertaining to a certain job name or job name mask, SMFUTIL automatically turns on selection for the record types that pertain to the type of selection requested (Batch, TSO, etc.). The records that are turned on all have a characteristic that they contain a job name string. Other records produced by SMF do not contain job name strings and, thus, it would not be logical to attempt to select or discard these records based on job name. SMFUTIL, however, does allow these other record types to be specifically selected by other control keywords (INCLUDE, SUBSET, etc.) during the same run that is screening the job name type records for the specified job name(s). The other records specifically selected will not be checked for job name, but will pass directly to output, providing they meet other selection criteria (i.e. DATE, TIME, etc.).

7.12 ACCOUNT CODE RECOGNITION

SMFUTIL supports selecting records based on accounting information supplied in the first accounting parameter position (on the JOB or EXEC card) and up to 8 bytes in length where the record supports that many (i.e. the type 26 record only supports 4 bytes in the normal position). When specifying an account code to be selected on via the ACCOUNT keyword parameter, an '*' may be specified in any character position to indicate that any character in that position is acceptable. If less than 8 characters are specified, the string will be compared only for the length specified and is therefor an account code prefix. Any record containing an account code that begins with the specified prefix will be selected.

7.12.1 SOURCE OF ACCOUNT CODE

Under a standard (unmodified) IBM OS/VS operating system of any type, the account number may be specified on the JOB card and/or on the EXEC card of any step in the JCL. The following IBM record types may (or may not) contain accounting information:

- 4 The type 4 record (step termination) will contain any account number specified on the EXEC card for the step. IF NO ACCOUNT NUMBER IS SPECIFIED ON THE EXEC CARD, THE TYPE 4 RECORD WILL CONTAIN NO ACCOUNTING INFORMATION AT ALL.
- 5 The type 5 record (job termination) will contain any account number specified on the JOB card for the job. IF NO ACCOUNT NUMBER IS SPECIFIED ON THE JOB CARD, THE TYPE 5 RECORD WILL CONTAIN NO ACCOUNTING INFORMATION AT ALL.
- 20 The type 20 record (job initiation) will contain the accounting information from the JOB card.
- 26 The type 26 record (job purge) will contain the first 4 characters of the account number from the JOB card.
- 30 The content of the type 30 record depends on the subtype. The subtype 1 (job start) will contain accounting data from the JOB card. The subtypes 2 (interval end) and 3 (step termination within interval) will contain no accounting data. The subtype 4 (step total) will contain accounting information from the EXEC card. The subtype 5 (job termination) will contain accounting information from the JOB card.
- 34 The type 34 (TSO step termination) will contain accounting information from the EXEC card of the LOGON PROCEDURE if one was specified otherwise it will contain no accounting information.
- 35 The type 35 (TSO LOGOFF) will contain the accounting information from the JOB card generated at LOGON. The account code inserted into this pseudo JOB card is obtained from the SYS1.UADS entry for the userid (or from the ACF2 user id record if ACF2 is in use with the NOUADS option).

7.12.2 ACCOUNT CODE LOCATION WITHIN RECORDS

The SMFUTLRT module supports the specification of the method to locate an account code via the 'ATYPE' keyword on the SMFUDRTE macro statement in conjunction with the "ACCOUNT=" keyword to specify the initial location within the record. The following 'ATYPE' values are supported:

FIXED4	<p>The account code is a single 4 byte field directly locatable in the record and is pointed to by the ACCOUNT keyword. As an example:</p> <p>'ACCOUNT=(YES,DIRECT,,120,0),ATYPE=FIXED4'</p> <p>would indicate that the account code located at 120 bytes (decimal) into the record. This type is used in record type 26.</p>
FIXED8	<p>This type is identical to FIXED4 except that the account field length is 8 bytes. This type is not used by IBM but it is provided for compatibility with any user generated records that contain 8 byte account codes.</p> <p>If the SMF26NAC field is filled in (network account number), FIXED8 could be used on record type 26 with a specification of:</p> <p>ACCOUNT=(YES,DIRECT,,260,0),ATYPE=FIXED8</p> <p>to locate the 8 byte network account code. This would require changing the SMFUDRTE macro for record type 26 and reassembling the SMFUTLRT module.</p>
MULTIPLE	<p>The number of account codes is contained in a single byte followed by a multiple set of 1 byte lengths that precede each account code text. This type is used in record type 5 and is coded as:</p> <p>'ACCOUNT=(YES,DIRECT,,120,0),ATYPE=MULTIPLE'</p> <p>This indicated that the byte containing the number of accounting fields is located at location 120 (decimal) into the record and it is followed by a 1 byte length field followed by the account text. Only the first account field is checked for a match against the code specified in the ACCOUNT keyword.</p>
OFFSMULTI	<p>The number of account codes is contained in a single byte followed by a multiple set of 1 byte lengths that precede each account code text. The set of accounting fields is at separated from the start of the record by a single variable length field that must be spanned across to reach the account codes. The length of the variable length field is in a half word pointed to by the ACCOUNT keyword on the macro. This type is used in record type 4 and is coded as:</p> <p>'ACCOUNT=(YES,INDIRECT,,106,4),ATYPE=OFFSMULTI'</p> <p>This indicates that the length of the variable length field is in a half word located at 106 bytes (decimal) into the record. The start of section containing the accounting fields is computed by adding the length of the variable length section to the base address of the record. This yields the start of section containing the accounting information. The byte containing the number of accounting fields is then located at 4 bytes into the accounting section and it is followed by a 1 byte length field followed by the account text. Only the first account field is checked for a match against the code specified in the ACCOUNT keyword.</p>
TRIPLET	<p>The section(s) containing the account information are located via an OFFSET/LENGTH/NUMBER triplet as in record type 30. The OFFSET location contains the offset from the start of the record to the start of the first accounting section. Each accounting section will contain a 1 byte length value followed by the account code text. The location of the start of the accounting section triplet is indicated via the ACCOUNT keyword on the macro. This type is used in record type 30 and is coded as:</p> <p>'ACCOUNT=(YES,INDIRECT,,64,0),ATYPE=TRIPLET'</p>

This indicates that the triplet that points the first accounting section is located at 64 bytes into the record. The start of section containing the accounting fields is computed by adding the offset value located at the triplet location to the base address of the record. This yields the start of first section containing accounting information. Only the first account field is checked for a match against the code specified in the ACCOUNT keyword.

7.13 SELECTING RECORDS BY DATA SET NAME

Some record types have one or more fields in them containing data set names. SMFUTIL offers a simple way to locate records that deal with specific data sets via the DATA SET and/or XDATA SET keywords. Each accepts one or more data set name mask prefixes to be used as filtering criteria. Multiple specification of the DATA SET and XDATA SET keywords is supported. Each specification string is used as a prefixing mask to compare to the actual data set name within the record field. If a specific data set is desired, the mask should be padded to the right with a blank to cause it to match only the desired dsname. If an '*' is used in any position within the mask, it will cause any character in the corresponding position in the data set name to be acceptable. The following examples will illustrate the used of DATA SET. XDATA SET works identically except that it excludes records matching the mask.

EXAMPLE 1 DATA SET(SYS1.LINKLIB)

Only records containing 'SYS1.LINKLIB' would be matched.

EXAMPLE 2 DATA SET(SYS1)

Any record containing a data set name beginning with 'SYS1' would match.

EXAMPLE 2 DATA SET(SYS1.***LIB)

Data set name 'SYS1.TESTLIB' would match. So would 'SYS1.LINKLIB' but 'SYS1.TESTLIB1' would not because of the trailing blank in the mask.

If a record type has two data set name fields (DSNAME1 and DSNAME2 specified in the SMFUTLRT table), both fields will be compared to all masks supplied. For DATA SET specifications, the records will be considered matched and selected if either field matches any mask specification. For XDATA SET specifications, the record will be considered matched and excluded only if both fields match the same mask specification.

7.14 SMF DATA ORDER CHECKING

SMF data, as dumped from the SMF data sets, is not strictly in sorted order. This is due to the fact that the data is placed to the SMF system by a variety of different sources. Each source builds a record, fills in the SMF header with the current date and time, and then performs an SMFWTM function to put the record to SMF. Because different periods of time elapse between the time the time and date values are obtained and the record is actually placed to SMF, the resulting records may arrive somewhat out of order. These minor discrepancies are, for the most part, not harmful and may be ignored in sort order checking by utilizing the SORTTOLERANCE keyword. The reason this condition can be ignored is that the records from an individual source will be in sorted order because they were produced synchronously.

Conversely, a major problem in SMF sort sequence checking is the presence of type 2 (dump header) and 3 (dump trailer) records in the input data stream. These records are not placed to SMF, but are written directly to the dump data set during dump processing by the IBM dump utility IFASMFDP. The time and date stamp in these records is determined at the time the dump function is executed. This means the output data is totally out of sort sequence when it is created if the type 2 and 3 records are considered. By default as shipped, SMFUTIL will ignore the type 2 and 3 records because TYPE2=NO and TYPE3=NO are specified in the SMFUFLT defaults module. This will prevent these records from causing problems in the future.

Certain record types may appear to be badly out of sort when they are created. These are normally records that are event driven and are reporting multiple events. An example of this is the RTE record produced by the CA/IDMS R-PM product of Computer Associates, Inc. Each RTE record reports on a number of IDMS transactions. The time and date in the header for the SMF record to be written is inserted when the record is started, not when it is finished. If the number of transactions to be reported on is large and activity is light, a long period may elapse between construction of the record and its completion. The solution is to keep the blocking factor low for records reporting on infrequent events. In addition, you may want to contact the vendor of your software that is causing the offending records and request that they fix the problem. The SMF manual states that the time and date in the record header are to be the time and date the record was moved to SMF.

It is extremely difficult to achieve perfectly sorted ordering in SMF data for DELDUP purposes. This is due to the fact that some SMF records do not vary until far into the record for the same date, time, and record type. Just putting the records into date/time/type sequence is often not sufficient for DELDUP to find all duplicates. A more thorough sort is to sort on an additional key after the date/time/type specifications. This should be a binary sort key in ascending order. The problem with this is that you cannot have a total sort key longer than the shortest record in the data set. This could present a problem if you have short records present. One solution is to split the data into multiple data sets and clean up each data set individually. Member SMFSORT in the INSTLIB data set has an example sort execution for sorting on the additional key as indicated. The TYPETODD keyword may be used to direct specific record types to different output data sets for additional processing.

The only function that may require close to perfectly sorted input data is DELDUP. It checks to see if an input record is identical to any of the previous 10 records encountered. If it is, it discards the duplicate record and continues. In order to accomplish this, the input data must be well sorted. The design purpose of DELDUP was to create a function that would delete duplicate data caused by improper dumping procedures. Because such duplicate data is, by its nature, out of sort to begin with, a preliminary sort step to properly sequence the data prior to passing it through SMFUTIL is not out of reason. Note that by its nature, DELDUP is a high overhead function. It should not be used on a routine basis. Please see DELDUPRV as a possible alternative to DELDUP.

7.15 CHECKPOINT DATA SET – USES AND RESTRICTIONS

If the “SYSCKPT” DDNAME is present in the execution JCL for SMFUTIL, it will be used to track several items of information within and between executions of the program. This data set is a PDS with all of the space used for the directory. It should be understood that while this data set is a PDS, only the directory area is used. No actual data is written to the data set under the member names. Instead, the required checkpoint information is stored in the user data area in each directory entry. As a result of this technique, ISPF may show “DELETED” as the status of the members in the data set when it is view under option 3.1 (Library Utility). Do not be concerned. The members are not deleted and are not going to disappear. ISPF is just mistaken the checkpoint data for flag bits that it uses to detect members deleted by ISPF in another session or window.

The SYSCKPT data set serves a number of purposes. It should be noted that the checkpoint records are closely tied to the date/time stamps of the records in the input stream. For this reason, it is essential that type 2 and 3 records not be processed when a checkpoint data set is in use. They may cause checkpoint records to be written with invalid date stamps and cause unpredictable results in the current or future executions of SMFUTIL. The default settings in the SMFUDFLT module is ALLOW2=NO and ALLOW3=NO. These settings will tell SMFUTIL to filter out type 2 and 3 records before they are passed to the output phase. This effectively prevents these record types from interfering with the check pointing process.

No physical data is written to the data set under the member names. It will never require compressing or other maintenance. All checkpoint information is stored in the user data field of the directory entry for the member name of the checkpoint.

A unique checkpoint data set should be thought of as task oriented. It should be dedicated to a single production use and named accordingly. For example, ‘SYSX.SMFUDUMP.SYSA’ might be used to for the started task that is used to dump the SMF data sets on the system with SMF identifier ‘SYSA’. This data set should not be used for any other execution of SMFUTIL. If it is, certain checkpoint records will be changed that may cause the next dump execution to abort. Set up a unique checkpoint data set for each production job or task that uses SMFUTIL.

Ad-hoc executions of SMFUTIL would not normally require the use of a SYSCKPT data set at all.

7.15.1 CHECKPOINT BROWSER UTILITY

An ISPF Browser Utility, SMFCKPUI, is provided to allow the user to examine and understand the purpose of each of the records present in the SMFUTIL Checkpoint PDS.

7.15.1.1 CHECKPOINT ISPF INTERFACE - INSTALLATION

The SMFCKPUI module is link edited into a LNKLSxxx dataset when the INSTALL job is run. Member SMFUPRIM in the ISPLIB data set is an example of how to invoke the CHECKPOINT BROWSER ISPF function from an ISPF menu. Member SMFCLBDF of the INSTLIB data set is a TSO CLIST that invokes the LIBDEF function for access to the SMFUTIL panel and message data sets and then launches the SMFCKPUI ISPF utility. This CLIST is invoked via the “CMD” function in menu SMFUPRIM. Using the SMFUPRIM as an example, modify a local ISPF menu to invoke SMFCLBDF.

7.15.1.2 CHECKPOINT ISPF INTERFACE - ACCESS CONTROL

Access to the ISPF interface can be controlled on a userid basis via a table module called CKPUUSRS. By default, full access is allowed to all Userid’s. This table can be used to limit access of to allow certain userid’s read only access (no delete capability). To implement limited access follow the instructions in the CKPUUSRS table source module in the INSTLIB data set, re-assemble and re-link the table with SMFCKPUI into a LNKLSxxx data set for access from TSO ISPF sessions. This can be done by simply rerunning the INSTALL job.

7.15.1.3 CHECKPOINT ISPF INTERFACE - INVOCATION

To invoke the ISPF interface, performing the following tasks:

1. Modify the SMFCLBDF CLIST member of the INSTLIB data set to conform to the local data set naming standards used for SMFUTIL ISPF panels and messages (the names given during the unloading of the product tape) and place the CLIST into a data set available on the SYSPROC CLIST library concatenation for your logon procedure.
2. Modify the your local ISPF primary menu (or another appropriate local ISPF menu), as illustrated in member SMFUPRIM of the ISPLIB data set, to invoke the SMFCLBDF CLIST.
3. Ensure the SMFCKPUI load module is available in a STEPLIB or LNKLISTxx data set.

7.15.1.4 CHECKPOINT ISPF INTERFACE - OPERATION

Launch the SMFUTIL CHECKPOINT ISPF interface by selecting the appropriate item on the SMFUTIL primary menu modified during the installation. Once execution has begun the following screen will be displayed:

```
SMFUC00P ----- SMFUTIL CHECKPOINT DATASET DISPLAY UTILILTY -----
COMMAND ==>

Enter or change the SMFUTIL Checkpoint Dataset name to be processed.

Checkpoint Dataset =>
```

The first time this screen is presented you must enter the name of the SMFUTIL Checkpoint PDS you wish to view. Type the name of the Checkpoint data set and press the “ENTER” key and the following panel will be displayed:

```
SMFUC01P ----- SMFUTIL ARCHIVE CHECKPOINT DATASET DISPLAY ----- Row 1 of 6
COMMAND ==>                                     SCROLL ==> CSR

Checkpoint Dataset => SMFUTIL.TEST.CKPT

Enter S by a line to select the record for full detail.
Enter D by a line to logically delete the record.

Rec Name          Description
-----
$GDGNRML GDG INDEX TO BE USED FOR INPUT
$PROCESS PROCESS IDENTIFICATION STAMP
INPTGDG1 FIRST INPUT RECORD IDENTIFIER FOR GDGCKPTI
INPTREC1 FIRST INPUT RECORD IDENTIFIER
LASTGDG1 LAST RECORD PROCESSED IDENTIFIER
LASTREC1 LAST RECORD PROCESSED IDENTIFIER
***** Bottom of data *****
```

This panel displays a scrollable list of data member records found in the specified checkpoint PDS. Each line represents a single record.

To delete a specific record, place a “D” on the line desired and press the “ENTER” key. A confirmation screen will appear. Press “ENTER” to confirm deletion of the record.

To view the full detail information about a specific record, place an “S” on the line desired and press the “ENTER” key. The following panel will be displayed:

```

SMFUC11P ----- SMFUTIL CHECKPOINT INPTREC1 DATASET RECORD DISPLAY -----
COMMAND ==>

Checkpoint Dataset => SMFUTIL.TEST.CKPT
Record Name       => INPTREC1

Purpose: This record is used to retain the first 18 bytes of data from the
         first record input for an execution. It is used to detect:
         1. If duplicate data is being processed
         2. If data regression has occurred (Date Time stamp backed up)
         3. If data has been skipped (gap between LASTREC1 and INPTREC1)

***** Record Data *****

Record Length      Hex=> 025C      Decimal=> 604
Record Flags       Hex=> 5E
Record Type        Hex=> 63      Decimal=> 99
Record Time        => 13:08:06.60
Record Date        Julian=> 202.353 Gregorian=> 12/19/2002
Record Sysid       => P390

*****

```

Please note that no changes may be made to the record content.

Each discrete record type has a unique display screen that describes the purpose of that record and shows it's entire contents in detail.

7.15.2 CHECKPOINT RESTART FOR MOD DATA SETS

SMFUTIL has a built in checkpoint restart capability for MOD output data sets. A normal exposure of MOD'ing onto existing data sets is that if a program or system failure occurs during processing, duplicate data will be created when the job is rerun plus a broken (segmented) record may be present in the final block left on the tape after the failure. If either the SYSCKPT DD statement or the CKPTDB keyword are included in the execution JCL, SMFUTIL will be able to detect a restart of a MOD process and prevent duplicate data from being written. SMFUTIL keeps track of the first record written to an output MOD data set in the checkpoint data set. Upon an execution for restart, a checkpoint record will be discovered and SMFUTIL will search the last volume of the data set for the point that was originally used as the start of output. The output to the data set will begin at that point. This has the added benefit of avoiding the problem of tapes that have no EOF mark after a system failure. The tape will stop at the original data output start point and never process to an EOF mark. Although not as critical, the disposition of MOD output tapes on DD's SYSUT2 and SYSUT2D should be:

```
DISP=(MOD,CATLG,KEEP)
```

This will not prevent additional volumes from being cataloged if the run should ABEND for some reason (like an operator cancel). SMFUTIL does not use the catalog to locate the last volume of the data set during a restart but instead keeps the volume sequence number of the first output volume in the checkpoint record. If additional volumes have become cataloged in a failed execution, the restart point will still be on the correct volume and the system will attempt to mount the subsequent volumes in the catalog set as output volumes. This may or may not cause a problem in your environment, depending on what tape management system is in use and how it is set up.

Disposition for the SYSUT3 and SYSUT3D should be:

```
DISP=(NEW,CATLG,DELETE)
```

This prevents a split data set from being kept in the event of an ABEND. SMFUTIL will properly recreate the split data set during the restart. JCL specified disk data sets should normally be avoided for checkpoint recoverable copy functions. The format 1 DSCB for the new data sets will be created even though the data set may not be opened and used. These DSCB's would remain on the VTOC of the disk volume. This could result in JCL errors during a subsequent rerun of the job (i.e. "DUPLICATE DSNAME ON VOLUME"). It is possible to insert an additional step to delete the split disk data sets not created. This step would execute when a condition code of 0 came out of the copy step indicating no split had occurred. This will work fine as long as a system failure does not occur during execution. If one does, the unused split data sets will be left on the DASD volume(s) and it/they will not be cataloged. A manual restart procedure would be required to delete the data set before reruns. Conversely, if properly set up, tapes are not created until they are opened. The unit specification UNIT=(TAPE,,DEFER) should be used to prevent tape mounts until the tape is actually required for output. The only drawback to using all tapes is that as many as four tape drives could be required if both a daily and month to date sets were being created with split specified for each. This could be set up to run in the early AM after all nightly production is finished and demand for tape drives is at a low.

An alternative to using JCL specified data sets is to use the DDA block structure to dynamically allocate the data sets. This has many benefits. The primary benefits for MOD data sets with split processing is that data sets do not have to be allocated until they are needed for output. This allows DASD data sets to be used for MOD processing because restart processing can automatically reset and reuse DDA created data sets.

The actual data set name of the data set involved is not maintained in the checkpoint record, thus SMFUTIL has no way of knowing if the JCL was changed between executions. If the restart point is not located, SMFUTIL will assume that the data set is valid as is and will continue with MOD to the end of the data set. This makes it very important to use a unique SYSCKPT data set for each production setup of SMFUTIL. At least one for each operating system environment (unique SMF ID) is required (for dump and clear processing). If a post dump process moves the SMF data via a MOD disposition, it should also have a unique SYSCKPT data set.

The SYSCKPT data set is also used to maintain an indicator that a CLEAR function is in progress. If a copy step starts and finds a clear in progress indicator set in the SYSCKPT data set, the copy will terminate with an error message. If the clear function has been executed successfully, the directory entry for SYSUT1 may be deleted from the SYSCKPT data set and the copy step restarted, otherwise rerun the clear step after making sure that SMF has not placed additional data in the data set since the aborted CLEAR run occurred. Production copy and clear procedures should provide a SYSCKPT data set, even if no MOD processing is to occur, in order to protect from attempting to read data from a data set that was supposed to be cleared.

The checkpoint data set is a PDS. Only the directory is used, so it never requires compressing. The sample job ALOCCKPT in the INSTLIB can be used to allocate the data set. For a two step copy and clear jobs, SYSCKPT should be included in both steps, even if MOD is not being used. This will prevent a partially cleared data set from being dumped again in error.

7.15.2.1 TLMS AND MOD RESTART

Shops that utilize the Computer Associates product TLMS must take special precautions for the restart of MOD tapes to be operational. Failure to ensure a proper setup may result in an S713-04 ABEND (MSG IEC148I with return code 04) during restart processing. There are two items that must be addressed. The TLMS options member TLMSIPO in 'CALPPOPTION' must have the DBLTIME parameter set to non-zero (DBLTIME=00001 is sufficient) and the TMBYPASS parameter in the SMFUTIL defaults module must be set to 'YES' (as shipped). This enables SMFUTIL to specify a special parameter of "SPACE=(1,(1,3))" during DDA block allocation for mod tape restart. Please see the section titled "BYPASSING DOUBLE OPEN PROTECTION" in the "TLMS MVS SYSTEM PROGRAMMERS GUIDE" for more information on how to implement this change. If required, please contact Computer Associates TLMS support for assistance.

Note that TMBYPASS=YES is only effective for dynamically allocated tapes. MOD tapes allocated via JCL must have the "SPACE=(1,(1,3))" coded in the JCL for the restart open to work correctly. This may be coded in the JCL and left permanently with no harm.

7.15.2.2 CA1/TMS AND MOD RESTART

Shops that utilize the Computer Associates product CA1/TMS must take special precautions for the restart of MOD tapes to be operational. Failure to ensure a proper setup will result in an ABEND S713 during restart processing. In older versions (below 5), the TMS macro TMMDBLOP must be modified. After the labels ".PTMC ANOP" and ".PDSNB ANOP" change the

```
"BZ      &LABEL"
```

to

```
"B      &LABEL"
```

at sequence numbers 00210000 and 00350000. Please refer to Computer Associates technical support for assistance in making this change. Note that the stated CA policy is to move to object only distribution so there is no guarantee that this modification will be supportable in the future. You should contact Computer Associates about support for this capability via an installation option prior to that change.

The newer release of TMS(5.0 and up) has an exit that will support the double open function. The exit point, named TMSUX1F, may have been implemented by the site and must "ALLOW" SMFUTIL to double open all output SMF tapes for correct execution of MOD restart.

TMS installations should set the TMBYPASS defaults parameter to "NO" as it is not required.

The quickest way to determine if any changes in the tape management system is to test the MOD restart capability. Use SMFUTIL to set up a test tape dataset with several thousand records from yesterday on it. Then run an SMFUTIL run (with a CKPTDB dataset present) to MOD onto the tape previously defined. Cancel the run while data is being copied (active I/O is happening, not just searching for EOD for the MOD to start at). This should leave the CKPT PDS with the restart checkpoint records necessary for the restart. Rerun the MOD job and let it run to completion. If it successfully processes the restart (removes the previous MOD data) reopens the tape for MOD output and finishes the run with a good return code, no changes are required to the tape management system. If not, contact SMFUTIL TECHNICAL SUPPORT for assistance.

7.15.3 CHECKPOINT RESTART FOR NEW DATA SETS

SMFUTIL has a built in checkpoint restart capability for NEW output data sets. A normal exposure of creating new data sets is that if a program or system failure occurs during processing, duplicate data sets will be created and not properly cataloged. If either the SYSCKPT DD statement or the CKPTDB keyword are included in the execution JCL, SMFUTIL will be able to track the creation of new data sets. In the event of a restart after a failure, these data sets will be scratched and/or un-cataloged prior to being created again.

7.15.4 CHECKPOINT PROCESSING FOR DUMPCLEARALL

When the DUMPCLEARALL function is requested, SMFUTIL uses a two part checkpoint function. First a record called “\$DUMPALL” is inserted into the SYSCKPT PDS data set. Second, a special sequential checkpoint data set is allocated to DDNAME SYSDCKPT. The name of this data set is controlled by the DACKPT parameter in the SMFUDFLT defaults module or by the DCACHECKPOINT keyword specified at execution time. Upon successful completion of the DUMPCLEARALL function the SYSDCKPT sequential data set is scratched and the “\$DUMPALL” record is removed from the SYSCKPT PDS. If the DUMPCLEARALL process is interrupted for any reason, the SYSDCKPT sequential data set is used to reconstruct the environment that was in place and to facilitate a restart of the process.

See the section titled “DUMPCLEARALL PROCESSING” on page 217 for more information on how to implement DUMPCLEARALL.

7.15.5 CHECKPOINTING INPUT DATA

If the CKPTINPT feature is enabled (by default as shipped or explicitly requested via the keyword) SMFUTIL will use the SYSCKPT DD statement data set (if present in the JCL or via CKPTDB keyword) to maintain a checkpoint record of the first input data record found. When SMFUTIL first begins execution, the first data record read in will be compared to the previous input record checkpoint, if it exists. If they match, SMFUTIL will report that the input data has already been processed and terminate the run. If the date and time of the checkpointed data record is younger than the date and time of the first input data record by an amount greater than that specified by SORTTOLERANCE, SMFUTIL will report that regressed input data has been detected and terminate the run.

Please note that the CKPTINPT feature should not be used (NOCKPTIN should be specified) for SMF data set dump and clear jobs where it is possible for the SMF data sets to be dumped out of the order they were filled. If this situation occurs with CKPTINPT enabled, SMFUTIL will report regressed input and terminate the run when an SMF file is input that has data older than the last SMF file dumped. If proper dumping procedures are followed (i.e. the IEFU29 exit causes each data set to be dumped when it is filled or otherwise switched off of) this situation should not occur. The “ORDERCHK” parameter can be used to prevent this problem by ensuring that the SMF data sets are dumped in sequence.

Note that during a restart of an aborted run containing MOD data sets, SMFUTIL will temporarily bypass the CKPTINPT process so the same input data can be processed again as it was not successfully processed in the prior run.

The checkpoint data set is a PDS. Only the directory is used, so it never requires compressing. The sample job ALOCKPT in the INSTLIB can be used to allocate the data set. For a two step copy and clear jobs, SYSCKPT should be included in both steps, even if MOD is not being used. This will prevent a partially cleared data set from being dumped again in error.

It should be understood that this feature is intended for use only in standard production jobs that move large amounts of SMF data on a regular basis (i.e. DUMP and CLEAR jobs or month to date data collection or consolidation jobs). A typical ad-hoc user job to retrieve a required portion of SMF data from an existing archive data set would not normally utilize a SYSCKPT DD statement, therefore this function would not be active. The same input data would then be able to be reused multiple times until the desired output data is obtained. If a SYSCKPT DD statement is used in such an ad-hoc job to protect MOD output data sets against a system failure, the NOCKPTIN keyword should be specified to disable this function.

7.15.6 CHECKPOINTING DUMPING SEQUENCE

If the SYSCKPT DD statement data set is present in the JCL, SMFUTIL will use it to maintain a checkpoint record called “LASTREC1”. This record contains the date and time of the last record processed in an execution. If the “ORDERCHK” feature is enabled SMFUTIL will use this information to validate that input data is being processed in the correct sequence. When SMFUTIL first begins execution, the first data record read in will be compared to the previous “LASTREC1” checkpoint record, if it exists. If the input data record date/time differs from that of the “LASTREC1” date/time by an amount greater than the value specified (or defaulted) by the “ORDERGAP” keyword, SMFUTIL will report that the input data is being processed out of sequence and terminate the run with a return code of 8.

7.15.7 BYPASSING PREVIOUSLY PROCESSED DATA

One additional function of the “LASTREC1” checkpoint record is to enable SMFUTIL to bypass previously processed data and begin input with the first record found after the last record processed in the previous execution. The “SKIPOLD” function invokes this process. If “SKIPOLD” is specified, the date/time value of each record input will be compared to the “LASTREC1” date/time values. If it is less than or equal to the “LASTREC1” values, the input record will be totally ignored and processing will continue with the next record. Note that any records ignored in this way will not affect the run in any way. It will be as if they did not exist in the input stream.

Note that if “SKIPOLD” and “CHCKINPT” are specified in the same execution, the processing for “CKPTINPT” is somewhat altered. If the exact same first input record is found, processing continues and “SKIPOLD” is in effect so that old data is bypassed. If input data older than the “INPTREC1” record is found (data logically BEFORE the first record of the previous run) SMFUTIL will report regressed input data and terminate with a return code of 8.

The “SKIPOLD” function can be very useful when extremely current SMF data is required for critical processing but must not be processed twice. A typical use of the SKIPOLD function would be the loading of SMF data to a performance database on an automated basis. Each day, an SMFUTIL execution could be invoked against the entire previous days SMF data with and data gathered so far today concatenated to it. This run would strip out the record types required to be added to the performance database. The specification of “SKIPOLD” would ensure that any data selected from the previous day in yesterdays run would be bypassed. The SYSCKPT data set used for this run should be unique and not used by any other execution of SMFUTIL. Note also that the concatenation of current data could logically include any recent data still in SYS1.SMF files via the use of the SYSUT1xx DDNAME structure but caution should be used to ensure no more than one MAN data set contains data. If more than one has data, they may be processed out of order. This would typically be the case when SMF was recording to the primary and an alternate data set contained data that had not been dumped yet (or is in the processing of being dumped). Proper timing of the job in question should alleviate this problem by ensuring that only one MAN data set has data because not enough time has elapsed since the last dump and clear for the data set to fill. Use of the supplied IEFU29 SMF exit to dump and clear MAN data set as they fill is highly recommended.

7.15.8 HIGH SPEED CARTRIDGE POSITIONING FOR MOD PROCESSING

If the SYSCKPT data set is present and an output data set on a cartridge type device is opened for MOD processing, SMFUTIL will inspect the SYSCKPT data set for a MOD positioning checkpoint record for the current DD name. If found, the block ID that it contains will be used to request the device perform high speed positioning of the data set for MOD output. If the checkpoint record is not found or does not match the data set in use, normal MOD open will be performed.

When a MOD data set is closed and the SYSCKPT data set is present, the final block ID in use by the data set is recorded and written to the checkpoint record for the current DDNAME thus making it available to the next execution of this type.

7.16 RECOVERING FROM B37/D37 ABENDS ON OUTPUT DATASETS

If a dynamically allocated output disk dataset fills to its maximum number of extents or no more space is available on the volume it resides on, a B37 or D37 ABEND will occur. NEW datasets can simply be redefined with more space or multiple volumes but existing datasets require more attention.

Recovery for OLD or MOD datasets can be accomplished by specifying a volume serial list in the DDA block (via the VOLUME keyword) for the dataset to extend the dataset onto additional volumes. Several cautions need to be observed in this process. First, the volume serial list must contain all existing volumes, in the correct order, that the dataset already resides on. Additional volumes may be added to the list as required. **It should also be noted that the System may or may not correctly re-catalog the dataset to include the new volumes. If not, it is the users responsibility to manually re-catalog the dataset to correct the volume list in the catalog entry. Before re-cataloging the dataset on the new volumes, ensure the dataset now physically resides (is in the VTOC) on each of the volumes.**

7.17 OUTPUT DATA SET I/O ERRORS

Please read this entire section before proceeding to attempt to recover from an I/O error on an output data set. You must follow the instructions exactly as specified in order to ensure a successful recovery without introducing logical errors in the output data set. To ensure recoverability, you must be using a SYSCKPT data set via JCL or a CKPTDB specification in the control card set.

As a general rule, MVS operators should be instructed to disallow a DDR (Dynamic Device Re-configuration) "SWAP" of an output SMF tape from one tape drive to another due to an output error. The SWAP process has been known to introduce errors in the structure of output data sets created with RECFM=VBS. Understand that this is a system function in conjunction with the hardware and totally out of the control of SMFUTIL. It is quite possible for a DDR swap to apparently correct an output error (i.e. return code of 0 from SMFUTIL) yet the resulting tape will be logically damaged. It is much preferable to allow the run to abort (reply 'NO' to the swap request and the I/O error will be intercepted by SMFUTIL).

RESTART SCENARIO 1 - ERROR CAUSED BY DEVICE

If the output data sets utilize data set name suffix generation utilizing a date from the output data and the tape or cartridge in question is not defective, you may restart the execution normally. The tape will be reused. If the output data set names are GDG and the aborted run cataloged new data sets, you must follow the procedure in SCENARIO #2 below.

RESTART SCENARIO 2 - ERROR CAUSED BY NEW TAPE

If an output tape is defective and was new to this execution (i.e. 'PRIVAT' mount), a restart is possible as long as the system catalog is returned to its prior condition. A restart of an aborted run will allow a new tape to be used as the restart will begin again on the previous MOD tape. For this to occur correctly, it is essential that any tapes added during the aborted run (including the defective tape) be dropped from the set in the system catalog if catalog processing added it (it normally will regardless of disposition). Use the IBM utility IDCAMS to do the following:

```
DELETE dsname NOSCRATCH
DEFINE NONVSAM(NAME(dsname) -
              DEVTYPE(device) VOL(vol-list))
```

where 'dsname' is the correct data set name, 'device' is the device type used, and 'vol-list' is the list of volume serials for the data set with all new volume(s) (i.e. tapes added as 'PRIVAT' mounts in the aborted run) omitted from the end. The last volume in the list should be the one mounted for 'MOD' processing to begin in the aborted run. The intent is to return the catalog entry to the condition it existed prior to the aborted run. After this is done, the job can be rerun and a normal restart will occur. Ensure that the defective tape causing the original error is not reused. It should be returned to the tape library as defective.

RESTART SCENARIO 3 - ERROR CAUSED BY EXISTING TAPE IN DATA SET

If the error occurs on a MOD tape already cataloged in the set (i.e. one that contains data from a previous run and has had new data added in the current aborted run), the entire data set must be copied to another set. Before this can be done, the data set needs to have the data that was added (MODed) in the aborted run removed. This can be accomplished by performing the following tasks:

1. Rerun the copy step only using the original SYSCKPT data set, JCL, and the original control card input with a specification of OUTPUTLIMIT(0) added to the set. This will cause the restart/reset of the output file(s) to their original condition to occur but no data will be transferred from the input data set to the output data set(s).
2. Delete any checkpoint records that remain in the checkpoint data set.
3. If the ARCHIVE data set is in use, use the SMFUAMNT ARCHIVE maintenance utility to delete the old data sets volume(s) so that duplicate data will not exist in the archive.
4. Copying the tape set to a fresh set of tapes with a copy only operation of SMFUTIL (no SYSCKPT data set). If the ARCHIVE was in use, you should archive the new set at this time (specify the ARCHIVDB keyword). Ensure that the correct archive set name is used by using the correct DDNAME for output. Use of the TYPETODD keyword will be necessary if the original output was not to SYSUT2 or SYSUT2D. Specify ARCHIVE for SYSUT2 and SYSUT2D or ARCHUSER if a user specified DDNAME is used along with the TYPETODD specification. Alternatively, you may use a DDA block to allocate the output data set(s) and specify the ASETNAME parameter in each DDA block to correctly specify the set name for the data sets archive record.
5. Rerun the job that aborted with the output directed at the new data set (DISP=MOD as before).

This process should result in an output tape set free of errors.

7.18 INPUT DATA SPLITTING

It is logical to have the archived SMF data grouped into specific time periods. SMFUTIL offers a method of splitting the output data into physical days, weeks, or months. If a split request is in effect, SMFUTIL will check the output data for a change on the appropriate boundary. If a change is encountered, the output input data, beginning with the record that triggered the change, will be placed on the SYSUT3 output data set. If an additional split boundary is encountered and if a SYSUT4 DD statement is present in the JCL, output to SYSUT3 will cease and SYSUT4 begins to receive data. This continues as long as split boundaries are encountered and additional DD statements are found (through SYSUTZ). Note that if the input data is badly out of sequence, an improper split could result. Type 2 and 3 records should NOT be allowed to be input to SMFUTIL during a split process unless the data has been pre-sorted. Only one type of split for the primary data sets (SYSUT2 and SYSUT3) can be in effect during a run. If more than one split type is specified, the last specification encountered will be used. A unique feature of SMFUTIL is that the duplex data sets can be split on a different boundary than the primary. This allows the creation of a daily tape on the primary outputs (SYSUT2 and SYSUT3) and a MONTH-TO-DATE tape on the duplexes (SYSUT2D and SYSUT3D). See the sample JCL in the installation library INSTLIB for more information on how to set up this process (members SMFUDUMP and SAMPDCA).

If DDA BLOCK dynamic allocation is not used, it is important to specify deferred mounting of the SYSUT3 DD statement (and SYSUT3D if used). This prevents them from being continuously mounted when they are used rarely. These data sets will not be opened unless data is to be written to them. For this reason, we can specify a “CATLG” disposition and the data sets will not be cataloged until they are actually created. Caution should be exercised when utilizing DASD as the target for output data sets using the split functions. The DSCB's will be created even if the data set is never opened. This can cause JCL errors in later runs. A conditionally executed step may be added to scratch the new data set when it is not used for split data (i.e. when the return code from the copy step is 0). Manual intervention to scratch the data set will be required when a system failure occurs during the run as the unused data set will remain on disk. The DDA BLOCK method of allocation is preferred because it forestalls these problems by not actually allocating the data sets until they are required to receive output data. It also allows disk type data sets to have suffixes generated for them.

If the boundary that is required to split on happens to be between two different data sets in two separate executions, SMFUTIL would normally not be able to detect the split boundary. An example of this would be attempting to collect a weeks worth of data by MOD'ing each days data onto a weekly tape. If the days are collected each to a separate tape (i.e. a day split is done at dump time) each tape would contain 1 and only 1 day with no overlap. As each complete day is added to the week tape in a separate execution (daily), SMFUTIL will never see the data cross the week boundary (i.e. over midnight) during execution. A solution to this is to use the SYSCKPT data set in all split executions. When SYSCKPT is present, SMFUTIL keeps a checkpoint record (called LASTREC1) of the data and time of the last record processed. When a split run is begun, this checkpoint record is compared to the date and time of the first input record to see if a split boundary occurred between the two runs.

Optionally, the user may specify the CKPTLAST keyword for dynamically allocated output datasets. The value specified will be used as a unique checkpoint record used to retain the final record date/time value that was placed to the dataset. Upon startup the checkpoint record named by the CKPTLAST parameter will be used to detect any split boundary that occurred between the two runs, in lieu of the value supplied via the LASTREC1 checkpoint record. This capability is useful for output dataset where the record population is sparse and may not occur frequently enough to allow the LASTREC1 checkpoint value to reliably detect the split boundaries for the dataset between runs. The CKPTLAST specification record only retains date/time values unique to the record type(s) involved in the output dataset and results in precise splits regardless of the length of time between record occurrences.

The MSPLIT, MSPLITDP, and DDAEND month split requests will normally cause data to be split on a calendar month boundary (i.e. first day of a month causes a split to be taken). The installation may wish to define monthly periods other than the normal Gregorian calendar. This may be accomplished by defining a MONTHCAL or MONTHPCAL table. This table contains dates that define the start of a new month period and a suffix to be used if suffix generation is requested. The members MONTHCAL and MONTHPCAL in the INSTLIB data set contain examples of these tables. The ‘MONTHCAL(cal-name)’ is used to place an installation defined month calendar into effect and MONTHPCAL(cal-name) places an installation defined perpetual calendar into effect. See the section titled “DEFINING INSTALLATION CALENDARS” on page 211 for more information on how to install either of these tables.

The xSPLIT and xSPLITDP keywords are supported for compatibility with SMFUTIL releases prior to 5.0. The DDA block method of data splitting is the recommended method for setting up a split request. It allows user output data sets to be split as well as SYSUT2 and SYSUT3.

7.19 AUTOMATIC DATA SET NAME SUFFIX GENERATION AT ALLOCATION

SMFUTIL has the ability to automatically generate a data set name suffix for data sets it dynamically allocates via a SUFFIX keyword within the DDASTART/DDAEND set.

The first character of the SUFFIX keyword parameter in the DDA block indicates the source of the date/time stamp to be used to generate the suffix.

C	-	Current machine date
P	-	A previous machine date/time calculated from current machine date/time adjusted backwards, by the suffix type specification.
D	-	Record date/time value

Input data sets may only use the “C” and “P” source type suffix specification for data set allocations. The “D” and “C” source type suffix may only be used on output data sets for creation of the suffix. The “P” type may not be used for output as it is considered invalid.

The second character defines the format of the suffix to be generated as follows:

T	-	Time	.Jccyyddd.Thhmmss
X	-	Extended Time	.Jccyyddd.Thhmmss.Hth
D	-	Day	.Jccyyddd
W	-	Week	.Wwwccyy
M	-	Month	.mmmccyy
Y	-	Year	.Yccyy
F	-	Full	.Yccyy.Dvvdd
S	-	Week start	.Yccyy.Wvvdd
P	-	Month start	.Yccyy.Mvv

In the above definitions, the capital letters shown are as they appear in the final suffix. Lower case letters are place holders as follows:

cc	-	Integer century value (i.e. 19 or 20)
yy	-	Integer year of century value (i.e. 99)
ddd	-	Integer Julian day of year (i.e. 001 for Jan 1)
hh	-	Integer hour of day (range 00 to 23)
mm	-	Integer minute of hour (range 00 to 59)
ss	-	Integer second of hour (range 00 to 59)
th	-	Integer tenths and hundredths of a second (range 00 to 99)
mmm	-	Month abbreviation (i.e. JAN)
vv	-	Integer month value (range 01 to 12)
dd	-	Integer day value (range 01 to 31)

Valid suffix types (as defined for use in the SUFFIX command) are:

DT	-	Data date date/time
DX	-	Data date date/time extended format
DD	-	Data date day
DW	-	Data date week
DM	-	Data date month

DY	-	Data date year
DF	-	Data date full date stamp
DS	-	Data date week start
DP	-	Data date month start
CD	-	Current day
CW	-	Current week
CM	-	Current month
CY	-	Current year
CS	-	Current week start
CP	-	Current month start
CF	-	Current date full date stamp
CT	-	Current time
CX	-	Current time extended format
PD	-	Previous day
PW	-	Previous week
PM	-	Previous month
PY	-	Previous year
PS	-	Previous week start
PP	-	Previous month start

Special consideration should be given if weekly suffixing is to be used (SUFFIX(DW)). Please note that the weekly suffix generation will create a suffix that numbers the weeks of the year from '00' to '52'. Week '00' will be created when the first week of the year contains less than 7 days. For example, if a week start value of 'SAT' is used and January 1 falls on Wednesday, The '00' week data set would contain Wednesday (January 1) through Friday (January 3). The '01' weekly data set would contain the first full week of the year, January 4 (Sunday) through January 10 (Saturday). The last week data set of the year will contain data for the final week start day of the year through the last day of the year.

A week start date is defined as the date of the day defined as the beginning of the week (as set by the WKSTART default or keyword) that contains the target date. For example, if WKSTART were defined as SAT (Saturday) and a SUFFIX(DS) were requested using a date stamp from a record for December 24, 2002 (Tuesday), the returned suffix would be ".Y2002.W1218". This is because 12/21/2002 is a Saturday and the beginning day of the week containing the target date, 12/24/2002.

The WEEK START suffix is unique in that it can cross a year boundary. If the target date were Wednesday, January 1, 2003 and the week start still Saturday (SAT), the generated suffix would be ".Y2002.W1225" (the last Saturday in 2002).

Note that the old GENDSNx keyword specifications are valid only for DDNAMEs SYSUT2 and SYSUT2D and may only be used for a new, non-GDG data set on a tape drive. **These keywords are supported only for compatibility with prior versions of SMFUTIL and should not be used in new implementations. They will be removed from a future version of SMFUTIL. Dynamic allocation should be used with a SUFFIX(xx) specification instead.** Only one of these keywords may be specified. It will apply to BOTH SYSUT2 and SYSUT2D. If SYSUT2 or SYSUT2D have a GDG data set name format, a warning message will be issued, generation of the suffix for that DDNAME is not performed and the run will continue normally. The return code is unaffected. This allows either of the DDNAMEs to be used as a GDG and the other as a suffixed name. If either of the data sets is on disk, an error message is issued and the run continues with a minimum return code of 4 set. If either data set name is unable to receive the suffix because the base name is too long to allow it (i.e. the suffixed name would be longer than 44 characters), the run is aborted with an error message.

Suffix generation for dynamically allocated data sets may be specified for any device type and DDNAME. If the data date content is to be used to generate the suffix (i.e. SUFFIX(Dx)), allocation must be deferred until the first record to be placed to the data set (i.e. the data date) is available. The data set name specified in the DDA block must be short enough for the suffix generated to be appended and the resulting data set name still be 44 characters or less.

A DDA SUFFIX specification is mutually exclusive with a data set name containing a relative GDG specification, a specific GDG syntax suffix or a temporary data set name structure.

The intent of the SMFUTIL suffixing capability is to allow SMF data to be copied to permanent archive data sets that are more easily recalled should the need arise. Once archiving of SMF data to a GDG is completed, it becomes very difficult to recall the data without careful consideration and counting of generation numbers. The SUFFIX(DM) keyword will probably be the most useful in this situation. It could be automatically used to copy a just completed monthly tape (from a successful monthly split operation) to a separate data set name containing the month and year as a suffix (i.e. SMF.MNTHMSTR.APR2002).

For the suffixing operation to function properly, it is essential that type 2 and type 3 records be EXCLUDE'd from processing if IFASMFDP was used to dump the data from the SMF data sets. They may cause an invalid data set name suffix to be generated. In addition, data to placed on a data set with a data set name suffix generated for it should be split on a proper boundary. This can be accomplished via the xSPLIT or xSPLITDP functions for SYSUT2 and SYSUT2D respectively, by the SPLIT operand of the DDAEND keyword for dynamically allocated data sets, or by selecting a date or date range via the DATE function.

The SUFFIX(DT) specification of the DDA block structure is unique in that it generates a two level qualifier suffix on the end of the data set name. This suffix contains both the date and time from the first output record processed. Note that a time suffix is logically incompatible with a disposition of MOD because a new data set would be created each time due to the time changes.

Split data sets must have unique suffixes generated for each data set created. If the SPLIT sub-parameter of the DDAEND keyword is used, the split boundary specified must be logically compatible with the suffix specification. For example, an output data set to be split on a daily boundary may not have a weekly suffix as this would yield identical suffixes for multiple data sets. Conversely, it is valid for a week split data set to have a day type suffix.

7.20 CATALOGED DATA SET INDEX LEVEL INPUT

The CATINPUT parameter has the ability to automatically prepare for input all data sets under a catalog index level. Optionally, the last data set in the list may be omitted from use and the data sets may also be automatically deleted at the end of the run if the return code is less than 8.

The CATINPUT parameter allows a list of data sets to be input without knowing every name in the list. Unlike GDGCKPTI which uses the checkpoint record created by CKPTGDG, CATINPUT selected data sets will not necessarily be input in the order in which they were created. They will be in name sorted sequence. The way to overcome this is to use the SUFFIX(DT) parameter to create the data sets to be later input via CATINPUT. The created suffix will force the data set names to sort in the order in which they contain data (i.e. oldest will be first on the list).

Time suffixed data sets can be useful in some installations for dumping and clearing, especially when combined with the CATINPUT keyword in a later run. A unique data set is created each time a dump is performed. This prevents contention for the use of the output data set during the dump process. For example, suppose the dump process specifies the output data set as follows:

```
DDASTART DDNAME(SYSUT2)
          DSNAME(SYSX.SMFDUMP.DATA)
          SUFFIX(DT)
          DISP(NEW,CATLG,DELETE)
          UNIT(SYSDA) SPACE(CYL,25,25,R)
DDAEND(DEFER,SPLIT(D))
```

When midnight is encountered, a new data set is created using the date and time stamp from the first record for the new day. This will also cause a return code of 01 indicating a day split has occurred. A subsequent step that was only run when the return code indicates the split happened (i.e. RC=01) could use the following to input all data sets for the previous day:

```
CATINPUT(SYSX.SMFDUMP.DATA(SKIPLAST,DELETE))
```

This step could be used to move the accumulated dump data sets to a complete daily tape. The specification of SKIPLAST causes the last data set in the catalog to be omitted from the run. This was the one created by the split and contains data after midnight. The specification of 'DELETE' causes all of the input data sets to be deleted at the end of the step if the run is successful (i.e. return code less than 8). This step would only run once per day during the first dump job that runs after midnight. It would collect together all of the data sets created by dumps during the previous day. This will allow all of one days dumps to be batched together and placed to a daily tape all at once. This has the advantage of never using a MOD disposition. The full daily tape is created all at once.

If the accumulated data were required during the day for analysis purposes, the following could be used:

```
CATINPUT(SYSX.SMFDUMP.DATA) MANXALOC
```

This would access all data already dumped and all data still in the SMF MAN files.

CAUTION: If a user exit issues a return code terminating the run the resulting return code for SMFUTIL will be 4. If the 'DELETE' keyword is specified, this will result in deletion of the input datasets even if all of them have not been read. Use caution in implementing user exits when the 'DELETE' keyword is to be utilized.

7.21 GDG DATA SET HANDLING

Versions of SMFUTIL prior to 6.0 required an external utility called GDGDDGEN to generate JCL for SMFUTIL execution to read in multiple generations of a GDG index in the correct sequence. This capability has now been incorporated directly into SMFUTIL via the GDGINPUT, GDGCKPTI and CKPTGDG keywords.

7.21.1 CONTROLLING RELATIVE GDG DSNAMES INTERNALLY

SMFUTIL has the ability to control the suffix used for relative generations of GDG type data set names. If the “GDGCTRL” parameter of the SMFUFLT table (see the section titled “MODIFICATION AND CUSTOMIZATION” on page 199 for more information) is set to “YES” or if the “GDGCONTROL keyword is specified at execution time, SMFUTIL will assume control of the relative GDG level suffixes to be attached to input and output GDG data set names. This prevents allocation from “ENQ’ing” on the base level of the GDG name and preventing other jobs from accessing other levels of the GDG while this SMFUTIL execution has a particular level allocated.

If this option is used, caution should be exercised to ensure that more than one SMFUTIL execution does not attempt to allocate for output the same GDG data set name at the same time (simultaneous executions). This would cause unpredictable results. One possible scenario is that the second job may allocate the data set correctly but fail to catalog it as a duplicate data set name entry would already exist in the catalog from the first job. This is especially true if the output device is tape. DASD output data sets might get the same results (if to different volser’s) or the second job may get an allocation failure as a duplicate data set already exists on the DASD volume.

7.21.2 INPUTTING GDG GENERATIONS VIA SYSCKPT RECORDS

The GDGCKPTI and DDA block CKPTGDG keywords work as a team in separate executions. The CKPTGDG keyword requests that a checkpoint record be created/updated for a dynamically allocated GDG data set. A unique 1 to 4 character name is assigned by CKPTGDG. This user defined name is appended to “\$GDG” to form a SYSCKPT record member name that is stored in the checkpoint data set. This record contains the GDG index level, the current absolute generation number for the GDG index and the next absolute generation number start point for input processing (oldest generation). A separate execution of SMFUTIL will specify the GDGCKPTI parameter with the same 1 to 4 character name. The checkpoint record will then be interrogated for the index level and the starting and ending generation numbers. Each existing generation from the start point (inclusive) to the end point (inclusive or not as defined by SKIPLAST on GDGCKPTI) will be input to the run much the same as defined for GDGINPUT.

The SYSCKPT data set for the execution containing the DDA CKPTGDG specification must also be specified in an execution that specifies the GDGCKPTI for the same \$GDG checkpoint record name. This same checkpoint data set should not be used in any way other than these two runs. They are, in essence, a linked execution pair that require the same SYSCKPT data set.

NOTE: If the CKPTLAST keyword is used in both creating and inputting job, unique values must be used in each job. A DDA block in the INPUT job must not use the same CKPTLAST value as the CREATING job or unpredictable splitting may occur.

The first time a CKPTGDG record is created, the starting absolute generation number will be set to the same as the first generation number created by default. For subsequent executions with the same CKPTGDG, the starting number will be unchanged until a GDGCKPTI execution is run using the same checkpoint record. After a successful GDGCKPTI execution, the checkpoint record starting (and ending) generation will be updated to reflect the next start point (and end point) for this GDG data set as input.

The CKPTGDG keyword has an optional second parameter to allow a different starting absolute generation number to be defined. This specification will replace the default for new records and will overlay the current number for existing records.

For example:

```
... DSNNAME(SYSX.SMF.DAILY(+1)) CKPTGDG(DALY,0001) ...
```

would cause the next specification of GDGCKPTI(DALY) (in a different execution) to use data set name SYSX.SMF.DAILY.G0001V00 as the first data set selected for input. All generations between the start point and end point defined in checkpoint record \$GDGDALY would be selected.

If one or more of the required generations is not found, a minimum return code of 4 will be issued.

Note that CKPTGDG and GDGCKPTI have the unique ability to span a “wrap around” GDG index. This is one where the index goes from “.G9999V00” to “G0001V00”. This allows perpetual usage of a GDG index without fear that data will be input out of order.

7.22 SELECTING INPUT DATA VIA THE CATALOG

Input data sets may be defined to SMFUTIL via the GDGINPUT or CATINPUT keywords. These function essentially the same.

7.22.1 INPUTTING ENTIRE GDG INDEXES

The GDGINPUT keyword identifies a GDG data set index that contains data sets to be input to SMFUTIL. Each generation in the GDG index is defined to SMFUTIL as if it were specified in a unique DDASTART/DDAEND definition block. The oldest generation (smallest number) is allocated first and then the next oldest and so on until all generations are read in. Optionally, the newest generation (current or (0)) may be omitted from the selection by specifying the SKIPLAST keyword in a GDGINPUT sub-parameter specification. GDGINPUT is functionally identical to CATINPUT.

NOTE: Perpetually using a GDG index for output and subsequent input without ever resetting the index to '0001' can cause the index to eventually "wrap" from '9999' to '0001'. If this happens, GDGINPUT will input data out of order because the lower number generations will be presented first by the catalog. There are three ways to avoid this situation:

1. **RECOMMENDED:** Avoid GDG's entirely by using dynamic allocation and specifying appropriate data set name suffixes based on date and time of the output data and then use the CATINPUT keyword instead of GDGINPUT. For example, a daily dataset with a suffix of '.J2003100' will always sort correctly with other datasets similarly named, no matter how many datasets exist. These dataset names also have the advantage of being more meaningful as to their dataset's content.
2. Devise your GDG structure and usage such that all generations occasionally get deleted. This ensures the GDG index will restart at '0001'.
3. Use the GDGCKPT and CKPTGDGI keywords to manage creation and retrieval of the GDG generations. This method is immune to the 'wrap' problem due to internal processing watching for it and resetting the order correctly.

7.22.2 INPUTTING ENTIRE CATALOG INDEXES

The CATINPUT keyword identifies a catalog index level that contains data sets to be input to SMFUTIL. This index level must have data sets cataloged under it that contain only one additional qualifier. Each data set in the index is defined to SMFUTIL as if it were specified in a unique DDASTART/DDAEND definition block. The data sets are defined to SMFUTIL in the order they are returned from the catalog search. This is sorted by dsname. Care should be taken when data sets are created to ensure they are named such that the correct order will be achieved on input. For example, a data set index using a Julian day suffix would be presented in ascending order by date. Optionally, the last data set (newest if named correctly) may be omitted from the selection by specifying the SKIPLAST keyword in a CATINPUT sub-parameter specification. CATINPUT is functionally identical to GDGINPUT.

7.22.3 AUTOMATED INPUT DATA SET DELETION

The CATINPUT, GDGINPUT and GDGCKPTI keywords allow a DELETE keyword specification as a second sub-parameter. This requests that all data sets input are to be deleted at the end of a successful run. The actual deletion of these data sets is done via LOCATE, SCRATCH, and CATALOG SVC's within SMFUTIL as the last process prior to termination of the program. All output data sets have been closed. All DDA output data sets have been dynamically freed.

Because the allocation that utilized the input data set has already been released, it is possible that a data set to be deleted via this method may be enqueued upon by another address space (such as maintenance utility doing extent consolidation) and thus the SCRATCH request may fail. SMFUTIL has the ability to retry such failed scratch request. If the SCRATCHRETRY keyword is specified or the SCRETRY and SCRWAIT keywords in the SMFUDFLT module indicate retry is to be attempted.

Because JCL output data sets do not go through catalog processing until step termination, it is strongly recommended that dynamic allocation be employed for output data sets for executions that utilize CATINPUT, GDGINPUT or GDGCKPTI processing. This will ensure that the output data sets are successfully cataloged (and thus retained) prior to the deletion of the input data sets that constructed them.

7.23 HIGH-SPEED CARTRIDGE POSITIONING

SMFUTIL has the ability to perform high-speed positioning of cartridge type devices for data retrieval. This high-speed positioning consists of two components: data retrieval support and MOD positioning support.

7.23.1 POSITIONING FOR DATA RETRIEVAL

Data retrieval support assists all cartridge type devices in high-speed positioning to the data block where input is to begin. This can be manually requested by the user or automatically invoked by SMFUTIL.

Manual positioning consists of specifying the input data set via a DDA block structure (dynamically allocating the data set) and specifying the BIDSTART keyword within the DDA block. The BIDSTART value is an 8 character hexadecimal value. Suggested BIDSTART values can be determined by examining the BID values for SMFUTIL ARCHIVE records.

While this method is feasible, its usefulness is somewhat limited due to the large amount of intimate knowledge required on the part of the user. They must know not only the date they wish to retrieve but the data set name it is stored on and a good idea of the amount of data blocks on the volume. The complications are compounded even more if we must deal with a multi-volume data set.

SMFUTIL can automate this process for the user. During the ARCHIVE process, SMFUTIL retains block ID values (BIDs) for data locations on the tape. For weekly or monthly datasets each BID value represents the beginning of a new day of data. For daily datasets, each BID value represents the beginning of a new hour of data. These BID values are relative to the start of the volume. Each ARCHIVE record has 31 slots to maintain these block id's. This is designed to allow a full month of data to be stored on a volume and still have each day individually addressable via a block id.

Normal data retrieval using the SMFUTIL ARCHIVE database will automatically perform high-speed cartridge positioning to the first data block required, if the BLOCK ID value required is present in the ARCHIVE.

Note that ARCHIVE records created before a version of SMFUTIL that supports retention of the Block ID values will not provide the required information. Normal positioning will be performed if data is retrieved using the old records. In order to gain the advantage of high-speed positioning support for prior historical data, the installation may perform an ARCHINPT function using the following example as a guide:

```

ARCHIVDB(SITE.ARCHIVE.DATABASE)
DDASTART DDNAME(SYSUT1A) DSNAMES(SMF.DATA SET.MONTH1) DDAEND
DDASTART DDNAME(SYSUT1B) DSNAMES(SMF.DATA SET.MONTH2) DDAEND
DDASTART DDNAME(SYSUT1C) DSNAMES(SMF.DATA SET.MONTH3) DDAEND
ARCHINPT
SIMULATE

```

This process will read the input data set(s) and replace the existing ARCHIVE records with new ones that contain the required Block ID information. As many input data sets as you wish may be specified, limited only by the length of time you wish to allow the job to run. Note that the ARCHIVE update is not actually done until the job completes normally so do not cancel the job prior to completion or it will have to be rerun from the beginning. Multiple jobs could be run over a period of days during the off-shift hours to re-archive as much historical data as you require. Alternatively ARCFORCE may be specified to force ARCHIVE updated to occur for an abended run.

For output dataset archiving, SMFUTIL automatically sets the BID slots to hourly or daily based on the type of split that is being requested. For datasets that do not have a split requested via the DDAEND keyword, SMFUTIL defaults to one BID slot per day of data. This can be changed to hourly by specification of the BIDBYHOUR keyword in the DDA bloc. This is also valid for input DDA blocks while archiving input data. Note that if SMFUTIL detects a change in the day in the output data stream (or input stream during ARCHINPT) a BIDBYHOUR request is automatically suppressed and the BID value slots revert to one per day values.

7.23.2 POSITIONING FOR DATA SET EXTENTION (MOD)

The process of opening an output data set for MOD processing can be very slow due to the time it takes for the access method to locate the end of the data set. Cartridge type devices (i.e. 3480, 3490, and 3590) support a high-speed locate function that has been implemented into SMFUTIL. If the SYSCKPT data set is present (either via SYSCKPT in the JCL or via the CKPTDB keyword) SMFUTIL will maintain a special checkpoint record that is used to save the Block ID of the trailer label for a cartridge data set processed as MOD. When SMFUTIL is run again and the same data set/DDNAME combination is opened for MOD, SMFUTIL uses the saved Block ID to request the access method to do a high-speed locate of the end of the data set. This locate is a device hardware function. It is a detached process and thus independent of control unit, channel or processor loading.

7.24 MIH TIMING REQUIREMENTS

SMFUTIL performs many tape operations that are of extended length. If the MIH timing value for TAPE is left at the default value, these processes may result in the Missing Interrupt Handler (MIH) of the Operating System incorrectly detecting a missing interrupt from the device in use. For example, during a data extract using the SMFUTIL ARCHIVE, a high speed positioning of a tape via a block ID search (BID) may take an extended period of time while the device positions the tape to the required block. The MIH default time-out for TAPE devices typically set to 3 minutes by default and this is too short for 3490E, 3590 or other high density devices. To determine the current MIH timing value perform the following command:

```
D IOS,MIH
```

The resulting output will look something like this:

```
IOS086I 11.06.43 MIH AND IOT TIMES 235
MOUNTMSG = NO,          HALT=00:05, MNTS=03:00, UREC=03:00,
DASD=00:15, TAPE=03:00, GRAF=03:00, CTC =03:00, COMM=03:00,
CHAR=03:00, IOTDASD=00:00,
US01=30:00 UDEV=0590.
```

Note the value for the "TAPE=" specification. This value may be altered by issuing the following command:

```
SETIOS MIH,TAPE=10:00
```

This resets the TAPE device MIH value to 10:00 minutes. This value should be sufficient for most cases but a larger value could be required in some installations.

To change this value permanently, place the following line in SYS1.PARMLIB member IECIOSxx and ensure the member is activated at IPL time:

```
MIH TAPE=10:00
```

8. ARCHIVING SMF DATA

During the process of copying SMF data to its output files, SMFUTIL can be instructed to 'remember' where the data is stored by specification of the "ARCHIVE" keyword parameter and the supplying of the SYSARCH DD statement in the JCL (or the specification of the ARCHIVDB keyword parameter in SYSIN). This archived data can subsequently be used to retrieve data by date/time stamps without the user being required to know where the data is stored.

Note that the SMFUTIL archive is merely a pointing device to show where SMF data resides. IT DOES NOT CONTAIN ANY SMF DATA ITSELF. It contains records that show where SMF data is stored by volume and data set name. These records show volume, data set name, starting date and time on volume and other pertinent information required to locate the data and dynamically allocate the data set for input processing.

NOTE: **The ARCHIVE must not be used for output datasets that have "stacked" systems on them. The ARCHIVE requires the output data to be in the order created by SMF. "STACKING" or MOD'ing on multiple systems to the same output data set violates this requirement as the date/time stamp will rise and then fall back when a new system is added. It is highly recommended that each system's data (unique SMF system ID) be kept on its own output datasets. Multiple systems may use the same ARCHIVE if they are on discrete datasets.**

It is important to think through the process that will lead to data being archived. It is essential not to clutter up the archive with many, overlapping copies of the same data. During the search of the archive for required SMF data for a retrieval execution, SMFUTIL attempts to prevent duplicated data by eliminating overlapping date/time ranges. It does this by truncating an input data set at the point where the next input data set begins. This can lead to excessive tape mounts as multiple data sets may be allocated to satisfy a request. In its preferred condition the archive database will contain a maximum of two records for data sets containing the same data. One will be for the "NORMAL" data (created on SYSUT2/3/4...) and one for the "DUPLEX" data (created on SYSUT2D/3D/4D...). During an SMF data retrieval execution, only one "set-name" ('NORMAL' or 'DUPLEX') will be used so duplicated input data is not possible. This is the way it is designed to function.

Although not normally recommended, if desired, the installation may utilize the "ARCHUSER" keyword to cause the user defined output DD statements to be archived also. If this is done, it is essential to set up the process such that the output data is archived only once.

If the same data is handled multiple times (i.e. dumped to individual data sets and then batched together to a daily or weekly tape) only the final pass of data that places it onto its permanent storage data set should have the "ARCHIVE" and "ARCHUSER" keywords specified. User data sets may be archived in runs also archiving regular (SYSUT2) and duplex (SYSUT2D) output data sets ('ARCHIVE' specified also) or they may be archived alone (only 'ARCHUSER' specified). In addition, archiving may be limited to specific DD names by specifying them along with the associated command. For example, ARCHIVE(SYSUT2) or ARCHUSER(RMFDATA). If one or more DD names are specified, only those (and any associated split datasets) will be archived. Other output datasets will not be archived.

8.1 DEFINING THE ARCHIVE DATABASE

The Archive database consists of a VSAM entry sequence cluster (ESDS) with three alternate indexes defined over it. The member ARCHDEFN in the INSTLIB data set can be used to allocate and initialize the required VSAM structure. Note that the base cluster itself may be any name desired but is limited to 31 characters in length in order to allow sufficient room for the required suffix additions. The alternate indexes and the path definitions must be named with the suffixes shown in ARCHDEFN. The following VSAM data set names are defined in ARCHDEFN as shipped:

SYSX.SMFUTIL.ARCHIVE	Base cluster
SYSX.SMFUTIL.ARCHIVE.VOLUME	VOLSER alternate index
SYSX.SMFUTIL.ARCHIVE.VOLUME.INDEX	
SYSX.SMFUTIL.ARCHIVE.VOLUME.DATA	
SYSX.SMFUTIL.ARCHIVE.DATE	Start date alternate index
SYSX.SMFUTIL.ARCHIVE.DATE.INDEX	
SYSX.SMFUTIL.ARCHIVE.DATE.DATA	
SYSX.SMFUTIL.ARCHIVE.DSNAME	Data set name alternate index
SYSX.SMFUTIL.ARCHIVE.DSNAME.INDEX	
SYSX.SMFUTIL.ARCHIVE.DSNAME.DATA	
SYSX.SMFUTIL.ARCHIVE.PATHV	Path definition for VOLSER
SYSX.SMFUTIL.ARCHIVE.PATHS	Path definition for start date
SYSX.SMFUTIL.ARCHIVE.PATHD	Path definition for data set name

Edit the ARCHDEFN member and make a global change of 'SYSX.SMFUTIL.ARCHIVE' to whatever cluster name you wish to use. The remainder of the names as show above must be left intact. The space defaults defined in the member should be sufficient for most shops. The required database does not have to be very large.

NOTE: Do not place the SMFUTIL Archive Database base cluster or alternate indexes on volumes that already have high activity. Avoid volumes with catalogs, SMF data sets, DB2 or IMS databases and the like.

8.2 SERIALIZATION OF UPDATES TO THE ARCHIVE DATABASE

During updates of the SMFUTIL Archive database, a reserve is held against the volume containing the base cluster. The QNAME used for this reserve is 'SMFUTIL '. This reserve is brief but can cause system degradation of the system if the database is on an extremely busy volume.

Installations that utilize GRS (Global Resource Serialization) or a like OEM product, can let GRS manage the SMFUTIL reserve and propagate it as a SYSTEMS level ENQUEUE across the entire ring. This precludes the SMFUTIL reserve from halting access to the entire volume on other systems for the duration of the reserve.

This version of SMFUTIL supports two RESERVE/DEQ protocols. The new protocol uses the base cluster name of the archive as the RNAME field thus allowing multiple databases to be updated at the same time. The old protocol used a constant of "ARCHIVE " as the RNAME field. The RSRVOLD parameter of the SMFUDFLT module controls which protocol is used. A value of 'YES' (the default) reverts to the old format that is compatible with prior releases. A value of 'NO' invokes the new protocol. See the section entitled "MODIFICATION AND CUSTOMIZATION" on page 199 for more information on modifying the SMFUDFLT module.

CAUTION: Do not set RSRVOLD=NO until all prior releases of SMFUTIL (version 6.0 and below) have been removed from the data center and only the new RESERVE protocol will be in use. Failure to follow this procedure may result in corrupted data in the ARCHIVE database.

8.3 PROTECTING UPDATES TO THE ARCHIVE DATABASE

At it's option, the installation may elect to protect updates applied to the ARCHIVE database via SMF records containing before and after images of the updates to be applied. In a subsequent execution of SMFUTIL doing archiving to the same database, if a before image SMF record is encountered and no corresponding after image is found, the update will be reapplied to the database. This protection is useful for insulating the SMFUTIL execution from functional problems within the ARCHIVE database. Normally, if problems exist in opening the database or applying updates to it, the SMFUTIL run will complete normally but issue a minimum return code of 4 plus any split code. If update protection is enabled, the return code 4 set is disabled under the assumption that the updates will be applied later. Messages are still issued to the system operator via WTO and to the SYSPRINT data set indicating the failure reason.

To enable update protection, three conditions must be met.

- 1) SMFUTIL must be running in an authorized environment.
- 2) The ARCHSMF# in the defaults module SMFUDFLT must contain a valid SMF record type in the range of 129 to 255 or the ARCHSMFRECNUM keyword must have been previously specified in this execution with a valid record type and this record type must be reserved by the installation for the exclusive use of SMFUTIL.
- 3) The ARCHSMF parameter in the SMFUDFLT module must be set to YES or the ARCHSMF keyword may be specified at execution time.

When these conditions are met ARCHIVE, update protection is enabled. During execution of SMFUTIL, when an output data set that is being archived is closed, a before image copy of it's archive record is placed to the SMF system. After the update has been successfully applied to the database an after image copy is written indicating the update was completed.

In an execution of SMFUTIL with archiving being done and archive update protection enabled, if a before image record of an SMFUTIL archive update is encountered with no after image copy found, the unfinished update will be applied to the database before any of the current runs updates are applied. This ensures that the sequence of updates is maintained.

Note that in the event that there was an ongoing problem with the database (i.e. no space to insert new records), before image records will tend to collect in the SMF system without the corresponding after images. Once the database problem is corrected, the next run of SMFUTIL will "flush" all the pending updates to the archive.

For ARCHIVE update protection to work effectively the archiving should only be done during the dump/clear process and the DUMPCLEARALL function is highly recommended. DUMPCLEARALL will ensure that all current data in the SMF MANx system is made available to the DUMP process. Thus, it is much more likely that paired before/after images of archive updates will be correctly detected. While unlikely, if the single data set dump and clear approach (or DUMPCLEARALL with switching disabled) is used, it is possible that a before image could be written to one MANx data set the after image written to a different one if the active MANx data set fills up and a switch occurs during the dump process. While this will not cause a problem with the database it does allow duplicate updates to be applied and results in additional overhead of archive processing. Unpaired after image records are simply ignored.

8.4 RECOVERING ARCHIVE DATA

If the archive update protection process described in “PROTECTING UPDATES TO THE ARCHIVE DATABASE” on page 178 has been enabled and SMF records representing updates to the SMFUTIL archive database are present in the SMF archives, SMFUTIL has the ability to recover updates to the database that have been lost due to problems with hardware (i.e. head crash on the ARCHIVE database base cluster volume). This is accomplished by executing a special purpose SMFUTIL function called ARCHIVERECOVER. The following JCL is an example of how to execute this function:

```
//RECOVER JOB (acct)...
//RECOVER      EXEC PGM=SMFUTIL,REGION=8192
//SYSPRINT     DD   SYSOUT=*
//SYSUT1       DD   DSN=smf.input.data.set,DISP=OLD
//SYSIN        DD   *
SIMULATE
ARCHIVDB(smfutil.archive.database)
ARCHIVERECOVER
```

SIMULATE must be specified and no output data set is allowed.

This process will recover all before image records contained in the input data set on SYSUT1 and apply the updates to the SMFUTIL archive. The ARCHSMF# record number must be correctly set in the SMFUFLT module or the ARCHSMFRECNUM keyword must be specified with the correct record type (before the ARCHIVERECOVER keyword) as this is the record type that will be expected to contain the before image archive updates.

Note that the ARCHIVDB keyword is only required if the data set name has not been supplied in the SMFUFLT module.

Encountered ARCHIVE update image records are retained in GETMAIN'ed buffers until all records have been read. If a long period of time is to be recovered and many updates are normally done per day, multiple executions may need to be scheduled to prevent the region size from being exhausted. Each before image SMF record encountered will require approximately 1024 bytes of region on top of the three megabytes required by SMFUTIL itself so structure your recovery accordingly. This is less of an issue with version 8 as these buffers reside above the line.

8.5 SUPPLYING THE ARCHIVE DATABASE NAME TO SMFUTIL

SMFUTIL only needs to be informed of the base cluster name. It will dynamically allocate the other data set names as shown in the previous section. There are three ways to inform SMFUTIL of the base cluster name. First, a SYSARCH DDNAME may be supplied in the execution JCL pointing to the base cluster with a disposition of SHR. Second the ARCHIVDB parameter may be supplied specifying the base cluster name. These two methods are mutually exclusive with each other. As shipped, SMFUTIL does not use the ARCHIVE database unless one of the first two methods are utilized. Alternatively, the local base cluster name may be permanently installed by defining it in the defaults table. See the section entitled “SETTING EXECUTION CONTROL DEFAULTS” on page 199 for more information on installing a modified defaults table. Once installed in the defaults table, all executions of SMFUTIL will have the archive database available. If more than one database is to be maintained, the SYSARCH DD statement or the ARCHIVDB keyword may be used to override the installed archive cluster name with a replacement name.

8.6 ARCHIVING OUTPUT DATA

The ARCHIVE keyword can be used to instruct SMFUTIL that it is to 'ARCHIVE' the location of the output data. Each volume of data produced with a unique data set name will be archived. Each archive record contains an archive set name. For the 'SYSUT2/SYSUT3...' DDNAME sequence, the set-name is "NORMAL". For the 'SYSUT2D/SYSUT3D...' the set-name is "DUPLEX. For the 'SYSUT2X/SYSUT3X...' the set-name is "EXCLUDES. The 'NORMAL' and 'DUPLEX' sets are essentially duplicates of each other, probably differing only in the split boundaries that have been applied to the data. The 'EXCLUDES' set is a catch all for SMF data excluded from the normal output (SYSUT2). It will only exist if the SYSUT2X DDNAME is present in the run (via JCL or DDA block definition). Any user defined output data set will have it's set name default to the first five characters of the DDNAME for the data set.

The user may override the default set names by defining an output set name via the ASETNAME keyword within the DDA block definition for a dynamically allocated data set. The default set name assigned to JCL defined data sets may not be altered

8.7 ARCHIVING USER DATA SETS

The ARCHIVEUSER keyword can be used to instruct SMFUTIL that it is to 'archive' the location data placed to user defined data sets. These are data sets written to as a result of TYPETODD, CICSSPLIT, DB2SPLIT, or SIDSPLIT keywords. The first 5 characters of the output file DDNAME will be used as the set name for user archived data. As an example, if "TYPETODD(70:79-RMFDA2)" were specified in the same execution with "ARCHUSER", output to the data set would be archived under the set-name of 'RMFDA'. This convention ensures that all data created under a user split structure is archived with a common name.

The user may override these default set names by defining an output dataset set name via the ASETNAME keyword within the DDA block definition for a dynamically allocated data set. The default set name assigned to JCL defined user data sets may not be altered

8.8 ARCHIVING INPUT DATA

The 'ARCHINPT' keyword can be used to instruct SMFUTIL that it is to 'archive' the location of the input data. This allows an installation to read in current SMF archive files and have SMFUTIL 'remember' where this data is stored. Only single copy of each date/time span should be passed utilizing "ARCHINPT". By default, data archived in this fashion will be placed under the "NORMAL" archive set-name. This may be overridden via the 'ISETNAME' parameter.

As an alternative, SMFUTIL can be used to 'clean up' existing archives by passing the data through SMFUTIL to new output data sets with the 'ARCHIVE' keyword specified for the run. Duplex data may also be produced at this time. This has the additional advantage of ensuring the integrity of existing SMF archive data sets as well as re-blocking data for maximum storage efficiency.

The ARCHINPT keyword is mutually exclusive with VSAM input data sets. All data archive during input processing will be placed under the "NORMAL" archive set name unless the 'ISETNAME' is specified with an alternate name for the input data archived.

Note that the defaults parameter and keyword NIORATIO control how often the input phase "samples" the block ID of the input data stream. Because of blocking and input device lag the accuracy of the BID values obtained is dependent on how often the NOTE macro is issued. Theoretically it can be issued once for every READ operation (block input READ) but this results in very high overhead. As a compromise, the NIORATIO value is used to define a ratio of READ operations to NOTE operations. The smaller the value the more accurate the BID value returned but the longer the run will take. Very small values of NIORATIO will result in extremely long ARCHINPT run times (hours instead of minutes) and thus are not recommended. The default value as shipped will result in reasonably good ARCHINPT run times with fairly accurate BID values. As a safe guard, SMFUTIL will automatically force a NOTE macro each time the input date changes in the input data stream.

8.9 ARCHIVE BLOCK ID RETENTION

During ARCHIVING, SMFUTIL automatically retains Block ID values for data locations on tape datasets. These BID values are retained in slots in the archive record. Each BID slot represents the location on the tape of a specific hour or day. SMFUTIL decides to use hourly or daily based on the split type request for the DDA block. For daily type splits the hourly type value is used. For weekly or monthly type splits, a per day type value is used. For datasets that do not have a split requested via the DDAEND keyword, SMFUTIL defaults to a per day type value. This can be changed to hourly by specification of the BIDBYHOUR keyword in the DDA bloc. This is also valid for input DDA blocks while archiving input data. Note that if SMFUTIL detects a change in the day in the output data stream (or input stream during ARCHINPT) a BIDBYHOUR request (or default selection) is automatically suppressed and the BID value slots revert to one per day values.

8.10 ARCHIVE DATABASE UPDATE SEQUENCE

When ARCHIVE or ARCHUSER is requested (or ARCHINPT for input data sets), each time a volume of SMF data completes processing an ARCHIVE update request block is placed on an internal queue. At the end of the SMFUTIL run, if the total execution return code is less than 8 this queue of updates is processed to the Archive database. In processing the update requests the following rules apply:

1. If the volume does not currently exist in the archive, a new record is inserted.
2. If the volume does exist in the archive and it is a DASD volume and no record exists with this data set name, a new record is inserted.
3. For a tape data set with file sequence 1, if a record exists for this volume and file sequence number and the data set name differs, the record will be deleted and a new record inserted.
4. For a tape data set, if the volume, data set name, and file sequence number exist in a record in the archive and the data set disposition is MOD, the record will be updated in place. If the disposition is NEW, OLD, or SHR and the starting date/time of the old record matches the starting date/time of the new record, the record will be replaced (overlaid in place) otherwise the old record will be deleted and a new record inserted.

Note that updating a record differs from deleting a record and inserting a new one. The ARCHIVE database is a VSAM ESDS (Entry Sequence Data Set) with alternate indexes defined over it. This structure was selected for speed and ease of use. Its primary limitation is that records cannot be physically deleted. They must be logically deleted by placing a software delete flag indicator into the record and rewriting it. In order for a record in the database to be reused, the alternate index fields must all match the desired record that will reuse it. This means that the VOLUME, DSNNAME, and start DATE must match the previous record. If this criteria is not met, the old record must be deleted and a new one inserted. The old records physically remain in the database until the SMFUAMNT program is executed to perform maintenance on the database.

8.11 ARCHIVING MULTIPLE SYSTEMS

There is no limit to the number of systems that may be archived so long as each system maintains its SMF data on discrete data sets. If multiple systems are combined onto a single data set, the SMFUTIL archive has the ability to track from 1 to 16 SMF system identifiers (SID's) per SMF data set. This allows the installation to either commingle all systems SMF data onto a single data set or create a discrete output archive data set for each system. The installation should pick one and only one method. These two methods should not be mixed.

Although not recommended, in order to commingle data, all systems should be dumped and the data retained in holding data sets until a complete time period is collected (i.e. a full days worth). This data would be sorted together and then passed through SMFUTIL (with 'ARCHIVE' specified) to create a final output archive data set. In the same run, the data could be MOD'ed to a month to date tape to create a DUPLEX data set at the same time.

The recommended method is to keep each system on separate data sets. If more than 16 SMF SID's are to be retained, this is the only method that will work. This allows the data to be dumped directly from SMF to the final archive data set (i.e. MOD'ed) with the 'ARCHIVE' specification. To cause a logical separation of the data a SPLIT parameter could be specified on the DDAEND keyword for dynamically allocated output data sets. Note that the '&DID' symbolic parameter can be used in the DSNNAME DDA block keyword to cause the data set name used to be unique to the system being dumped. In addition, a GDG data set name structure or SUFFIX generation must be used to ensure a unique data set name will be used for each period from a split request.

SMFUTIL uses the SMF id as part of the required archive record recognition sequence during record creation and retrieval. Installations that change processors frequently, and change SMF id's each time, may find that they quickly surpass the 16 system limit in the archive. It might be helpful to limit the number of characters of the SMF id that SMFUTIL records in the ARCHIVE. This is done by setting the ASMFIDLN parameter in the SMFUDFLT module to 2 at installation time. You should then keep each environments SMF id stable (across processor changes) in the first 2 characters and only alter the last 2 characters. This allows multiple incarnations of processors under the same operating environment to keep adding data to the archive under the same SMF id name. The SMF id would then take on the characteristic of having the first two characters denote the operating environment (usage) and the last two characters denote the physical processor (CPU) on which it was active.

8.12 RESTRICTIONS AND LIMITATIONS ON ARCHIVE PROCESSING

The following rules should be kept in mind when designing processes that place data to the SMFUTIL archive database.

1. Do not use 'ARCHINPT' if one or more input data sets begins on other than the first volume of a multiple volume data set. When recording existing data sets to the archive via the 'ARCHINPT' keyword it is essential that the entire data set be processed. Do not use the 'IVSTART' keyword or specify a volume sequence number other than 1 in the DD statement. Do not concatenate data sets. Specify a unique SYSUT1xx DD name for each input data set.
2. When archiving input or output data, do not use more than one device per data set. This makes it impossible to determine which volume serial has just been completed.
3. Ensure that output disk data sets cannot be implicitly extended onto another volume (via any x37 prevention software utility program). SMFUTIL must know about multiple volume datasets when they are created (i.e. specify multiple units).
4. Do not specify a volume serial list for output tape data sets being archived. Allow the system to request 'PRIVAT' scratch volumes to extend the data set.
5. If possible, avoid archiving tape data sets that begin on other than file sequence 1 of a tape. Multi-file tapes are supported but they tend to clutter the archive with many records and will lead to degraded processing when accessing the archive to recall data. It also wastes the tape required for label usage. It is much more efficient to use a single data set structure and allow multiple volumes to accumulate for the data set name. Avoid the use of 'MOD' processing on a tape data set that is other than file sequence 1. This could cause it to spill over to an additional tape. If existing SMF archives exist on multi-file tapes, they should be condensed onto single file tapes prior to archiving.
6. Always use the catalog for SMF archive data sets whether on disk or tape. This will make maintenance on the ARCHIVE database much easier in the future.
7. Sites that routinely migrate and recall SMF datasets on disk should specify "FULLDASD=YES" in the SMFUFLT module to force SMFUTIL to read the entire disk dataset when accessing disk datasets via the ARCHIVE. When a dataset is recalled to disk from migration, it will likely be on different volumes and have more or less volumes than represented in the ARCHIVE. Sites that do not use HSM (or another migration product) should leave FULLDASD set to NO as shipped. This allows SMFUTIL to read only the portions of multiple volume disk datasets that are required to meet the specified date/time range specifications.

8.13 ARCHIVE DATABASE BACKUP

The SMFUTIL ARCHIVE database should be backed up frequently. This can be accomplished external to SMFUTIL or automatically via an internal SMFUTIL function.

The external process can be performed via an IDCAMS REPRO function. In shops with only one system, it can be done by appending a backup step to the SMFUDUMP PROC so the backup is automatically taken after each dump and archive update. When multiple systems are updating the archive, this is not as easy. It might be better to schedule a backup job on a daily basis. The JCL member ARCHBKUP in the INSTLIB data set can be used to perform the backup. Note that the GDG backup data set is on disk. For security, this GDG should be set up so that at least 5 generations are retained.

An easier method is to allow SMFUTIL itself to perform the backup on a periodic basis. Certain parameters of the SMFUDFLT member are used to define whether and when an archive backup is to be taken and what the backup data set name is to be. The following parameters control the backup process:

ARCBKUP When to perform the ARCHIVE backup function. Valid values are “NEVER”, “ALWAYS”, “DAILY”, “SUNDAY”, “MONDAY”, “TUESDAY”, “WEDNESDAY”, “THURSDAY”, “FRIDAY”, or “SATURDAY”. A default value of “NEVER” (the default as shipped) disables the backup function. “ALWAYS” will cause a backup to be taken whenever an update is performed against the archive. “DAILY” will cause a backup to be taken only after the first update each day. A day value will cause a backup to be taken only after the first update on the specified day of the week.

ARCBDSN Specifies the data set name of a new sequential data set to be used to back up the SMFUTIL ARCHIVE base cluster if ARCHIVE backup is enabled. It should be unique and reserved solely for the use of SMFUTIL. There should be no other datasets that begin with this prefix. This data set name may be a GDG or a data set name prefix that will be suffixed by SMFUTIL to make each backup dataset name unique. SMFUTIL will automatically detect if a GDG base is specified and set a relative GDG level of “+1” each time a data set is allocated. The GDG base must have been previously defined to the system for this to work correctly. If a non-GDG base is specified, a suffix will be attached to the data set name to ensure it is unique. If ARCBKUP=ALWAYS is specified, the suffix will be of the time format (i.e. ‘.Jyyyddd.Thmmss’) and the maximum data set name length specified by ARCBDSN is 27. If “DAILY” or a specific day of the week is specified a day suffix format is used (i.e. ‘.Jyyyddd’) as only one data set will be created on a specific day and the maximum data set name length specified by ARCBDSN is 35.

If ARCBDSN is specified as null in the DEFAULTS module or overridden with blanks via the ARCBKUP keyword, the ARCHIVEDB cluster name with the qualifier ‘.BACKUP’ appended to it will be used instead. This has the added benefit of tying the ARCHIVE cluster name to its backup datasets via the naming convention. The rules for GDG or date/time suffixing as defined for ARCBDSN still apply. Note that caution should be exercised with respect to the cluster name length so that sufficient room is available for the suffixes to be appended. As a rule, the following maximum lengths apply:

DATE/TIME suffixing	- ARCHIVEDB max length is 20 BACKUP=ALLWAYS
DATE suffixing	- ARCHIVEDB max length is 29 BACKUP=DAILY or greater
GDG DSN structure	- ARCHIVEDB max length is 29

ARCBNUMB Specifies the maximum number of ARCHIVE backup datasets to be retained in the system. When 0 is specified, SMFUTIL does no automatic maintenance of the backups. When a value of 1 to 999 is specified, SMFUTIL will automatically scratch and un-catalog older datasets when their count exceeds this number.

CAUTION: Specifying a value greater than 0 for ARCBNUMB will invoke the automatic backup maintenance function of SMFUTIL. It does this by scanning the catalog for candidate datasets for deletion. If the ARCBDSN index is not unique to the ARCHIVE backup process, it is possible that SMFUTIL will delete datasets that are not ARCHIVE backup datasets. Use caution in selecting the ARCBDSN index.

ARCBSID If specified, the archive backup function will only be performed on the system with the designated SMF ID. This effectively assigns responsibility for archive backups to one and only one system. If left blank, backups will be taken on any system that performs archive updates.

ARCBUNIT The unit name to be used to allocate the backup data set. The default value is “SYSDA”.

- ARCBSTYP** Space type to be used if ARCBUNIT is a DASD type device. This field must be set omitted or set to "TAPE" if a tape device is specified by ARCBUNIT. The default value is "CYL".
- ARCBSPRI** Primary space quantity for DASD backup data set. This field will be ignored if ARCBSTYP=TAPE was specified.
- ARCBSSEC** Secondary space quantity for DASD backup data set. . This field will be ignored if ARCBSTYP=TAPE was specified.
- ARCBVSER** A specific volume to be used to receive the backup data set. If omitted, (null value) or set to blanks, the system will determine the target volume of the data set.
- ARCBDCLS** An SMS Data Class to be assigned to the backup data set, if required.
- ARCBMCLS** A SMS Management Class to be assigned to the backup data set, if required.
- ARCBSCLS** An SMS Storage Class to be assigned to the backup data set, if required.

The ARCBACKUP keyword can be used at execution time to override the default values for the archive backup process and can selectively turned on or off the backup process. For example ARCBACKUP(NEVER) will disable the backup function. See the keyword definition ARCBACKUP on page 25 for more information on changing the backup processing function at execution time.

The ARCBKUP default of "NEVER" specified in the SMFUDFLT module should not be changed until the ARCHIVE backup strategy is completely thought out. The ARCBDSN default of "SMFUTIL.ARCHIVE.BACKUP" is probably not acceptable for most shops and should be changed to conform to local naming conventions. If the backup is to be taken to a GDG data set, the GDG base must be defined to the system before use. Although the archive backups can be targeted to a tape device this is not recommended due to the wait on the tape mount. The reserve on the ARCHIVE cluster will be held for this duration and this could cause system delays with other dump jobs. In most installations, the backup data sets are relatively small so using disk data sets is not a problem. If the ARCHIVE cluster is located on a STORAGE device, the backup unit name, volser, SMS class specification, or local DASD pooling specifications should be utilized to prevent the backup data set from being allocated on the same volume with the base cluster.

See the section entitled "MODIFICATION AND CUSTOMIZATION" on page 199 for more information on modifying the SMFUDFLT module and for information of how to define the archive backup defaults.

8.14 ARCHIVE DATABASE MAINTENANCE – BATCH MODE

Due to the structure of the archive database, records are logically deleted but not physically deleted during the updating process. This requires that occasionally the database be ‘cleaned out’. A utility program, SMFUAMNT, and supporting JCL is provided to accomplish this. Please see member ARCMANT in the INSTLIB for instructions on how to execute the utility program.

The maintenance process consists of four steps.

1. Backup (REPRO) of base archive cluster via IDCAMS.
2. Deletion of existing cluster.
3. Maintenance processing on backup data to create input to new database creation.
4. Construction of new database via IDCAMS.

The maintenance program accepts the following keywords for control purposes.

TARGETDSN(dsname-mask) Limits the actions to be taken by SMFUAMNT (as directed by other control verbs) to a records containing data set name that match the specified mask. An asterisk (“*”) may be used in any position to indicate that any character in that position is acceptable. In addition, the dsname-mask specification acts like a prefix when it is compared to the data set name in the archive record. They are only compared for the length of the dsname-mask specification. A single space may be placed after the dsname-mask to prevent unintentional matches with other data set names that begin with the same string. Multiple TARGETDSN keywords may be specified to allow actions to be applied to more than one data set name structure but not all data set names. When one or more TARGETDSN verbs are specified, all records with a data set name that does not match one of the specified target data set name masks will be passed directly to the output data set without being processed by other action verbs.

TARGETSET(setname) Limits the actions to be taken by SMFUAMNT (as directed by other control verbs) to a specific set or sets. If omitted, all set names will be valid for processing by control verbs. Multiple TARGETSET keywords may be specified to allow actions to be applied to more than one set but not all sets. When one or more TARGETSET verbs are specified, all records with a set name that does not match one of the specified target set names will be passed directly to the output data set without being processed by other action verbs.

NOTE: If the TARGETDSN and TARGETSET keywords are omitted, all data set names and set names will be valid for processing by other control verbs.

CAUTION: Defining a target set or target data set name mask does not prevent SMFUAMNT from dropping records with passed expiration dates or with a logical delete flag or user delete flag turned on. Use the ‘EXPIRED’, RESETEXPD, or SETEXPD keywords to prevent expired records from being deleted. Logically deleted records will always be dropped as they are unusable anyway.

CAUTION: User deleted records that must be retained as valid should be recovered (un-deleted) via the ISPF interface prior to running the SMFUAMNT utility.

EXPIRED(KEEP|DELETE) Instructs SMFUAMNT how to respond to expired records. DELETE (the default) indicates they should be dropped from processing (and thus from the database). KEEP indicates they should be retained and passed on to further processing.

KEEPDATE(yyyyddd) Indicates the earliest date that is to be retained in the archive. This allows stale data to be dropped if it is no longer required.

NOTCATLG(KEEP|DELETE) Instructs the program what to do with records with data set names no longer in the catalog (i.e. data sets that have been deleted or tapes that have been scratched). A specification of (DELETE) causes them to be dropped from the archive. A specification of (KEEP) retains them. The default is (KEEP).

DELETEDSN(dsname) Instructs SMFUAMNT to delete all records that contain the specified data set name.

DELETEVOL(volser) Instructs SMFUAMNT to delete all records that contain the specified volume serial.

DELETESET(setname) Instructs SMFUAMNT to delete all records that contain the specified Archive set name (i.e. "DUPLEX"). If any TARGETSET keyword verbs are specified, the set name to be deleted must be one of the target set names.

VERBOSE Instructs SMFUAMNT to produce a full report of all records retained after all processing has been completed. For a large database this may be a very lengthy report.

SETEXPD(n) Instructs SMFUAMNT to reset records with un-initialized expiration dates to the record's end date plus 'n' days. The value of 'n' may 'n' is 1 to 4 integer digits.

RESETEXPD(n) Instructs SMFUAMNT to reset expiration dates for all records to each record's end date plus 'n' days for all records, not just records without the expiration date field initialized.

CAUTION: The user specified days value is applied immediately and any resulting expiration dates will be honored. Records with calculated expiration dates older than (before) the current execution date will be dropped immediately.

A specification of **RESETEXPD(0)** resets all expiration dates to un-initialized (binary zeros) so that they will be ignored.

RENAME(oldname,newname) Instructs SMFUAMNT to change all occurrences of the set name "oldname" to "newname". This allows a site to alter an existing archive database with entries containing five character extracts from TYPETODD DD names to match a new set name now being specified via the ASETNAME keyword of a DDA block. USE EXTREME CAUTION WHEN UTILIZING THE RENAME COMMAND. IF RECORDS ALREADY EXIST THAT CONTAIN THE "newname" SET NAME, YOU MAY CAUSE DUPLICATION OF DATA DURING EXTRACTS AS MORE THAN ONE SET WILL EXIST WITH THE SAME SET NAME AND DATE/TIME RANGES. If any TARGETSET or TARGETDSN keywords verbs are specified, the set name to be renamed must be one of the target set names and/or must contain the specified dsnames.

8.15 ARCHIVE DATABASE MAINTENANCE – ISPF MODE

An extremely simple method of examining and logically changing the content of the archive database is via the "SMFUTIL ARCHIVE ISPF INTERFACE". The ISPF program SMFUADBU can be used to view, edit, and/or delete records contained in the archive. Member SMFUPRIM in the ISPLIB data set is an example of how to invoke the ARCHIVE ISPF function from an ISPF menu. Member SMFULBDF of the INSTLIB data set is a TSO CLIST that invokes the LIBDEF function for access to the SMFUTIL panel and message data sets. This CLIST is invoked via the "CMD" function in menu SMFUPRIM. Using the SMFUPRIM as an example, modify a local ISPF menu to invoke SMFULBDF.

8.15.1 ARCHIVE ISPF INTERFACE - ACCESS CONTROL

Access to the ISPF interface can be controlled on a userid basis via a table module called ADBUUSRS. By default, full access is allowed to all Userid's. This table can be used to limit access of to allow certain userid' read only access. To implement limited access follow the instructions in the ADBUUSRS table source module in the INSTLIB data set, re-assemble and re-link the table with SMFUADBU into a LNKSTxx data set for access from TSO ISPF sessions..

8.15.2 ARCHIVE ISPF INTERFACE - INSTALLATION

To install the ISPF interface, performing the following tasks:

1. Modify the SMFULBDF CLIST member of the INSTLIB data set to conform to the local data set naming standards used for SMFUTIL ISPF panels and messages (the names given during the unloading of the product tape) and place the CLIST into a data set available on the SYSPROC CLIST library concatenation for your logon procedure.
2. Modify the your local ISPF primary menu (or another appropriate local ISPF menu), as illustrated in member SMFUPRIM of the ISPLIB data set, to invoke the SMFULBDF CLIST.
3. Ensure the SMFUADBU load module is available in a STEPLIB or LNKSTxx data set.

8.15.3 ARCHIVE ISPF INTERFACE - OPERATION

Launch the ISPF interface by selecting the appropriate item on the primary menu modified during the installation. Once execution has begun the following screen will be displayed:

```

----- SMFUTIL ARCHIVE DATABASE DEFINITION -----
COMMAND ==>

Enter or change the SMFUTIL Archive Database name to be processed
and a specific set name or mask of names to be viewed.

Archive Database =>  SYSX.SMFUTIL.ARCHIVE
Set Name Mask =>   -      ( '-' to view all sets)
System Id      =>      (Blank for all systems)
Lowest Date    =>  2001001  (Blank for no lower limit)
Highest Date   =>  2002300  (Blank for no upper limit)
Show Deleted Records =>  Y

```

The first time this screen is seen you must enter the name of the SMFUTIL Archive base cluster, the name of the set of data to be viewed (i.e. “NORMAL”, “DUPLICATE”, or user defined set names) and whether or not user deleted records are to be listed in the table of data records displayed. After the first execution, this information will be retained by the utility in the user’s ISPF profile data set and redisplayed at the next execution. The low and high date specifications are optional. If specified, they are used to limit the range of data to be displayed.

NOTE: If user deleted records are listed they may be recovered or “undeleted” from the next panel displayed. Optionally, the set name may be entered as a mask to select more than one set. Entering a single dash (“-”) will cause all sets to be selected.

Press the “ENTER” key and the following panel will be displayed:

```

----- SMFUTIL ARCHIVE DATABASE DISPLAY ----- Row 17 of 27
COMMAND ==>                                SCROLL ==> CSR

Archive Database =>  SYSX.SMFUTIL.ARCHIVE
Set Name Mask =>   -
Enter S by a line to select the record for full detail and update.
Enter D by a line to logically delete the record from the database.

-----START-----END-----
|DATE      TIME| |DATE      TIME| VOLUME VSQ FSQ SET-NAME DSN (Truncated)
-----
2002001 00000000 2002031 15375330 W00031 1 1 NORMAL 0.NORMAL.Y2002JAN
2002034 08441908 2002059 23480162 W00023 1 1 NORMAL 0.NORMAL.Y2002FEB
2002060 00000000 2002090 23570169 W00035 1 1 NORMAL 0.NORMAL.Y2002MAR
2002091 00000000 2002120 23470793 W00025 1 1 NORMAL 0.NORMAL.Y2002APR
2002121 00000000 2002151 23545325 W00030 1 1 NORMAL 0.NORMAL.Y2002MAY
2002152 00000000 2002181 23511337 W00017 1 1 NORMAL 0.NORMAL.Y2002JUN
2002182 00000001 2002212 15244277 W00045 1 1 NORMAL 0.NORMAL.Y2002JUL
2002216 13361000 3897241 16221575 W00029 1 1 NORMAL 0.NORMAL.Y2002AUG
2002245 10424598 3897273 23501207 W00037 1 1 NORMAL 0.NORMAL.Y2002SEP
2002274 00000000 2002286 10491742 W00038 1 1 NORMAL 0.NORMAL.Y2002OCT
***** Bottom of data *****

```

This panel displays a scrollable table of data records found in the archive for the specified set or sets. Each line represents a single volume for a data set. The line shows the starting and ending date/time stamps for that volume. To delete a volume from the archive, place a “D” by the line it is on and a delete confirmation screen will be displayed. Press enter on the confirmation screen and the record will be logically deleted. Logically deleted records physically remain in the archive database until the next batch archive maintenance is run. They are simply ignored by SMFUTIL during an archive search for data to fill a request.

If logically deleted records are being displayed, each deleted records will have an “*” to the left of the line. To recover or “undelete” the record, place an “R” next to the line and press enter. A confirmation panel will be displayed. Press enter on the confirmation panel and the record will be returned to use.

To view the full detail information about a specific record, place an “S” on the line desired and press the “ENTER” key. The following panel will be displayed:

```

----- SMFUTIL ARCHIVE RECORD DETAIL DISPLAY -----
COMMAND ==>

Date Range => 2002245 10424598 Thru 3897273 23501207
Set Name   => NORMAL
Expire Date => 0000000
SMF SID's  => P390
           =>
Dsname     => SYSX.SMF.P390.NORMAL.Y2002SEP
Volser     => W00037
UCB Type   => 78008080      NON-Auth Unit Name => 3480
Volume Seq => 1             File Seq   => 1
Block Count => 1,357      Record Count=> 97,241
Created By  => SMFUDUMP     Created On  => 2002245 12321878
                               Modified On => 2002279 16371400

Block ID's: (EACH SLOT REPRESENTS ONE DAY OF DATA)
=> 01000004 0100004E 01000073 01000082 01000095 01000099 010000A4 010000B9
=> 010000EB 0100011D 01000168 01000199 010001CE 01000200 0100022A 0100023D
=> 01000287 010002A5 010002F8 01000308 0100031C 0100035C 0100039A 010003F0
=> 01000436 01000469 01000494 010004C0 010004FC 0100059C 010007DC

Change highlighted fields as required and press ENTER to update.
Press END to exit without updating.

```

This panel shows all information contained in the archive database about this specific volume. The first four items (DATE RANGE, SET NAME, EXPIRE DATE, and SMF SID's) and the Block ID's are editable and may be changed by the user. Caution should be exercised in making any changes as inappropriate alteration of any of these fields may cause SMFUTIL to skip required data during an archive search. If any changes are made to the record, when you press enter a change confirmation panel will be displayed. Press “ENTER” to confirm that the changes are to be made or “END” (PF3) to cancel the changes.

Please note that the “Block ID's” are only displayed for tape device data sets.

8.16 ARCHIVE DATABASE REPORTING

A batch utility program, SMFUARPT, is also provided to report on the contents of the archive database in a hardcopy format. Member ARCREPRT in the INSTLIB data set contains the JCL necessary to execute this batch program. The program accepts verb commands to indicate what records and what order the data is to be presented on the report.

INCLUDE filter-name=value Selects data records to be reported via data content. 'filter-name' may be DSN, VOL, SET, and SID. The 'value' specification is a masking string of the type to be used as the filter value to match against records. Any masking string may contain an "*" to indicate any character in that position is acceptable. In addition, DSN masking strings may contain a "-" (single dash) to indicate any further characters in the current data set name qualifier are acceptable, and a "--" (double dash) to indicate any further characters in the string are acceptable.

EXCLUDE filter-name=value Excludes data records from the report based on data content. 'filter-name' may be DSN, VOL, SET, and SID. 'value' is a masking string of the type indicated to be used as the filter value to match against records.

BY fieldname Selects the ordering field and causes a report to be produced. The parameter 'fieldname' may be 'DSNAME', 'VOLSER', 'DATE', or 'ENTRY'. DSNAME prints in data set name order. VOLSER prints in volume serial order. DATE prints in ascending date order for the starting date on each volume. ENTRY prints in entry sequence as originally recorded in the database. Multiple 'BY' statement control cards per execution are acceptable to produce more than one report sequence.

VERBOSE Instructs SMFUARPT to produce a multiple line report per record. By default, the SMFUARPT program produces a single line of pertinent information for each record in the database. The 'VERBOSE' keyword instructs SMFUARPT to produce a full information report. This report contains multiple lines per record entry and is more difficult to read but it contains full information on data in the archive.

USABLE Instructs SMFUARPT to report only records that are usable by SMFUTIL to retrieve archived SMF data. This suppresses the listing of defunct records such as ones for datasets that are no longer cataloged. If specified, USABLE must precede the "BY" keyword that selects the report sequence.

TERSE Instructs SMFUARPT to return to single line reporting.

RESET Instructs SMFUARPT to discard previously specified filtering criteria and return to full reporting. New filters may be specified for the new report.

AUDIT Instructs SMFUARPT to detect and report gaps between the ending date/time and the starting date/time for the next volume for a specific set name and system ID. To properly invoke the audit function, use the following control card setup:

```
INCLUDE SID=ssss
INCLUDE SET=NORMAL
AUDIT=mmmm
BY DATE
```

Where: "ssss" is the desired systems SMF SID value.

"mmmm" is the desired minute delta to look for in the range of 0 to 9999.

This must be a numeric value

If AUDIT is specified without a minute value, SMFUARPT will default to 15 minutes. It is required to limit the listing to a specific system and set name and to list by date to allow the gaps to be detected. Each gap detected is flagged in the REPORT output with message ID ARPT017. Message ARPT018 is issued once in the SYSPRINT output report.

NOTE: Gaps are detected ONLY between volumes, not within the data contained on a volume! The SMFUTIL function DATESCAN can be used to detect missing data within a data stream on a volume or dataset.

LOWDATE=ccyyddd Instructs SMFUARPT to not report on any dates lower than the specified date.

HIGHDATE=ccyyddd Instructs SMFUARPT to not report on any dates higher than the specified date.

ARCHIVE=cluster.name Instructs SMFUARPT to report on data in the specified SMFUTIL ARCHIVE. "cluster.name" is the predefined SMFUTIL Archive Database cluster to be reported on. Multiple Archive Databases may be reported on in a single run. Specify a new ARCHIVE keyword before each "BY" statement to change the database being reported on for that report. Once specified, a dynamically allocated archive remains in effect until a new ARCHIVE= verb statement is encountered or until the run ends. **RESET** has no effect on the Archive in use. It remains in use. A blank ARCHIVE= verb is ignored. The previous Archive name in effect, if any, remains in effect.

NOTE: Ensure that the SYSARCHx DD statements do not also appear in the JCL otherwise an error will occur. The archive must be supplied either by JCL or via the ARCHIVE= verb but not both.

EXAMPLE: An example report control set would be:

```

ARCHIVE=SMFUTIL.ARCHIVE
LOWDATE=2004.001
HIGHDATE=2004.120
INCLUDE DSN=SYSX.SMF.SYSA.--
VERBOSE
USABLE
BY DSNNAME
BY VOLUME
RESET
TERSE
BY DATE

```

This would produce two verbose reports for all data sets beginning with SYSX.SMF.SYSA, one ordered by DSNNAME and the other by VOLUME. The filtering criteria and the USABLE filter are removed by the RESET command and a full archive database report is produced in single line mode and ordered by DATE.

8.17 RETRIEVING SMF DATA USING THE ARCHIVE DATABASE

Any execution of SMFUTIL that is NOT dumping or clearing an SMF VSAM data set is known as a data extraction run. Any data extraction run that does NOT contain an explicitly specified input data set (i.e. SYSUT1xx in JCL or defined via a DDA control card block) or an implicit input data set via MANXALOC can utilize the SMFUTIL archive to locate the required data and allocate the data set(s) it resides on as long as the following conditions are met:

1. The name of the SMFUTIL archive database cluster must be available during the run. See the section titled "SUPPLYING THE ARCHIVE DATABASE NAME TO SMFUTIL" on page 179 for more information.
2. The execution must request an explicit date range via a DATE, GDATE, EDATE, or DTRANGE keywords or an implicit date range via a THISMONTH, PREVIOUSMONTH, THISWEEK, PREVWEEK, TODAY, or YESTRDAY.
3. If multiple SYSID's are specified (with the ARCHSID keyword defaulting to YES), GOTOEOF will be forced on. This prevents the defaulted SHORTCPY function from terminating the run prematurely before all data sets are passed, and thus allows SMFUTIL to recover all required SID values for the requested date ranges.
4. If multiple date ranges are specified (via the DATE, EDATE, GDATE or DTRANGE keywords) they should not overlap.

If no specific SYSID is requested, SMFUTIL will select data records from the archive based on the date range requirement only. If one or more SYSID's are specified, SMFUTIL will select records from the archive to supply data for each SYSID requested, for the date ranges specified. If systems are commingled on the SMF archive data set, this may result in the same data set being processed more than once. Each SYSID selected will cause the data set to be passed and selected from. This will not result in duplicated output data as only one SYSID is selected from the data set on each pass. The way to prevent this overhead is to avoid use of the SYSID keyword. The XSYSID keyword can be used, instead, to filter out undesired system identifiers. If this is undesirable, an alternative solution is to use the SYSID keyword to specify the required systems and to also specify the NOARCSID keyword to cause the archive search routine to ignore SID values in selecting archive records. Each record will only be selected one time, as a result of date range requirements. The SYSID values will then determine which records are passed to the output phase. Note that this scenario will only work if ALL SMF data from ALL systems are commingled onto ONE archive data set (plus one duplex set if applicable). If any system is kept separate, SMFUTIL will not be able to reliably recall SMF data. The separate system data may be selected for input even though it contains no required data.

Note that if the SMF data is maintained in discrete data sets for each system, the data will be produced in the order that the system id's occur in the database. This means that if 'SYSID(SYSA,SYSB)' were specified on a data extraction execution, SYSA would not necessarily be output first. The first archive record encountered containing a required system ID and a required date range would be processed first. The net result is that output data from a data extraction run almost assuredly will not be in sorted order. It may require sorting prior to use if sorted data is necessary to the target program that will use the data being extracted.

As previously described, SMFUTIL ARCHIVE classifies SMF data into "SETS". Each set has a unique "SETNAME". Whenever SMFUTIL creates an output data set and has been instructed to "ARCHIVE" (or ARCHIVEUSER) it SMFUTIL assigns a set name to the data set. The SETNAME assigned to an output archived data set is derived as follows:

1. Assigned by the user via the ASETNAME keyword in a DDA block definition for the output data set.
2. Assigned "NORMAL" if DDNAME is SYSUT2 (or is a split data set from SYSUT2) or "DUPLEX" if DDNAME is SYSUT2D (or is a split data set from SYSUT2D) and no ASETNAME keyword supplied by user in the DDA block or DD is in the JCL and not dynamically allocated.
3. Assigned the first five characters of the DDNAME if user defined output data set and no ASETNAME keyword supplied by user in the DDA block or DD is in the JCL and not dynamically allocated.

During a data extract run that retrieves data via the ARCHIVE, SMFUTIL will default to the "NORMAL" set name. This may be overridden by the user via the SETNAME keyword. The user may specify one or more sets to be retrieved. If multiple sets are to be retrieved, it is the users responsibility to ensure that the sets are defined and created in such a way as to ensure the intended result. When more than one set is retrieved without due consideration to the results, the output data may contain multiple copies of the same data or be output in asynchronous order (not in sorted order).

8.18 ARCHIVED DATA SET UNIT NAME DEFINITION

During dynamic allocation of archive data sets for data extraction runs that are not authorized (i.e. not executed from LNKSTxx or authorized library), SMFUTIL utilizes unit names derived to match the device type that was in use when the SMF archive data set was created. The DEVTYPE macro is used to extract the UCB device type for the output device. The four byte device characteristics word returned is saved in the archive record. During allocation for retrieval, if SMFUTIL is authorized, this UCBTYPE definition will be used as the unit definition for allocation. If SMFUTIL is not authorized, dynamic allocation does not allow device type definitions to be used as unit names for the four byte UCBTYPE is matched against defined device type masks in a table. Each mask has associated with it an 8 character unit name to be used for devices matching that mask. The first byte of UCBTYPE defines model definitions and is currently not required by SMFUTIL to uniquely define the unit in question. The second byte defines special features installed on the device and is currently used to indicate the presence of IDRC on 3480 devices. The third byte defines the device class. Disk units are device class x'20' and tape units are device class x'80'. The forth byte defines the device type within class. For example, x'0F' means a 3390 device when device class is x'20'. For information on how to locally modify the unit name table to include additional devices, please see the section titled "MODIFYING THE UNIT NAME TABLE" on page 213.

9. DYNAMIC ALLOCATION OF DATA SETS

SMFUTIL has the ability to dynamically allocate both input and output data sets. The DDA block command structure is designed to allow great flexibility in the specification of data sets to be allocated. This has the additional advantage of deferring the allocation of an output data set until its actual use is required. This precludes the required device allocation that JCL statements would cause even though the data set may be only used once a month (for a month split for example). For complete syntax of each DDA block statement, please see the section entitled "CONTROL KEYWORDS - DDA BLOCK" beginning on page 120.

9.1 DDA BLOCK EXAMPLES

The DDA block structure takes on the following format:

```
DDASTART
    DSNAME(SYS1.MAN1)
    DDNAME(SYSUT1M1)
DDAEND
```

Note that the keywords may be placed on a single card or split among multiple cards as long as each keyword with its parameter list is completed on a single card. The only keywords valid between the DDASTART and DDAEND commands are defined as DDA block commands. For example

```
DDASTART
    DSNAME(SYS1.MAN1)
    DDNAME(SYSUT1M1)
    DETAIL
DDAEND
```

is invalid as the "DETAIL" keyword is a normal keyword not a DDA block keyword. DDA block commands may appear in any order between the DDASTART and DDAEND keywords. Note also that each DDASTART/DDAEND block structure represents a single allocation for input or output usage. It may IMPLY multiple data sets by specifying a "SPLIT" sub-parameter within the DDAEND keyword parameter list but, in general, it represents a single input or output stream.

9.2 DDA BLOCK SPLIT REQUESTS

The SPLIT sub-parameter is used to dynamically set up multiple output data sets in support of a split request. The SPLIT sub-parameter can be logically viewed as a way of 'chopping' the output data stream into pieces and each piece is to be contained in a unique data set. For example:

```
DDASTART
    DDNAME(SYSUT2)
    DSNAME(SYSX.SMFSYSA.DAILY(0))
    DISP(MOD,CATLG,KEEP)
    UNIT(TAPE)
DDAEND(,SPLIT(D,4))
```

illustrates the SPLIT sub-parameter. The existing generation of the data set 'SYSX.SMFSYSA.DAILY' will be allocated to DD name SYSUT2 (normal output) for 'MOD' processing. If a change in day occurs (data passing midnight), a split will occur, SYSUT2 will cease to receive data and SYSUT3 will be dynamically allocated as '+1' of the GDG. Note that the disposition of SYSUT3 will be 'NEW,CATLG,DELETE'. If another midnight occurs, the same process repeats with SYSUT4 being allocated to '+2' of the GDG. The maximum number of splits that may be specified is 34 with the minimum at 2.

For the DDA SPLIT request to do its job properly, it is important that the input data be in chronologically correct order. If data is presented out of sequence, a warning message will be issued. Note that no split will occur, even if the regression is across a requested split boundary. This prevents duplicate data set name creation via suffix generation when multiple pieces of the output data are in an out of sort sequence. This does not imply that SMF data must be sorted prior to being processed by SMFUTIL for the SPLIT function. It only means that steps should be taken to ensure that the data is dumped in the order it was created (written to the SMF MANx files).

9.3 DDA BLOCK DATA SET NAME SUFFIX GENERATION

Usage of GDG's is sometimes confusing as it is difficult to tell what data they contain. The SMFUTIL archive database can be used to overcome this difficulty by 'remembering' where data is stored and recalling it for you. An alternative is to automatically assign data set name suffixes based on the current date or the date stamp of the first record placed to the data set. Two type of suffix formats are supported, daily and monthly. The DDA block structure can be used to set this up as follows:

```
DDASTART
  DDNAME(SYSUT2)
  DSNAME(SYSX.SMFSYSA.DAILY)
  SUFFIX(DD)
  DISP(MOD,CATLG,KEEP)
  UNIT(TAPE)
DDAEND(,SPLIT(D,4))
```

The data set name would be appended with a daily suffix format of '.Jccyyddd' where 'cc' is the century, 'yy' is the year, and 'ddd' is the Julian date. The date used to generate this suffix is extracted from the first record that required output. If the data set already exists (created earlier in the day) it will be MOD'ed onto. If it does not exist, the MOD will automatically be changed to NEW and the data set will be created. Since the data set is created dynamically when it is required to receive data, there is no problem with disk data sets getting allocated and never receiving data. At midnight, a new data set would be allocated utilizing the new date to create a new data set suffix yielding a new data set name and thus a unique catalog entry.

9.4 DDA BLOCK SYMBOLIC SYSTEM ID SUBSTITUTION

SMFUTIL support two special case symbolic substitution parameters within the data set name.

For an input or output data set, if the data set name contains the character string '&SID' anywhere within the name, this string will be replaced with the SMF system identifier of the system on which the job is executing. The CKPTDB control keyword also supports the '&SID' symbolic.

For an output data set only, if the data set name contains the character string '&DID' anywhere within the name, this string will be replaced with the SMF system identifier extracted from the first SMF data record processed.

These two symbolic parameters make it possible to create a single set of SMFUTIL control cards and JCL procedures that can be used for multiple operating systems.

Note that if system symbolic substitution is enabled, caution should be exercised to avoid conflicts with SMFUTIL internal symbolic terms.

9.5 AUTOMATIC DATE INSERTION INTO DATA SET NAME STRING

SMFUTIL has the ability to automatically substitute (overlay) certain parameter specifications in data set name strings with corresponding date information. For output data sets, the date used for making the replacement is taken from the first record to be placed to the data set.

The supported substitution parameters are as follows. The text shown in the replacement example is for Sunday, December 1, 2002:

&YJJJ	Would be replaced with 03335
&YYYYJJJ	Would be replace with 2002335
&MDDYY	Would be replaced with 120102
&DMMYY	Would be replaced with 011202
&MMYY	Would be replaced with DEC02
&MMYYYY	Would be replaced with DEC2002
&MMDDYY	Would be replaced with DEC0102
&MMDD	Would be replaced with DEC01
&MM	Would be replaced with DEC
&M	Would be replaced with 12
&JJ	Would be replaced with 335
&YYYYMM	Would be replaced with 200212
&YYY	Would be replaced with 2002
&Y	Would be replaced with 02
&C	Would be replaced with 20
&DD	Would be replaced with FRI
&DMMYY	Would be replaced with 01DEC02

Each symbolic is exactly the same length as the string that will replace it (starting with the "&").

Note that for the data set name syntax to be valid, replaced strings that begin with a numeric (i.e. 02335) must be preceded by an alphabetic character within the qualifier. For example, 'SYSX.&YJJJ' would not be valid as 'SYSX.02335' is an invalid data set name string according to IBM syntax rules.

Note that this capability is primarily intended for the deferred allocation of output data sets. For output dsnames that are specified with "ALLOCATE" in the DDAEND keyword or for input data set name specifications, the current (execution) date will be used for the substitution. This may or may not be what is intended. The careful attention of the user is required to ensure the result is as desired.

Note that each parameter specification is exactly the same number of characters as the string that will replace it. This allows for the data set name length to be unchanged after the substitution is made. Certain combinations are not valid and are there for not supported. An example would be the full date string of "12012002". This string is 8 characters long and begins with a digit. This is invalid for a data set name qualifier and cannot be prefix with an alphabetic character to make it valid because the resulting string would be longer than 8 characters which is also invalid.

Combinations may be used. Consider the following examples, again using the December 1, 2002 date:

'SMF.&DD.Y&YYYYJJJ.DATA'	would yield	'SMF.SUN.Y2002335.DATA'
'SMF.&DD.&MMDD.Y&YYY'	would yield	'SMF.SUN.DEC01.Y2002'
'SMF.&MMDDYY.&DD'	would yield	'SMF.DEC0102.SUN'
'SMF.Y&YYY&MM.DATA'	would yield	'SMF.Y2002DEC.DATA'

The possible combinations are numerous. It is left to the user to implement this capability to local requirements.

One additional consideration that must be examined is the use of embedded date strings for data sets used with data splitting. For SPLIT to be specified in a DDAEND parameter for a non-GDG output data set, SMFUTIL normally requires the data set name be made unique for each split taken. This is normally accomplished via the SUFFIX keyword. When a SUFFIX parameter is used to append a suffix onto a data set name in a DDA block, SMFUTIL automatically determines if the granularity of the suffix is sufficient to ensure the data set name will always be unique. For example, you cannot use a month suffix on a data set being split on a day boundary as each data set within the same month would yield the same suffix. If an embedded date parameter is used to provide data set name uniqueness across a split, the SUFFIX requirement and the granularity check are bypassed. **It is the user's responsibility to ensure the embedded date parameter is sufficient to provide uniqueness across a split boundary.**

Likewise, you should avoid a data set name date insertion with too fine a granularity for any split that is requested. For example, if a data set name of "SYSX.SMF.Y&YYYJJJ.MONTHLY.DATA" were used to receive data for a DDA block that specified a month split, the data set name would change each time the DUMP job was run on a new day. This would cause the data set disposition to be changed from MOD to NEW and a new data set created even though a split boundary had not been passed. A data set name of "SYSX.SMF.Y&YYY&MM.MONTHLY.DATA" would be more appropriate as the inserted date field only changes when a month boundary is passed.

NOTE: If system symbolic substitution is enabled, caution should be exercised to avoid conflicts with SMFUTIL internal symbolic terms such as &SID and &DID.

10. MODIFICATION AND CUSTOMIZATION

10.1 SETTING EXECUTION CONTROL DEFAULTS

Some of the control keywords may be preset as defaults in a control section (CSECT) called SMFUDFLT. This module is generated via the SMFDFLT macro contained in the INSTLIB data set. The sample module SMFUDFLT in INSTLIB is the setup of the defaults as shipped in the program. The member USERDFLT is an duplicate copy of SMFUDFLT. It is recommended that any changes be applied to USERDFLT and SMFUDFLT be left unchanged for references purposes. The INSTALL job stream used to install the product will automatically assemble the USERDFLT member for local usage. The keywords that may be specified in the module, their default setting, and usage are as follows:

PARAMETER	DEFAULT	USAGE
ACTIVEOK	YES	Can the active SMF data set be used as input?
AEXPIRE	0	How long should ARCHIVE records be considered valid (in days) after they are last updated. The default of 0 means permanent retention.
ALL	YES	Are all record types included by default?
ALLOW2	NO	Should old type 2 records be retained in the output data set(s)? ("NO" means ignore them.)
ALLOW3	NO	Should old type 3 records be retained in the output data set(s)? ("NO" means ignore them.)
ALTUNITS	NO	To instruct SMFUTIL whether to use the Tape Alternate Unit Name Table, SMFUTALT, to substitute a unit name for a UCB device type string during allocation for data retrieval using the SMFUTIL Archive.
ASMFIDLN	4	Number of characters of the SMF identifier to used in the archive database records to be able to recognize a system. This should normally be left at 4. Installations that change processors frequently may wish to limit this to 2 and then keep the SMF id stable (across processor changes) in the first 2 characters. This allows multiple incarnations of processors under the same operating environment to keep adding data to the archive under the same SMF id name. The SMF id would then take on the characteristics of having the first two characters denote the operating environment and the last two characters denote the physical processor it was active on.
ARCBKUP	NEVER	When to perform the ARCHIVE backup function. Valid values are "NEVER", "ALWAYS", "DAILY", "SUNDAY", "MONDAY", "TUESDAY", "WEDNESDAY", "THURSDAY", "FRIDAY", or "SATURDAY". "NEVER" disables the backup function. "ALWAYS" will cause a backup to be taken whenever an update is performed against the archive. "DAILY" will cause a backup to be taken only after the first update each day. A day value will cause a backup to be taken only after the first update on the specified day of the week.

ARCBDSN		<p>'SMFUTIL.ARCHIVE.BACKUP' Specifies the data set name prefix of a new sequential data set to back up SMFUTIL ARCHIVE cluster to if ARCHIVE backup is enabled. This data set name prefix may be a GDG base index or a data set name prefix that will be suffixed by SMFUTIL to make it unique. SMFUTIL will automatically detect if a GDG base index is specified and set a relative GDG level of "+1" each time a data set is allocated. The GDG base must have been previously defined to the system for this to work correctly. If a non-GDG base is specified, a suffix will be attached to the data set name to ensure it is unique. If ARCBKUP=ALWAYS is specified, the suffix will be of the time format (i.e. '.Jyyyyddd.Thhmmss') and the maximum data set name length specified by ARCBDSN is 27. If "DAILY" or a specific day of the week is specified a day suffix format is used (i.e. '.Jyyyyddd') as only one data set will be created on a specific day and the maximum data set name length specified by ARCBDSN is 35. If the GDG format is used, a limit should be specified during its definition. At least 3 should be kept at all times. Suffixed backup data sets that are no longer needed may be scratched automatically by changing the SMFNUMB value below to a positive value above zero. Care should be taken to ensure that an adequate level of backup data sets are retained at all times.</p> <p>NOTE: If ARCBDSN is specified as null, the ARCHIVE cluster name (ARCHIVDB keyword value or ARCHIVDB default parameter value) with the qualifier '.BACKUP' appended to it will be used instead. The rules for GDG or date/time suffixing as defined above still apply. Note that caution should be exercised with respect to the cluster name length so that sufficient room is available for the suffixes to be appended.</p>
ARCBNUMB	0	<p>The number of ARCHIVE BACKUP datasets that are to be retained. Valid range is 0 or 1 to 999. When set to 0 (the default) SMFUTIL takes no action to eliminate prior archive backup datasets. When set to 1 or greater, after the current backup is completed, SMFUTIL will automatically scan the system catalog for all datasets that begin with the archive backup prefix (ARCBDSN). If more datasets exist than the ARCBNUMB value, older datasets will be scratched and un-cataloged until the number of datasets remaining is equal to the ARCBNUMB value. When set to a non-zero value, a value of at least 3 should be specified for safety.</p> <p>NOTE: When ARCBDSN specifies a GDG base index, the ARCBNUMB value should be left at 0 and the GDG specification in the catalog should be defined so as to limit the number of generations of the GDG that are to be retained.</p>
ARCBSID	null	<p>If specified, the archive backup function will only be performed on the system with the designated SMF ID. This effectively assigns responsibility for archive backups to one and only one system. If left blank, backups will be taken on any system that performs archive updates.</p>
ARCBUNIT	SYSDA	<p>The unit name to be used to allocate the backup data set. The default value is "SYSDA".</p>
ARCBSTYP	CYL	<p>Space type to be used if ARCBUNIT is a DASD type device. This field must be set omitted or set to "TAPE" if a tape device is specified by ARCBUNIT. The default value is "CYL".</p>
ARCBSPRI	5	<p>Primary space quantity for DASD backup data set. This field will be ignored if ARCBSTYP=TAPE was specified.</p>
ARCBSSEC	5	<p>Secondary space quantity for DASD backup data set. This field will be ignored if ARCBSTYP=TAPE was specified.</p>
ARCBDCLS	null	<p>An SMS Data Class to be assigned to the backup data set.</p>
ARCBMCLS	null	<p>An SMS Management Class to be assigned to the backup data set.</p>
ARCBSCLS	null	<p>An SMS Storage Class to be assigned to the backup data set.</p>
ARCFORCE	NO	<p>Should SMFUTIL force ARCHIVE updates when an ABEND occurs during an SMFUTIL ARCHIVE execution?</p>
ARCHIVDB	blanks	<p>SYSARCH database name to be used when ARCHIVDB control parameter is not present. Blanks indicate no archive database is available?</p>

NOTE: If ARCBDSN is specified as null, the ARCHIVE cluster name (ARCHIVDB keyword value or ARCHIVDB default parameter value) with the qualifier 'BACKUP' appended to it will be used instead. The rules for GDG or date/time suffixing as defined for ARCBDSN still apply. Note that caution should be exercised with respect to the cluster name length so that sufficient room is available for the suffixes to be appended. As a rule, the following maximum length apply:

DATE/TIME suffixing - ARCHIVDB max length is 20 BACKUP=ALLWAYS
 DATE suffixing - ARCHIVDB max length is 29 BACKUP=DAILY or greater
 GDG DSN structure - ARCHIVDB max length is 29

ARCHSID	YES	Should the SMF system ID be used for search of the archive database?
ARCHSMF	NO	Should updates to the ARCHIVE database be protected via before and after image records cut to the SMF system. If 'YES' is specified, ARCHSMF# must contain a valid record number to be used. This may be disabled at execution time via the "NOARCHSMF" keyword. If 'NO' is specified and ARCHSMF# specifies a valid record type (129-255), ARCHIVE update record cutting may be enabled at execution time via the ARCHSMF keyword.
ARCHSMF#	0	A valid SMF record number in the range of 129 to 255 or 0. If ARCHSMF=YES is specified, this number will be used to cut SMF records containing before and after images of the updates to the SMFUTIL ARCHIVE database. If '0' is specified, ARCHIVE update record cutting is disabled but may be enabled at execution time via the ARCHSMFRECNUM keyword.
BDETAIL	NO	Should SMFUTIL produce the BYTEDETAIL report by default. This report shows the total bytes and percent of bytes moved for each record type.
BIDINDEX	1	Number of days of data represented per Block ID slot in the archive record when a daily type slot is being used. There are 31 slots in the archive record for each volume. This is designed to allow each volume to contain up to one month of data and still have each day individually addressable with a block ID for high speed retrieval. Do not change this value unless you will routinely store more than one month of data per VOLUME (not per data set). BIDINDEX has no effect on hourly type BID slot calculations.
BIDPOINT	YES	Should SMFUTIL use the POINT macro to high-speed position cartridge drives on data retrieval.
BDETAIL	NO	Should SMFUTIL produce the total bytes moved report?
CKPTINPT	YES	Should input date/time value of first record be check pointed to prevent accidental rerun?
COPY0OK	NO	Should runs that copy no data end with a return code indicating an error?
COPYBOTH	NO	Should records copied to a user targeted DDNAME via TYPETODD also be copied to the standard output DD's (i.e. SYSUT2 and SYSUT2D)?
COPYVBTH	NO	Should records copied to a value targeted DDNAME via a value testing keyword (i.e. CICSSPLIT , SIDSPLIT, etc.) also be copied to the standard output DD's (i.e. SYSUT2 and SYSUT2D)?
CREATES0	NO	Should the standard output data sets (SYSUT2 and SYSUT2D) be created (opened) even if no data is placed to them?
CREATEU0	NO	Should User output data sets defined via TYPETODD statements be created (opened) even if no data is placed to it?
DACKPT	YES	If a checkpoint PDS specified via the CKPTDB keyword does not exist, should SMFUTIL create it automatically? If 'YES' is specified, SMFUTIL will allocate the PDS using the unit, volume, primary space and directory blocks specified below.
DACKPTU	SYSDA	Unit name to be used to create a CKPTDB PDS checkpoint dataset.
DACKPTV	blanks	Volume serial a created CKPTDB PDS is to reside on.
DACKPTP	1	Primary amount of space, in tracks, for a created CKPTDB PDS.
DACKPTD	40	Number of directory blocks for a created CKPTDB PDS.

DARETRY	NO	Should dynamic allocation attempts that fail for lack of available resources (data set name or volume in use or no units available) always be retried. Individual DDA block DDARETRY specifications will override this value.
DATESCAN	NO	Should gaps in the date/time sequence between records be searched for?
DCACKPT	'SYS1.SMFUTIL.DMPCLALL'	DUMPCLEARALL checkpoint data set name to be allocated during DUMPCLEARALL execution. This data set is used for restart purposes in case of an interrupted run. Note that the character string ".SYSxxxx" will be appended to the string where "xxxx" is the SMF system ID being executed on. This ensures a unique data set name structure for each system using the DUMPCLEARALL process.
DCADCLAS	null	SMS Data Class to be used to allocate DCA Checkpoint data set.
DCAMCLAS	null	SMS Management Class to be used to allocate DCA Checkpoint data set.
DCASCLAS	null	SMS Storage Class to be used to allocate DCA Checkpoint data set.
DCAPRIM	1	Primary space value for DCACKPT (DUMPCLEARALL processing) data set in tracks.
DCAUNIT	'SYSDA'	Unit name to use to allocate DCACKPT data set.
DCAVSER	null	Volume serial where the DCACKPT data set will be allocated. If left blank, any valid "DCAUNIT" type device will be eligible.
DCBABXIT	YES	Should DCB Abend exits be used to intercept ABENDs that may be safely ignored?
DCBRESET	YES	Should OLD disposition data set DCB characteristics be reset to optimum?
DDRETAIN	10	Retain this many previous records for duplicate record compares during DELDUP execution.
DELETERC	7	Instructs SMFUTIL as to the return code at or below which dataset deletions requested via the CATINPUT or GDGINPUT control keywords will be performed. This is the MAXIMUM return code that will allow deletions. Any return code above the value will suppress deletions. The acceptable values of 3 or 7 are intended to allow a run completion of 0 or 4 plus any split indication return code of 1, 2 or 3.
DETAIL	YES	Should the detailed record statistics report be produced?
DFLTLIST	YES	Should execution defaults be listed to SYSPRINT during initialization? This can be overridden at execution time by a PARM=DFLTLIST to turn on the listing or PARM=NODFLTLI to turn off the listing.
DFLTMBR	null	The name of a member in the PDS defined by the PARMLIB keyword that is to be read and used to update the execution defaults table prior to reading the SYSIN control statement data set. If left blank (the default) no default override will be used unless specified via execution parm or in the JCL PARMLIB DD statement.
DSDELTA	60	If DATESCAN finds a between record date/time gap greater than this number of minutes, report it.
ESTAEON	YES	Should ESTAE ABEND protection be used during record movement?
FASTMOD	YES	Should SMFUTIL utilize high-speed positioning for MOD data sets?
FORCE	YES	Should DATESCAN continue if out of order data is detected?
FULLDASD	NO	Should SMFUTIL read the entire dataset when a disk dataset is used to satisfy a specific date/time range data retrieval using the SMFUTIL ARCHIVE? A setting of "NO" allows SMFUTIL to input only the specific pieces of a multiple volume disk dataset required to satisfy the request. A setting of "YES" causes the entire dataset to be allocated to satisfy the input request. Sites that use HSM (or another migration product) should set this parameter to "YES" because recalled datasets may no longer reside on the same volser's or number of volumes as before migration.
GDGMODEL	'SYS1.MODEL'	Use this name as GDG model name when MODEL keyword is omitted from DDA block for a GDG data set allocation.

GDGCTRL	YES	A specification of 'YES' allows SMFUTIL to internally control the generation of relative GDG suffixes. This effectively prevents SMFUTIL from ENQ'ing on the base GDG name to avoid preventing other jobs from accessing other levels of the same GDG base. A setting of 'NO' instructs SMFUTIL to handle relative GDG data set names in the normal fashion. This keyword setting may be overridden at execution time via the "GDGCONTROL" and NOGDGCONTROL" keywords.
GOTOEOF	NO	Should input continue to the physical end of the data set when date/time values higher than highest required are encountered and the data is in order? GOTOEOF=YES can be negated at execution time by the "STOP" keyword.
IDETAIL	NO	Should detailed statistics be reported on each input data set?
IOELIMIT	25	Maximum number of consecutive input data I/O errors that will be tolerated prior to terminating the run. This avoids flooding the operator console with IOS0001 messages due to a bad input data set. Valid range is 0 to 999999.
ISMF	NO	Should a switch always be performed during a DUMPCLEARALL execution. Ignored if MAN1LEAV is specified (or defaulted) and the primary data set is the only available data set. Also ignored if all data sets are marked "DUMP REQUIRED" except the active data set. NOTE: "ISMF=NO" will be ignored if "MAN1LEAV=YES" is specified. The MAN1LEAV parameter overrides and allows SMFUTIL to perform a switch (if required) to satisfy the MAN1LEAV request.
ISMFSYNC	null	Name of the routine to be invoked to synchronize the SMFUTIL switch of SMF datasets with an external OEM product. Leave this parameter set to null unless it is required for your other product(s) use of the MANx datasets. The only known product to require this service is the "SMF EXTRACTOR" from 21 st Century Software. If this product is in use you may have to specify ISMFSYNC=SMFWAIT to cause the "SMF EXTRACTOR" to be notified when an SMF switch occurs.
LARGEBLK	NO	Should SMFUTIL create Large Block Interface type output datasets by default. NOTE: This should not normally be changed, as creating LBI datasets by default for all outputs may render some programs unable to read data they require if they do not support the LBI specification.
LBI	YES	Should SMFUTIL enable the LBI interface for input and output datasets. If LBI=NO is specified, input datasets that contain large block datasets will cause an ABEND. Output datasets will not be created with large blocks unless LARGEBLK is specified (or defaulted) or unless the DDALARGEBLOCK keyword is specified in the DDA block for a specific dataset. This parameter should normally be left set to YES as shipped.
LCNSMBR	null	The name of the member in the PDS specified by PARMLIB that contains a valid SMFUTIL execution license. If left blank (the default) no additional license information will be read and the license installed in the load module will be used.
LOCATE	YES	Should a data set name found in the ARCHIVE database be verified as still cataloged before it is used as input?
MAN1LEAV	YES	Should the primary SMF data set be left as AVAILABLE after DUMPCLEARALL has finished execution? This allows a "Z EOD" to be performed without data being orphaned in the an alternate SMF data set.
MAXABEND	25	Maximum number of ABEND retries before terminating.
MNTHPCAL	null	The site defined MONTHPCAL (perpetual month calendar) table to be used for all month splits and data set name month suffixing. If specified, the table must be pre-installed into a LNKLSSTxx data set or a STEPLIB dataset.
MONTHCAL	null	The site defined MONTHCAL (month calendar) table to be used for all month splits and data set name month suffixing. If specified, the table must be pre-installed into a LNKLSSTxx data set or a STEPLIB data set.

MULTICLR	NO	Should multiply MANx datasets be clear simultaneously during a DUMPCLEARALL execution?
NIORATIO	25	The maximum number of READ operations to NOTE macros during ARCHINPUT processing. The larger this number the faster the run. The smaller the number, the more accurate the BID values recorded will be. The value range is 1 to 999,999,999. Because of buffering, there is a lag between SMF record date values and BID (Block ID) values that represents a position on the tape for that date. A value of 1 will cause a NOTE macro to be recorded for each CHECK macro issued and thus result in a very accurate BID value for the current DATE being seen in the input data stream but at a very high overhead cost. A NOTE macro is forced each time a date change is seen in the date stream to ensure that BID values used are never more than one day off. Normally, the default value should not be changed unless advised to do so by SMFUTIL Technical Support staff.
ODETAIL	NO	Should detailed statistics be reported on each output data set?
ORDERCHK	NO	Should the order of SMF data set be verified? This means that each execution checks to see if the data from the (first data set if DUMPCLEARALL is used) SMF data set being dumped is logically higher than the last data set dumped in the previous run and without a large gap between them?
ORDERGAP	300	ORDERCHK uses this number as the maximum number of minutes that may exist between the last record of the previous dump and the first record in the current dump.
PARMLIB	blanks	The name of a PDS data set that contains a valid license member (specified by LCNSMBR) for executing SMFUTIL. Note that this should <u>NOT</u> be “SYS1.PARMLIB”.
PROCESS	YES	Record the process name and execution system id (SMF id) in the \$PROCESS checkpoint record and verify upon each execution. This prevents the same checkpoint data set from being inadvertently used for more than one process. The process name defaults to the job step name unless the PROCESS keyword is specified at execution time.
RSRVOLD	YES	Should SMFUTIL use a RESERVE protocol compatible with previous releases. Set to “YES” if you are testing this version of SMFUTIL while a previous version is still in production usage (i.e. version 6.0 and below). Set to “NO” when this version is installed into production usage and all other versions have been removed from use.
SCRRETRY	30	Number of retries to be attempted for a failed scratch SVC as a result of a GDGINPUT or CATINPUT DELETE request. Valid values are 0 to 999 with 0 meaning no retry is to be attempted.
SCRWTIME	30	Length of time in seconds to wait before retrying a failed scratch attempt. Valid values are 0 to 60 with 0 meaning no retry is to be attempted.
SCLRTHLD	80	SHORTCLEAR/FULLCLEAR transition threshold percentage. When a SHORTCLEAR is requested during a DUMPCLEARALL execution SMFUTIL will automatically transition to the more efficient full clearing process if the MANx data set is at or above this percent full.
SDB	NO	Is SDB (System Determined Blocksize) active in the operating system. If set to ‘YES’ SMFUTIL will never fill in a BLKSIZE value when one is not specified in an output data set, but will trust the operating system to choose an appropriate BLKSIZE value based on the supplied (or defaulted) LRECL and RECFM values. If set to ‘NO’, SMFUTIL will supply an appropriate block size value for JCL defined output data sets with ‘0’ as the BLKSIZE value. For DDA block defined output data sets, SMFUTIL differentiates between a specification of ‘0’ and an omitted block size. If ‘0’ is explicitly specified, it will not be overridden by SMFUTIL, regardless of the SDB parameter, and the system will be allowed to supply the block size. If the block size is omitted, the SDB parameter controls if SMFUTIL supplies a block size or not, just as for JCL data sets.

SHORTCPY	YES	Should we stop execution after the maximum date/time value requested is passed, even if the data was not in order or sort order checking is not in progress? SHORTCPY=YES can be negated at execution time by the "NOSHORT" keyword.
SHORTCLR	YES	Should the DUMPCLEARALL process use a smart clearing process that only overwrites data blocks actually containing data that was read in? SHORTCLR=YES can be negated at execution time by the "NOSHORTCLEAR" keyword.
SMFIDCHK	YES	Should SMFUTIL check that the SMF SYSTEM ID's in the data stream are identical. This is useful for detecting when data contamination has occurred and when a non-standard SMF SYSTEM ID had been used in a record other than the creating systems SMF ID.
SMFIDCRC	4	Return code to be set when SMF ID CHECKING (SMFIDCHK=YES) detects a change in the SMF System ID in the input data stream. Valid values are 0, 4, 8, 12 and ABEND. A specification of ABEND will result in an abend of the run with a condition code of 999.
SORTTOL	15	Sorted order checking uses this number as the maximum negative deviation in date/time value that may exist between records before an out of sort condition is reported.
SPINBKUP	3	The number of data blocks that High Speed Spin will back up after finding a data block containing a date/time stamp that occurs after the lowest required date/time.
SRBUFF	DYNAMIC	Number of sequential read buffers to be used for each input sequential data set. Valid values are "DYNAMIC" or numeric entries of 0 or 2 to 99. "0" is the same as "DYNAMIC". The default of "DYNAMIC" is highly recommended. It lets the system determine the optimum buffering level.
SUBUFF	DYNAMIC	Number of sequential read buffers to be used for each user defined output sequential data set. Valid values are "DYNAMIC" or numeric entries of 0 or 2 to 99. "0" is the same as "DYNAMIC". The default of "DYNAMIC" is highly recommended. It lets the system determine the optimum buffering level.
SUCLEAR	NO	Controls whether or not SMFUTIL is to use an SMFUTIL type clear block for safety (YES) or a normal IBM type clear block (NO).
SWBUFF	DYNAMIC	Number of sequential read buffers to be used for each standard (SYSUT2/2D) output sequential data set. Valid values are "DYNAMIC" or numeric entries of 0 or 2 to 99. "0" is the same as "DYNAMIC". The default of "DYNAMIC" is highly recommended. It lets the system determine the optimum buffering level.
SYMBOLS	YES	Should SMFUTIL invoke the system utility ASASYMBM (if available) to translate system symbolic terms?
SYNAD	YES	Should SMFUTIL utilize SYNAD exits to intercept I/O errors?
SYSPDSN	*	Dynamic allocation data set name for SYSPRINT output if require (not in JCL). Default of '*' causes a JESx spooled output dataset to be allocated.
SYSPDISP	N	N, M or O for NEW, MOD or OLD as initial status of SYSPRINT dataset.
SYSPSUFx	blank	D, DT or DTX to indicate the type of suffix to attach to the SYSPDSN dataset prefix prior to allocation. Ignored if a relative GDG data set name is specified.
SYSPVOL	blank	VOLSER to place SYSPRINT dataset to.
SYSPUNT	SYSDA	Unit name to allocate SYSPRINT dataset to.
SYSPDCLS	blank	SMS Data Class for SYSPRINT dataset.
SYSPSCLS	blank	SMS Storage Class for SYSPRINT dataset.
SYSPMCLS	blank	SMS Management Class for SYSPRINT dataset.
TERSE	NO	Should record type with zero output counts be omitted from reports?

THSUFFIX	NO	Instructs SMFUTIL that all TIME suffixes generated are to be of the extended format. This will include an additional suffix at the end of the form “.Hth” where ‘th’ are the tenths and hundredths of a second value from the time used. Specifying THSUFFIX=YES will cause a “DT” or “CT” type suffix request to be treated like a “DX” or “CX” request, respectively. All time suffixes generated will be of the extended format.
TMBYPASS	YES	For TLMS shops only. Should SPACE=(1,(1,3)) be used as a special allocation parameter during dynamic allocation of tape data sets during a checkpoint restart? For this to be effective, the TLMS option DBLTIME must be non-zero (DBLTIME=00001 is sufficient) in the TLMSIPO member of the ‘CALPPOPTION’ data set. For tape data sets in the JCL, the SPACE=(1,(1,3)) must be specified by the installation in the JCL.
VCBUFF	42	Number of VSAM output buffers to be used for clear process of an SMF data set.
VRBUFF	34	Number of VSAM input buffers to be used for reading an SMF data set.
VWBUFF	40	Number of VSAM output buffers to be used for writing to a user defined (TYPETODD) VSAM data set.
WEEKPCAL	null	The site defined WEEKPCAL (perpetual week calendar) table to be used for all week splits and data set name week suffixing. If specified, the table must be pre-installed into a LNKLSSTxx data set or a STEPLIB data set.
WKSTART	SAT	What day should be used as the first day of the week for WEEK splits?
WTOLEVEL	5	Above what message level should messages also be issued to the operator console via WTO? If set to 5 (the default) no messages will go to console by default.
XERROR	YES	Should extended error handling (SYNAD exits) be used during record processing.

Most keywords that are set on (or off) in the default module may be negated by the specific inverse form of the keyword. For example ‘XERROR=YES’ in the default module may be reversed at execution time by specifying NOXERROR in the SYSIN input stream. See each keyword for the specific inverse value required.

SPECIAL NOTES FOR MODIFICATIONS:

Setting the ‘ALL’ keyword to ‘NO’ in the SMFUDFLT module will cause all records to be turned off for selection. Each record type desired must then be selected explicitly (via INCLUDE or INONLY) or implicitly (via TYPETODD or SUBSET) for any records to be copied. If none of these keywords is included in the SYSIN input stream, no records will be copied.

Setting the ‘DFTLIST’ keyword to ‘NO’ will remove the default listing from the SYSPRINT report. This can only be permanently changed by changing this setting. This can be temporarily overridden at execution time by a “PARM=DFTLIST” to turn on the listing or “PARM=NODFLTLI” to turn off the listing. There is no SYSIN keyword parameter to suppress the default listing because the listing is produced before the control statement set is processed. The default listing is essential to problem resolution if you call for technical support. If you set this to ‘NO’ and subsequently require support, you must rerun the failing job stream with a “PARM=DLFTLIST” specification in order to have the listing available for the technician.

The values for the VSAM buffers should not be changed in the default module unless substantial testing is done to determine values that result in improved performance in your environment. The preset default values were the result of a substantial number of test executions. The default values arrived at resulted in the lowest CPU utilization to process a 70 cylinder VSAM file. Use the VCBUFF, VRBUFF, and VWBUFF keywords to test new values prior to replacing the shipped defaults.

The ARCHIVDB archive database cluster name is undefined as shipped in the product. Defining the local cluster name in the defaults module will make the database available to all executions of SMFUTIL. The SYSARCH JCL DD statement or the ARCHIVDB control keyword may still be used to override the defaults specification. The cluster name is limited to 35 characters to allow room for the appending of the path names to the cluster name.

10.2 EXECUTION ACCESS CONTROL

The installation may limit access to SMFUTIL via the optional security exit SMFUSECX or via the AUTHUSER macro in the SMFUDFLT module. The SMFUSECX module must be used if the installation has unique definitions of Userid's or locally defined control blocks that contain required information. The AUTHUSER structure is pre-defined in SMFUTIL and is easily defined. It will use the ACEEUSRI (RACF USERID field) if the ACEE control block is present or the JMRUSEID field in the JMR record (SMF common exit parameter list) if the ACEE is not present. If neither one of these fields is valid in your system the job name will be used as the userid, but the AUTHUSER method will may not be effective because SMFUTIL has no internal way of locating the userid it is being executed under. The job name may or may not be adequate for control in your shop. The SMFUSECX exit point is another possible solution.

The AUTHUSER macro is coded in SMFUDFLT by default as follows:

```
AUTHUSER    (-)
```

The '-' field is a single userid mask that allows any userid to execute SMFUTIL. This can be replaced with one or more Userid's or userid masks to represent users who are authorized to execute the product. In a mask, the character '*' (asterisk) is used as a single place holder to indicate that any character in that position is acceptable. Thus 'SYS*TST' would allow any user who's userid is seven characters long and begins with 'SYS' and ends with 'TST' to execute SMFUTIL. A '-' (dash) indicates that any remaining characters in the userid are acceptable. Thus 'SYSP-' would allow any user who's userid begins with 'SYSP' to execute SMFUTIL.

If required, multiple AUTHUSER macros may be specified but they must be contiguous as follows:

```
AUTHUSER    (userid1,userid2)  
AUTHUSER    (userid3,userid4)
```

At least one AUTHUSER statement must appear in the SMFUDFLT module.

Please note that the execution userid is located in one of two control blocks. If an ACEE control block is present, the userid it contains will be used. If the ACEE is absent, the JMR block will be used. If the userid is blanks for any reason, the job name will be used in it's place. In some systems the ACEE will contain an asterisk ("*") as the userid for started tasks that are running under a default userid. In this case, SMFUTIL will ignore the AUTHUSER and AUTHWORD tables and allow full access to all functions. This is done because most installations do not control started tasks anyway. If you wish to limit the access of certain started tasks you must have your security administrator implement full userid control over started tasks so that a valid userid will be available to SMFUTIL.

You may also limit access to the ISPF interface to the SMFUTIL archive database via the ISPFUSRS table that is assembled during the INSTALL job. This table contains AUTHUSER macros similar to SMFUDFLT but with one addition. On the AUTHUSER macro you may code a keyword of TYPE=READONLY for users that are to be able to examine the data but not alter it. For example:

```
AUTHUSER    (userid1,userid2)  
AUTHUSER    (userid3,userid4),TYPE=READONLY
```

would allow users "USERSID1" and "USERID2" to have full display and update authority but Userid's "USERID3" and "USERID4" would be able to display the data but not alter or delete it in any way. No other users would be able to execute the ISPF interface.

10.3 RESTRICTED KEYWORD ACCESS CONTROL

The installation may limit access to critical SMFUTIL keywords via the AUTHWORD and WORDUSER macros in the SMFUDFLT module. Access to the executing userid is required as described above for the AUTHUSER macro.

The AUTHWORD macro is coded in SMFUDFLT by default as follows:

```
AUTHWORD wordtext,ULIST=listname
```

The 'wordtext' field defines an SMFUTIL keyword that is to be protected and ULIST defines the name on an associated WORDUSER macro that defines one or more Userid's that may invoke the specified keyword. Only the base level keyword need be defined, not all alias's of it (i.e. 'DDNAME' but not 'DDN' also).

At least one AUTHWORD macro must appear in the SMFUDFLT module. The 'wordtext' field may be omitted if no keywords are to be protected but the 'listname' field must specify a label on a WORDUSER macro.

The WORDUSER macro is coded as follows:

```
listname WORDUSER (umask1[,umask2,...])
```

Where 'listname' is the name specified in the ULIST parameter on the AUTHWORD macro. The parenthesis contain one or more Userid's or userid masks that define users who are authorized to use the associated keyword. Masking rules apply the same as for the AUTHUSER macro. At least one WORDUSER macro statement must be in the SMFUDFLT module and it must be associated with one or more AUTHWORD macro statements. Subsequent WORDUSER macro lists do not have to be associated with a AUTHWORD statement. This will facilitate changing from one list to another and back as required.

If required, a WORDUSER list may be continued via the following syntax:

```
listname WORDUSER (userid1,userid2)  
WORDUSER (userid3,userid4)
```

All continuation WORDUSER list macros must follow the one it is a continuation of and may not have a label specified. The first WORDUSER macro statement in the SMFUDFLT module must have a label name on it.

As shipped, the AUTHWORD and WORDUSER tables allow full access to all keywords. Several keywords that are prime candidates for control are listed in AUTHWORD macro but the associated WORDUSER macro does not restrict access. It is recommended that some thought be given to the limitation of access to these critical keywords.

10.4 DEFINING SUBSETS

The SUBSET keyword uses pre-defined logical subset names to tell SMFUTIL to copy records of a certain group to the output data set(s). This is a very convenient way to group records together and move them as a class. SMFUTIL comes with several subset names already defined. The user may modify these definitions or add new definitions if so desired. See member SMFSSTBL in the INSTLIB for information on how to code a subset name. Member SMFSSASM may be used to assemble and link edit the new table into the SMFUTIL program. NOTE: Do not remove any subset name in the table as shipped. Some names are used internally by SMFUTIL for record processing.

10.5 MODIFYING THE RECORD ACCOUNTING TABLE

The control section (CSECT) SMFUTLRT is used to track data on individual record types. This CSECT is created by assembling the SMFUTLRT module in the INSTLIB data set. The source code is a series of macro statements. Each defines a record type (or a range of types). These macro entries are used to define what records are eligible for selection or exclusion by job name, userid, subtype, dsname, and/or account code. As shipped, SMFUTIL defines all user records without the job name, userid, subtype, dsname, and account code valid flags. If the user site is producing user record types that contain the user ID, job name, data set name and/or subtypes values, the record type should be generated as such in the table. Comments in the source code will show how to code the macro to indicate that a user record contains the userid and job name in the standard positions or in an identification section that is indexed to via an offset (i.e. a local version of type 30 records using a user record number instead of the type 30 number). Any record type so defined in the record table should also be added to the list for of record types for the subset names that pertain to userid and job name. See the above section for more details. If the record types are added to the subset definitions or are listed in user defined subset names, job name and userid validation will be bypassed unless the appropriate flags are set for the record type in the SMFUTLRT module. See the source code for the module for more information. The JCL stream named RTASMLK may be used to assemble and link the new SMFUTLRT module into the SMFUTIL program.

10.6 SECURITY EXIT – INSTALLATION AND USE

The control section (CSECT) SMFUSECX is a weak external reference that may be replaced with a user defined security module. This exit has complete control over the access to data allowed to the user executing SMFUTIL. This exit is called after completion of control card processing and prior to the processing of data. Upon entry to the user security exit module, register 1 contains the address of a parameter list as follows:

- +0 4 byte address of Job name buffer.
- +4 4 byte address of Userid buffer.
- +8 4 byte address of System ID buffer.
- +12 4 byte address of Record Security Exit Buffer.
- +16 4 byte address of 256 byte record type access mask.
- +20 1 byte of return flags as follows:
 - x'80' - Restrict to specific record types.
 - x'40' - Restrict to specific job names.
 - x'20' - Restrict to specific user id's.
 - x'10' - Restrict to specific system id's.
 - x'08' - Call record exit for each record.
- 3 bytes reserved for future use.

Macro SMFUSEWA in the INSTLIB data set may be used to map this calling parameter list and the buffers pointed to. The first three parameters are addresses of buffer fields. These buffers consist of a two byte length value followed by an eight byte character field (four bytes for SYSID). These buffers are used to restrict access to specific records when the appropriate bits are set on in the flag byte (offset +20 in the parm list). A length value must be provided. If it is less than eight (four for SYSID) the character string will serve as a prefix mask for the specified value. An asterisk (*) may be specified in any position to serve as a wildcard character to indicate that any value is acceptable in that position. Note that the Job name and Userid buffers only restrict access to record types that have job name and userid information capability. Access to all other record types will be allowed unless restricted by record type number in the record mask.

If the Security Exit terminates with a return code of other than zero, the user will be denied the use of SMFUTIL.

The fourth parameter is the address of a Record Security Exit buffer. This buffer consists of a four byte address field followed by an eight byte load module name field. If the x'08' bit is set in the return flag, this buffer must supply either the address or the name of an exit to call for each record. This exit will have final say if the user is allowed access to a specific record. It is recommended that a Record Security Exit not be invoked if at all possible as it is a high overhead expense to call it for each and every record. Most if not all security requirements can be satisfied via the other restriction parameters. If an installation must use such an exit, it is best to provide the address in the parameter list rather than a load module name. A load module name will be 'LOAD'ed by SMFUTIL and a user could easily defeat this by utilizing a STEPLIB or JOBLIB DD statement with a dummy exit of the same name. The most secure method is to imbed the record exit in the SMFUSECX module via a 'VCON' type address reference which will cause the exit to be link edited into the SMFUTIL load module. This address would then be passed to SMFUTIL in the buffer. See below for the calling sequence to a Record Security Exit.

The fifth parameter is an address of a 256 byte record access mask. Each byte position in the mask corresponds to a specific record type of like value (i.e. byte 0 is for record type 0, byte 1 for type 1, etc.). If a byte position has a value of 1, the corresponding record type is available to the user. If a position has a value of 0 (zero) the record is not available and all such records will be bypassed. The record type mask is only considered if the x'80' bit is set on in the flag byte otherwise it is ignored and the user will have access to all record types.

If an improper buffer is returned by SMFUSECX (i.e. Job restriction flag on but Job name length is zero) SMFUTIL will issue a message and terminate with an ABEND U999. A dump will be taken.

If a Record Security Exit address or name is returned from the Security Exit module (SMFUSECX) it will be called for each record processed. At entry to the Record Security Exit, register 1 points to a two word parameter list. Word 1 contains the address of the record being processed. Word 2 is available to the exit for retention of the address of a GETMAIN'ed work area. It is set to zeros the first time the exit is called and left undisturbed thereafter.

A return code of 0 from the Record Security Exit will allow the user access to the record. A return code of 4 will prevent access to the current record and processing will continue with the next record. To the user, it will appear as if the record never existed. A return code of 8 will terminate the current execution of SMFUTIL. The present record will be discarded.

Member SECXASML in the INSTLIB data set may be used to assemble and link edit a security exit into SMFUTIL. There is a sample exit, SMFUSECX, also in the INSTLIB data set that can be used as a shell to design a local exit.

10.7 DEFINING INSTALLATION CALENDARS

The default month and week split functions split data at the beginning of a normal calendar month and at the WKSTART point within a week. SMFUTIL makes provision for installations that use accounting periods of some other structure. Three jobs level tables allow this change. They are called MONTHCAL, MONTHPCAL and WEEKPCAL. Alternatively, the DDAUCAL keyword may be used to specify a specific Calendar for an output DDA block. This calendar is defined as a Month Calendar but in actuality it simply defines a group of periods as calendar periods and an associated suffix for each period. It can be used to specify logical weeks or months.

The MONTHCAL table is a running month calendar with entries that define the first date that constitute a monthly accounting period. Many dates are coded into the future. In use the MONTHCAL table must contain periods below and above all SMF record dates processed. For month splits with data set name suffix generation active, each table entry also contains a suffix to be attached to the new data set name. Note that the MONTHCAL table must be maintained and updated on a periodic basis. The MONTHCAL member of the INSTLIB data set contains an example of an installation month calendar. As many periods as desired may be defined in the table. The only restrictions are that the table must contain at least one date above and one date below the lowest and highest record date processed. The table is assembled and linked (APF=1) into a LNKLSSTxx library (or authorized STEPLIB) data set. Member MCALASML in the INSTLIB data set can be used for this. The 'MONTHCAL(cal-name)' keyword is used to place the month calendar table of load module name 'cal-name' into effect for an execution of SMFUTIL.

The MONTHPCAL and WEEKPCAL tables are perpetual in nature. Each entry describes a Julian day of the year that a calendar period starts on. Once coded, these tables usually do not require updating unless the needs of the installation change. Each entry contains a 3 character suffix to be used when data set name suffixing is required. This user supplied suffix is combined with the year value of the current date for suffix generation. The first character of the suffix must be alphabetic unless YEARFIRST=YES is coded in the calendar module. This allows the supplied 3 character suffix to be all numeric. If this is used, the YEARFIRST parameter must be specified (or set in defaults) so that the supplied suffix will not be at the beginning of the suffix attached to the end of the dsname.

The MONTHPCAL table applies to month splits and suffixing and the WEEKPCAL table applied to week splits and suffixing. Members MNTHPCAL and WEEKPCAL of the INSTLIB data set are examples of these two tables.

NOTE: If both a MONTHCAL and MONTHPCAL table are supplied, the MONTHCAL table is ignored and the MONTHPCAL table is used.

A site may permanently define the use of any of these tables by SMFUTIL via the SMFUDFLT defaults table or PARMLIB defaults member. The MONTHCAL, MNTHPCAL and WEEKPCAL keywords of the SMFUDFLT module, or the same named verbs in the defaults member, can be used to define the name of the calendar load module to be used. At execution initialization the defined module will be loaded into storage. The default defined table may be removed at execution time by specifying a '0' as the required name for the MONTHCAL, MNTHPCAL or WEEKPCAL keywords in the SYSIN data set or it may be overridden to a different calendar table module name.

10.8 AUTOMATING THE GENERATION OF USER CALENDARS

A utility program, SMFUGENC, is included in the DISTOBJ off loaded from the product tape and will be installed by the INSTALL job. This utility can automatically generate a variety of user calendars. Sample execution JCL for the utility is in member GENCAL in the INSTLIB data set. The following is an example of how the utility is executed:

```
//GENCAL    EXEC PGM=SMFUGENC,PARM='2003001,0999,28,P,Y'
//SYSPRINT DD   SYSOUT=*
//SYSUT2    DD   DSN=&&TMP,UNIT=SYSDA,SPACE=(CYL,(1,1)),
//          DCB=(LRECL=80,BLKSIZE=6140,RECFM=FB),DISP=(,PASS)
//*
//ASM       EXEC ASMACL
//C.SYSLIB DD DSN=SMFUTILD.INSTLIB,
//          DISP=SHR,DCB=BLKSIZE=27920
//          DD DSN=SYS1.MACLIB,DISP=SHR
//C.SYSIN DD DSN=&&TMP,DISP=(OLD,PASS)
//L.SYSLMOD DD DSN=target.loadlib(USERCAL),DISP=SHR
```

The input execution parm specifies the type of calendar to be generated and has the following syntax:

PARM='ccyyddd,nnnn,xx,t,s'

WHERE:

ccyyddd – Beginning date of calendar table

nnnn – Number of periods to generate

xx – Number of days in each period

t – Suffix type

- J	- Julian	(SUFFIX = 'Jccyyjjj')
- D	- Date	(SUFFIX = 'Dmmddyy')
- A	- Account	(SUFFIX = 'Ammddyy')
- W	- Week	(SUFFIX = 'Wmmddyy')
- P	- Period	(SUFFIX = 'Pnnccyy')

s – PERIOD SPANNING INDICATOR

- Y	- Year end periods span into new year.
- N	- January 1 begins a new period for each year.

The suffix is generated using the start date of each period.

NOTES:

1. The difference between "SPANNED" AND "NON-SPANNED" periods is whether a new period always begins on January 1 of each year. A non-spanned period always begins on January 1 of a new year. A spanned period is allowed to overlap into a new year until the days-per-period is elapsed and then the first named period of the new year is begun. If the spanned indicator is 'Y', all periods will contain the same number of days. If it is 'N', the final period of each year will be short (if less than the number of days per period remain in the year).

2. Even though they have the same suffix structure, there is a significant difference between Date and Account suffix types. An Account suffix will cause a new **full** period to always begin on the same day of the week as the day of week value of the first date specified to begin the calendar (the one specified in the EXEC parm). For Spanned period calendars, the last Account period of the year will contain 0 to xx-1 days of the following year. The first period of the new year will begin after the previous year's final period is complete. It's start will be on the correct day of the week. For non-spanned period calendars, the last period of the year will begin on the same day of the week as the first date specified (to begin the calendar) and will contain 1 to xx days of data. The first period of the year of a non-spanned calendar will always begin on January 1 and end the day before the first full Account period begins. In this case, the first period will contain 1 to xx-1 days of data depending on how many days there are until the first full account period begins. This structure allows the year to be carved up into accounting periods. The usual duration of such periods is 7 or 28 days. Note that type 'W' (Week) is identical to type 'A'. It's only purpose is to allow the generated suffix to begin with 'W' instead of 'A'.

The load module name specified on the SYSLMOD DD statement should be unique. It is the name to be specified in the MONTHCAL keyword. Note that regardless of the period duration chosen for the calendar, SMFUTIL itself considers this a MONTH calendar and each period is a logical month.

10.9 MODIFYING THE UNIT NAME TABLE

During dynamic allocation of archived data sets for data extraction runs that are not authorized (i.e. not executed from LNKLSxxx or authorized library), SMFUTIL utilizes unit names derived to match the device type that was in use when the SMF archive data set was created. There is a separate table for tape and disk unit names. The member SMFUUNIT in the INSTLIB data set contains the definitions as shipped. It uses macro SMFUNITS (also in INSTLIB) to define the required tables. The site may modify or extend the table as required. Member UNITASML in INSTLIB can be used to reassemble and link the table with SMFUTIL proper.

The 'NAMES' parameter of the SMFUNITS macro is used to define the unit names to be used for the existing device types for disk and tape. At least one invocation of the macro for DISK and one for TAPE is required. As shipped, if an unknown device is detected, 'TAPE' will be used for tape devices and 'SYSALLDA' will be used for disk devices. The installation may modify or extend the unit names of both tape and disk if desired. The 'NAMES' parameter consist of pairs of eight digit hex device types and a one to eight character unit name to be used for that device type within class. The device type specification may contain asterisks (*) in any position to indicate that any hex character in that position is to be considered acceptable for this unit name. For example:

```
SMFUNITS      DISK,NAMES=( ****200E,3380,****200F,3390)
```

will define '3380' as the unit name to be used for disk devices of class x'0E' and '3390' as the unit name to be used for disk devices of class x'0F'. Generic device names (as shown above) may be used or esoteric names (i.e. 'SYSDA') may be used.

Caution should be used when defining tape devices. If SMFUTIL will be creating output SMF data sets on tape devices with data compaction or other special feature (i.e. 3480X), ensure that the TAPE unit names parameter is properly coded to cause recalled data sets to be allocated to a device that will be able to read them.

10.10 ESOTERIC TAPE UNIT NAME IDENTIFICATION

During dynamic allocation of output datasets, SMFUTIL must determine if the output device is tape or disk. In a authorized environment this is accomplished by extracting a candidate sample UCB device number for the unit name specified and examining the UCB Common Segment Device Type. This cannot be done in an unauthorized execution so SMFUTIL utilizes a table of tape unit names to indicate an allocation is targeted for a tape device. The SMFUUNIT source module contains an SMFTAPES macro that defines a list of tape unit names for the installation. The user may add to this list any local unit names that are tape devices and rerun the INSTALL job.

will define '3380' as the unit name to be used for disk devices of class x'0E' and '3390' as the unit name to be used for disk devices of class x'0F'. Generic device names (as shown above) may be used or esoteric names (i.e. 'SYSDA') may be used.

11. SMF DATA COLLECTION METHODOLOGIES

11.1 THE COLLECTION AND ARCHIVING PROCESS

Collecting and maintaining SMF data can be one of the most difficult operational tasks faced by a data center. The amount of resources and manpower dedicated to this task is directly proportional to the importance that SMF has in your data center. Many shops, that have a critical need to preserve SMF data at all costs, will duplicate SMF data as soon as it is dumped. On the other end of the scale, some shops, without such a perceived critical need, will MOD onto the same tape all month long and change around the first of the month if someone remembers to reset the GDG number in the JCL.

While SMF data should be safeguarded as much as practical, it is possible to minimize the resources dedicated to the task by utilizing SMFUTIL. SMFUTIL allows data to be accumulated onto tape in a variety of ways. By the day, week, month, or some date range specified by the user. There are two basic approaches to performing the tasks of dumping and clearing the SMF data sets. The old, standard way is to dump each data sets as it is marked "DUMP REQUIRED" and then clear it for reuse. The IEFU29 exit point can be used to automatically start of job that will dump and clear a data set when it is marked "DUMP REQUIRED". With SMFUTIL this is a two step process. The first copies the data to the archives (dumps it) and the second step clears the data set for reuse (CLEAR function).

Alternatively the DUMPCLEARALL function of SMFUTIL can be used to locate, allocate, dump and clear all SMF data sets that contain data. In addition, it can automatically leave the primary data set as "AVAILABLE" to prevent the data ordering problems that can be caused by "Z EOD".

11.1.1 SINGLE SMF DATA SET DUMPING

When each SMF data set is to be dumped individually, it is most efficient to use two SMF data sets, with each large enough to hold at least a full day's worth of SMF data. A third, smaller, data set should be allocated for emergencies, but it will not normally be used. SMF will rotate between the Primary and First Alternate as long as they are available. The JCL procedure described here is contained in member SMFUDUMP in the INSTLIB. If it is not possible to allocate this much space to SMF files, smaller ones may be used. They will be dumped more frequently but the SMFUDUMP procedure can be run as often as required with no problems. An automatic or manual command could be used to invoke a switch of the SMF data sets at an appropriate time (early AM hours). The IEFU29 SMF exit should be used to automatically start the SMFUDUMP procedure to perform the SMFUTIL copy and clear task as outlined. The sample IEFU29, included in the INSTLIB member IEFU29S, will perform this task. This procedure will copy the just completed data set and clear it for reuse. The MOD disposition can be safely used because of the checkpoint protection (via SYSCKPT) against duplicate data in the event of system failure. In addition, it is useful to create two monthly 'month to date' masters at one time. This is accomplished by dynamically allocating the output data sets SYSUT2 and SYSUT2D and specifying the SPLIT sub-parameter on the DDAEND command. This will result in each month being totally contained on one physical data set. Full utilization of each tape volume is achieved yet easy access to the data at the volume level is maintained via the SMFUTIL ARCHIVE.

An optional second step could be added and triggered only when the return code from the dump step was 3. This return code would mean that the dump step had encountered a month boundary and begun a new monthly data set. The previous months data set could now be copied to another data set name and sent to off site storage for data protection purposes.

While an automatic switch of SMF data sets due to a full condition will cause the SMF EXIT IEFU29 to automatically kick off a dump job, it is recommended that a switch be performed manually (or by automatic command) each night after the batch workload is completed (after midnight). This will ensure that any split that occurs due to a month end condition will happen on an off shift dump run. If a second step is to copy an entire month of data (the one just completed) to one or more archive copies, it is better to do it on off shift rather than during prime shift while the machine is at it's greatest load.

NOTE: Extreme caution should be used when MODing onto an existing SMF tape data set if the SYSCKPT data set is not supplied. If a machine or program failure occurs during the run, the output data set will contain part of the data to be copied. When the operation is repeated, the data already copied during the first run will be duplicated on the second run. If this problem has already occurred, the DELDUPRV function can be used to eliminate such blocks of duplicate data while copying the data to a new output data set.

During a copy step from an SMF data set, it is a good practice to include the COPY00K keyword. This prevents a return code of 8 from aborting the run when an empty SMF data set is inadvertently dumped and cleared.

11.1.2 MULTIPLE DATA SET DUMPING

The DUMPCLEARALL function gives SMFUTIL the ability to dump and clear one or more SMF data sets in a single step. Based on the MANXALOC parameter, it goes further by automatically switching SMF data sets and clearing them after the dump process is complete.

If your installation is currently using the old IBM IPO supplied utility SMFDUMP to invoke the IFASMFDP program to dump all SMF files that contain data, this process can be replaced with SMFUTIL and the DUMPCLEARALL function.

The INSTLIB member IEFU29 contains the source code for the IEFU29 exit that should be used when the DUMPCLEARALL process is to be used. It is functionally identical to the IEFU29S exit source except that it omits the 'SMFDSN=dsname' parm from the start command issued. If you already have the other IEFU29 in place you can still switch to DUMPCLEARALL by changing the SMFUDUMP PROC as follows:

1. Change the SYSUT1 DDNAME in the COPY step to SMFUT1.
2. Remove the CLEAR step.
3. Add the DUMPCLEARALL parameter statement to the SYSIN input data member used in the COPY step.

As soon as practical, replace the IEFU29 with the new one and remove all references to SMFDSN in the SMFUDUMP PROC.

11.1.2.1 DUMPCLEARALL PROCESSING

The DUMPCLEARALL function of SMFUTIL greatly simplifies the task of dumping and clearing SMF data sets. The DUMPCLEARALL function is 'intelligent' about the current SMF data set structure and which data sets contain data. It automatically allocates SMF data sets for input in the order in which they contain data (oldest data first) so data is dumped in the order it was created. In addition, (if MAN1LEAV is NOT specified) if at least one SMF data set is currently marked 'AVAILABLE' (able to accept a switch), an 'I SMF' command is issued to perform a switch so all data currently available can be dumped.

DUMPCLEARALL operates in three phases: 1) Input selection, 2) Dumping, 3) Clearing. Note that all candidate MANx data sets are determined during phase 1. Dumping will not begin until all switching decisions have been made and performed. Likewise, phase 3 does not begin until all dumping has been successfully completed.

During DUMPCLEARALL processing, a special DUMPCLEARALL checkpoint data set is allocated to DDNAME SYSDCKPT. The data set name used is defined by the "DCACKPT" parameter in the SMFUDFLT module. Parameter "DCAUNIT" defines the unit name to be used to create it and DCAVSER. This data set must be allocated by the DUMPCLEARALL run or it will not proceed. The default data set name is 'SYS1.SMFUTIL.DMPCLALL.SYSsid' where 'sid' is the four character SMF system identifier of the operating system. Note that this data set would be cataloged in the master system catalog. A user cataloged data set name can be used but it must be unique to the system and not confused with another system. In addition, any security system present must allow the SMFUTIL run full access to the data set name specified. The DCACHECKPOINT keyword may be used to override the specifications in the SMFUDFLT module and cause a different data set name prefix, unit, or volser to be used.

In addition to the SYSDCKPT DUMPCLEARALL checkpoint data set, the normal SYSDCKPT data set must be available in the JCL or via the CKPTDB parameter. A special SYSDCKPT record, \$DUMPALL, is used to indicate a DUMPCLEARALL process is underway. In the event of a failure, the \$DUMPALL record and the SYSDCKPT data set are used to correctly restart the DUMPCLEARALL process. Note that full MOD output protection is still functional during DUMPCLEARALL. This is because all dumping tasks are performed first. After all output data sets are successfully closed the ARCHIVE database (if used) is updated. Only then are the SMF data sets cleared. This prevents any SMF data from being at risk until it has been successfully written to the output data sets and recorded in the archive database.

During the CLEAR phase the MULTICLEAR option determines how datasets are cleared. If MULTICLEAR is set to YES in the defaults module or specified at execution time, SMFUTIL will perform the clears asynchronously. This is more efficient and returns MANx datasets back to SMF quicker.

During CLEAR processing, SMFUTIL automatically writes a dummy holding record as the first block in the MANx dataset and then clears the required remainder (SHORT or FULL) of the dataset. After the entire clear on the MANx dataset is completed the first dummy holding record is overwritten with a clear record indicator. This process protects the integrity of the clearing process in the event of a system outage. A restart will automatically detect the previous incomplete CLEAR and resume where it left off.

NOTE: If an operator "T SMF=xx" set command is used to place a new set of SMF MANx data sets into use an IEFU29 exit will be triggered and a DUMPCLEARALL job/stc submitted. This job should NOT be allowed to run because it will attempt to dump and clear the NEW set of data sets (the ones now in the control blocks) and the previous set's data would not have been dumped. To simplify this situation, before doing the set command perform an "I SMF" for force a switch and allow the current data to be dumped and cleared. Note which data set is currently active prior to doing the SET command. Note that the amount of data in the previously active will be small and many shops may wish to just discard it as irrelevant. If you must retain the data, perform the following functions. After the SET command completes, but prevent the dump job from executing. An easy way to prevent it from running is by altering the JCL procedure to force a JCL error when it is submitted. Manually dump the previously active MANx data set using the same output data sets and structures the production DUMPCLEARALL job uses. Make sure to use an empty SYSDCKPT PDS data set if you are doing MOD to output data sets but do not use the same one that is used in the production run. After this is complete, correct the standard JCL procedure to work again and perform an "I SMF" for force a switch and start a standard DUMPCLEARALL execution.

11.1.2.2 MANILEAVE OPTION

NOTE: The MANILEAVE option is no longer the recommended method of preventing out of order data dumping that can be caused by a “Z EOD” operator command. See the section titled “SAFE CLEAR PROCESSING” on page 220 for more information.

Optionally, if the MANILEAV parameter in the SMFUDFLT module is set to ‘YES’, DUMPCLEARALL will always attempt to leave the primary SMF data set as available at the end of the run. This is to allow a “Z EOD” command to switch to the primary data set. The subsequent IPL will continue adding data to the primary data set, preventing the “Z EOD” command from leaving a small amount of data in the first alternate data set after the EOD switch, and then starting back with the primary after the IPL and ‘orphaning’ the data in the alternate as out of sequence.

Note that when MANILEAV is specified, it will not do a switch unless it needs to do so as to satisfy the MANILEAV request. This may be changed by also specifying ISMF. This tells SMFUTIL to always do a at least one switch if it can do so and still honor the MANILEAV specification.

If DUMPCLEARALL with MANILEAV approach is used, the primary SMF data set should be relatively small to reduce the time it takes to clear it, and the alternate data sets (at least two) should be rather large. In normal operation, if the primary data set and at least one other is available, DUMPCLEARALL will issue two switch commands. The first will cause the current data set to be closed and will switch to the primary. The second switch will close the primary and switch to the other alternate.

If the primary is currently active and at least one other data set is available, only one switch will be issued. All data sets having data, including the primary, are then dumped and cleared.

This process will always leave the primary data set available. The only exception would be if the primary was active and all other data sets were marked dump required. This can be prevented by using the IEFU29 exit to automate the start of the DUMPCLEARALL run.

11.1.2.3 SMF DATA SET SWITCHING

The DUMPCLEARALL process requires that input data be supplied or it will terminate with a return code of 8. To ensure data is available and to ensure that all current data is dumped, SMFUTIL can cause SMF to switch from the currently active MANx data set to a new one causing the current data set to be marked “DUMP REQUIRED”. Two keywords are involved in how and when SMFUTIL will perform a switch command. ‘ISMF’ instructs SMFUTIL to always perform at least one switch and MANILEAVE instructs SMFUTIL to leave the primary MANx as ‘AVAILABLE’ and the end of the DUMPCLEARALL run. Note that if the primary data set is currently marked ‘AVAILABLE’, MANILEAVE will cause two switch commands to be issued. The first causes the primary to become the ACTIVE data set and the second causes it to be marked ‘DUMP REQUIRED’. Note that if the primary is the only ‘AVAILABLE’ data set or if all data sets are ‘DUMP REQUIRED’ and SMF is buffering data, and MANILEAVE is specified, no switch will be performed as it would cause the primary to become ‘ACTIVE’ thus negating the MANILEAVE request. In the situation where the primary is active and all other data sets are marked ‘DUMP REQUIRED’ no switch can be performed because no data set is available to accept the switch request. The MANILEAVE parameter will be ignored and an informational message will be issued. MANILEAVE will be honored on the next DUMPCLEARALL run.

Both MANILEAVE and ISMF may be set as defaults in the SMFUDFLT module. See the section entitled “MODIFICATION AND CUSTOMIZATION” on page 199 for more information on the keyword definitions in the defaults module and their usage.

NOTE: If SUCLEAR=YES is defaulted or SUCLEAR specified at execution time, the MANILEAVE option is not required and should be turned off. SMF will not reuse the primary dataset until SMFUTIL has properly cleared it and notified SMF of it’s availability.

11.2 CLEARING SMF DATA SETS

In a single data set approach, SMFUTIL requires the user to clear data from SMF data set in a separate step from that in which the data was copied to the archive data set. This forced separation of dump and clear functions is to ensure that copied data is retained via proper catalog disposition after completion of the copy step.

If the DUMPCLEARALL function is used, SMFUTIL will dump and clear all SMF data sets having data in a single step. This capability is dependent upon all input and output data sets being dynamically allocated. This gives SMFUTIL full control over ensuring that all output data sets are successfully created and cataloged prior to clearing the SMF data sets.

The CLEAR function (via CLEAR parameter or DUMPCLEARALL invoked) will automatically return the data set to the SMF system if it is operated in an authorized environment (i.e. an authorized library or LNKLIST data set). If not authorized the data set will be cleared and SMF will not be POSTED by SMFUTIL. SMF may or may not 'discover' the cleared data set and return it to use. APF authorization of SMFUTIL is essential for proper CLEAR operation and is REQUIRED for DUMPCLEARALL usage.

There may also be a problem if the data set is only partially cleared due to a system failure or operator cancel. During IPL, SMF detects a cleared data set by reading the first data block and checking for a software end of file indicator. If detected, the data set is assumed empty and returned for use. If a clear step did not complete, additional data may be left in the data set. If SMF overwrites the entire file this will not cause a problem. If it writes into the old data but not completely to end of file, the old data will be dumped again (causing duplicate data on the output archive files). This problem can be prevented by ensuring that all clear steps are completed successfully via an operational procedure to verify the output of the CLEAR function. It is also a good policy to allow SMF to completely fill each data set and switch over to an empty data set. The same SYSCKPT data set should be used to protect each copy and CLEAR function step. This will allow SMFUTIL to detect when a CLEAR function has not been completed. If message SMFU163 is produced by a copy step, it indicates that a CLEAR function did not complete in a previous run. Before proceeding, ensure that: 1) the same SMF data set is being processed, 2) the SMF system has not added additional data to the data set since the failed CLEAR execution (check SYSLOG). If no data has been added and it is still marked "DUMP REQUIRED", it is safe to clear the data set. If the data set is marked "AVAILABLE" it has been re-opened by SMF and CLEAR will not function as the data set cannot be opened for output by SMFUTIL. If the data set is now active data has been added to it. The data set should be switched off of and dumped to a separate data set on disk. SMFUTIL can then be used to copy only the new data to the archive data set. The DELDUPRV function should make this quite easy as it will discard data at the rear of the file that is out of order (reversed time sequence). However, this situation should never occur, if proper operational procedures are observed.

NOTE: An MVS/XA or MVS/ESA SMF data set should never be cleared by any program (including SMFUTIL and IFASMFDP) running on a NON-XA (MVS/SP or MVS/370) operating system. XA/ESA and NON-XA SMF data set have different clear structures.

NOTE: It is extremely important that the CLEAR function execution never be canceled or allowed to ABEND for lack of CPU time (system 322 ABEND). Such an interruption could cause logical damage to the SMF data set. An IDCAMS verify may or may not be able to recover the data set. It may take more than one verify to recover the data set (up to eight). Once a clear function starts, it is too late to recover any data from the data set being cleared, so canceling the clear will not save any data.

11.2.1 SAFE CLEAR PROCESSING

The “SAFE CLEAR” functionality of SMFUTIL targets the habit of the SMF system of reusing MANx datasets at IPL. This can cause out of order data if an IPL is done after a “Z EOD” operator command. The halt command causes SMF to quit recording on the current dataset and begin recording on a new dataset. If the current one is the primary (i.e. MAN1) the problem will occur. After IPL, SMF goes back to recording on the primary, leaving a small amount of data “abandoned” in another MANx dataset cause when the halt command was issued. During the IPL, this dataset is marked “DUMP REQUIRED” by SMF and a dump process is started. Even if a switch occurs the data is still not in the correct order because MAN1 was used twice with the “abandoned” data being form it’s middle. SMFUTIL can prevent this problem by utilizing two methods.

The first is automatic and requires no action by the user. During CLEAR processing SMFUTIL will write a special indicator record as the first record of the dataset prior to completing the remainder of the clear. This indicator record prevents SMFUTIL from placing the MANx dataset back into service as “AVAILABLE” in the event of a system outage and subsequent IPL during the CLEAR process when the CLEAR was only partially completed. The dataset will remain “DUMP REQUIRED” and an SMFUTIL restart will be performed and the CLEAR subsequently completed.

In addition, if the installation chooses, **SUCLEAR=YES** may be specified in the defaults module or SUCLEAR specified at execution time. This will cause the same special indicator record will be used to clear the entire dataset and only the first record will be overwritten with an IBM clear indicator record. This option prevents SMF from “MOD’ing on” to a MANx dataset (i.e. reusing MAN1 as above) during the IPL process. During IPL, only datasets that are actually CLEAR will look clear to SMF initialization. All others will look full. This prevents SMF from creating out or order data by reusing a partially used dataset before it can be dumped and cleared.

The “special” record written by SMFUTIL is actually an invalid SMF record with binary zero for the time, packed zero for the date and a blank SID field. If IFASMFDP is used it will discard these records as invalid. If you have other OEM type software that directly reads the MANx datasets, please call SMFUTIL Technical Support for instructions and assistance with the product interface.

NOTE: If the SUCLEAR function is implemented, the MANILEAVE option may be omitted as SMF will no longer be able to MOD onto MANx dataset at IPL thus eliminating the problem of out of sequence data caused by “Z EOD” processing followed by an IPL.

NOTE: If the SUCLEAR function is implemented, it is recommended that NOSHORTCLEAR be specified for the initial executions until all MANx datasets have been cleared at least once. This will ensure that the SMFUTIL special clear record is used to completely clear each MANx dataset. After all datasets have been cleared at least once the SHORTCLEAR option may be re-implemented.

NOTE: If the “SMFUTIL CLEAR RECORD” method is used (SUCLEAR=YES is defaulted or SUCLEAR specified at execution time), the MANILEAVE option is not required and should be turned off. SMF will not reuse the primary dataset until SMFUTIL has properly cleared it and notified SMF of it’s availability.

11.2.2 SHORTCLEAR THRESHOLD PROCESSING

SMF MANx dataset can be cleared using a sequential or an indexed approach. The sequential approach is more efficient (faster) but using it requires the entire dataset to be cleared because of the IBM design of the SMF system. The HIGH USED RBA of the MANx cluster must always equal the HIGH ALLOCATED RBA value. This means the MANx cluster must always appear to be “full” of data. The indexed approach is less efficient (slower per record) but the results are faster if something less than a full dataset needs to be cleared. During the DUMPCLEARALL process, SMFUTIL keeps track of how many blocks on the MANx dataset being dumped contained data (input block count). This allows SMFUTIL to “decide” if a full clear or a short clear should be used. The SCLRTHLD value is used to make this decision if SHORTCLEAR is in effect. A value of 0 to 100 is specified which defines the percentage full above which a FULLCLEAR is to be utilized instead of a SHORTCLEAR. A value of 0 means a FULLCLEAR will always be done. A value of 100 means a SHORTCLEAR will always be done. The default value as shipped is 80 percent. The benefit of a SHORTCLEAR over a FULLCLEAR are dependant on many factors including the actual device being used, the availability of caching, and the overall environment (channels, controllers, paths, etc..). Some experimenting with this value may result in reduced CLEAR times on average but the default value is adequate in most installations.

11.3 DUMPING FREQUENCY

SMF data should not be allowed to sit in full or partially full inactive SMF data set. As soon as a “DUMP REQUIRED” message appears on the operators console, the data set should be dumped and cleared. After an IPL, the previously active data set will require dumping. It should be dumped and cleared immediately. There should always be at least one empty SMF data set on the system. If a system outage occurs while all inactive SMF data set have data in them, upon IPL SMF will begin placing data in the first data set that has space left. This may or may not be the data set that was last active. This situation can cause data to be added to the data set out of order.

The size of the SMF data sets will be somewhat dependent on the method chosen for dumping and clearing them. If one data set is to be dumped at a time, it is most efficient, time wise, to have two SMF data set the same size and a third, smaller one, to be used only if required. Ideally, the primary and first alternate should each have enough space to hold an entire day's worth of data. This will allow the dumping and clearing to happen at off hours (early AM) during low system usage periods. It also enables the primary and first alternate to be used round-robin and reduces the dumping frequency to once a day. SMF will always go back to the primary data set if it is available. The second alternate (the small one) will be used only if one of the others is damaged and dropped from SMF. In addition, this allows all of the SMF data set to be on one pack. If dumping is occurring frequently during high system usage times, the accessing of the data on the data set being dumped may impact the writing of SMF data to the active data set, if they are both on the same device.

If the DUMPCLEARALL function is to be used to dump multiple data sets and the MAN1LEAV parameter of the defaults modules is set to ‘YES’, it is better to have the primary data set be relatively small so that it can be dumped and cleared quickly. During DUMPCLEARALL processing SMFUTIL will issue a switch command (I SMF) to cause SMF to switch off of the current data set. If the primary is available, SMF will always switch back to it. The MAN1LEAV=YES specification tells SMFUTIL to issue a second switch immediately so the primary can be dumped and cleared in this execution thus leaving it available at all times. This is of great benefit if the ‘Z EOD’ command is issued at system shutdown. This command causes the current SMF data set to be closed and a switch to be performed to another data set. A small amount of data will be written to the new data set. If the data set switched off of was the primary, during the subsequent IPL the SMF system will reopen the primary and add data to it again, leaving the data in the other alternate as out of sequence. The MAN1LEAV=YES specification prevents this as the primary data set is always available. The ‘Z EOD’ command causes the primary to begin to receive data and the IPL will also begin with the primary. The IEFU29 exit (if in place) will cause an immediate dump of the alternate that has data (the one switched off of by the ‘Z EOD’) and SMFUTIL will switch off of the primary and dump it and clear it leaving it available again.

12. USER EXITS

SMFUTIL supports two exit structures. The first are defined SMFUTIL exit points. These exit points are much more capable and flexible. Multiple exits may be defined for each point and each has more control over the execution.

The second type of exit point are compatibility exits. They are designed to mimic the exit points defined for IFASMFDP allowing the user to utilize OEM exit modules designed to be run under IFASMFDP.

12.1 SMFUTIL DEFINED EXIT POINTS

SMFUTIL allows the installation to have complete control over and access to all records being processed through the implementation of several types of exits. Each exit point is activated by specification of a keyword that supplies one or more load module exit names. The exit points and their activating keywords are:

- INTEXIT** An initialization exit point to be used to initialize an environment for user exits prior to the start of data processing. Called only once during program initialization.
- PREEXIT** An exit point after record construction completion but before record validation. Any changes the user makes will appear to SMFUTIL as if there were already present when the record was read.
- RVALEXIT** An exit point after SMFUTIL has validated the record date and time values but before any user filtering criteria are processed.
- EXIT** A general exit point that receives control for each record that has passed data validity and user filtering specifications and prior to SMFUTIL passing the data set to the output data set.
- RECEXIT** A record exit point that receives control only when a specific type of record is encountered.
- POSTEXIT** A post processing exit that receives control after a record has completed SMFUTIL processing. All records that the user is authorized for are passed to the POSTEXIT, even ones that failed filtering and selection criteria. The POSTEXIT may insert additional data records into the input stream for processing by SMFUTIL by overlaying the record buffer supplied and issuing the required return code.
- TERMEXIT** A termination exit receives control when all input data has been processed. It may be used to clean up the exit environment and perform any final totaling or reporting.

Upon entry to a user exit module, register 1 points to a four word parameter list:

- +0 Address of the record being processed.
- +4 A full word available to this specific user exit.
- +8 A full word available to all user exits. The contents returned by one exit is passed to the next exit and retained throughout the run.
- +12 The address of the standard output DCB in use by SMFUTIL for writing output data (i.e. SYSUT2/SYSUT3/etc.). Note that in certain exit points the DCB may not be open as yet if SMFUTIL has not written any output data to it prior to calling the exit.

Member UEWAPARM in the INSTLIB data set contains a mapping macro for the parm list.

Word one will contain the address of the record being passed. After all data has been processed, each exit is called one last final time for cleanup purposes. This is indicated by passing x'FFFFFFFF' as the record address. This final close out call is made to each exit even if the run was terminated by an exit (via return code) or the exit was "Killed" (via a return code). This word will contain zeros for an initialization or termination exit.

Word two is initialized to zeros the first time an exit is invoked and is left undisturbed thereafter. This word may be used to store the address of a user work area that is GETMAIN'ed and specific to this exit only. The work area will then be available to this exit for the duration of the SMFUTIL run. Note that any initialization and termination exits are only called once.

Word three is a full word that is available and common to all user exits specified. Word three is initialized to zeros prior to calling the first user exit, including the initialization exit, if present. Its contents are duplicated as each exit terminates and are passed to the next exit taken. This word may be used to store the address of a user work area common to all exits or to pass flags between exits.

Word four is the address of the standard output data set currently in use by SMFUTIL. This will be SYSUT2 or one of its split data set if split is in process. Note that the DCB address will be 0 on the first call to an exit as the data set has not been opened. SMFUTIL does not normally open output data sets until they are actually required. Output to this DCB should be on a delayed basis. Each call to an exit can be used to flush out data created in the previous call first and then create any new data for this pass. The termination call can be used to flush out any final data. No duplexing of the data will be performed. Note that the POSTEXIT is preferable for inserting data into the output files as it causes SMFUTIL to fully support the new data record with complete data validation, split, and duplexing functions.

The INITEXIT keyword is used to specify the name of an initialization exit load module. The TERMEXIT is used to specify the name of a termination exit load module. The RECEXIT keyword allows one or more user exits to be invoked by SMFUTIL prior to final disposition of each record of a particular type that will be written. Please note that any RECEXIT's for a particular record type are called before any general (all record) exits are called for the record. The EXIT keyword allows one or more user exits to be invoked by SMFUTIL prior to final disposition of each record that will be written. The POSTEXIT keyword is used to specify the name of a post processing exit load module that is to receive control after each record has completed processing. The POSTEXIT is unique as it is the only exit that may insert data into the input data stream for SMFUTIL to process itself (rather than the exit writing it directly to the output data set).

All user exits must be load modules available on STEPLIB or JOBLIB or placed in a LNKSTxx data set. Because SMFUTIL usually runs in an authorized state (i.e. during DUMPCLEARALL), they should be link edited with an authorization code of 1 (APF=1).

Each record passed to a user exit is fully assembled (not segments) and has met all selection criteria that SMFUTIL is invoked with (i.e. DATE, TIME, INCLUDE, etc.).

Upon return to SMFUTIL from a regular or record exit, register 15 is tested for a return code. The following return codes are valid from regular and record exits:

- 0 Retain this record and pass it to the next user exit.
- 1 Retain this record but call no further user exits of this type (record or general) for it.
- 2 Retain this record. Do not call this exit again (except for final close out call).
- 3 Retain this record. Do not call any user exits of this type (record or general) again during this run (except for final close out calls).
- 4 Discard this record (no further user exits are called for it) Continue with next record.
- 5 Drop this record. Do not call this exit again (except for final close out call).
- 6 Drop this record. Do not call any user record exits again during this run for this record type (except for final close out call).
- 7 Retain this record and terminate this SMFUTIL execution.
- 8 Discard this record and terminate this SMFUTIL execution.

Upon return to SMFUTIL from a POSTEXIT, register 15 is tested for a return code. The following return codes are valid a from post processing exit:

- 0 Continue normally.
- 1 Insert returned record into processing (write to output file).
- 2 Delete the post exit (do not call again).
- 8 Or greater, terminate execution and save this return code as run return code.

Note that a return code of 1 from the POST exit causes the record buffer to be processed again by SMFUTIL. This allows the POST exit to insert new or modified records into the data stream. The new record is handled by SMFUTIL as if it were read from the input data set. It must meet all filtering criteria for the execution in process or it will be discarded by SMFUTIL as any other input record would be.

When terminated by a user exit (i.e. Exit return code of 7 or 8), SMFUTIL will issue a minimum return code of 4. The final return code may be 4 to 7 depending upon whether any data splitting function was invoked. All data processed to that point is retained in the output files and normal data set disposition will take place.

When all input records are exhausted, SMFUTIL will call each regular, record, and post exit one final time, for clean up purposes. This final call is indicated by high values (X'FFFFFFFF') in the first parameter word. It is the responsibility of the each user exit to close any files it has opened and free any storage it has obtained. This final call is made to the user exit even if the run is being terminated by a return code request from a user exit or one or more (or all) user exits have been canceled. Any return code from the user exit for the final call is ignored.

If a TERMEXIT is provided, it will be called after all record and general exits have been called for final termination. This exit may then clean up any remaining work areas left by the user exit environment.

NOTE:

The user exit points are under the protection of the normal SMFUTIL ESTAE recovery routine. If a user exit ABENDs a recovery will not be attempted. SMFUTIL will attempt to print a snap out the location and register contents during the ABEND but this may or may not be accurate depending on the cause and location of the ABEND. The execution will then abort with a U999 ABEND. If additional diagnostic information is required, the exit developer should specify ESTAE OFF during exit testing. This will disable the SMFUTIL ESTAE environment and allow a full SYSUDUMP or SYSABEND formatted dump to be produced when a user exit ABENDs. Caution should be exercised in coding user exits to avoid problems. They should be thoroughly tested prior to being placed into production use.

During SMFUTIL execution, a SPIE environment is in effect to detect invalid record date fields. If a user exit ABENDs with a data exception (S0C7), the current record will be discarded and processing will continue with the next record. During development of user exits it might be beneficial to specify the SPIE OFF keyword to turn off the SPIE environment. If an S0C7 occurs in a user exit with SPIE OFF specified, SMFUTIL will terminate with the ABEND and a dump will be available if the proper SYSUDUMP or SYSABEND DD card was provided. All testing should be done with input data that is known to be error free.

A regular and record user exit is free to do anything to the record in process it wishes to up to and including replacing it entirely. The buffer containing the record is 32,767 bytes in length. A new record could be built in a user exit work area and moved to the SMFUTIL buffer via an MVCL instruction. The returned record will not be validated again by SMFUTIL, so it is the responsibility of the user exit to ensure that any changes it makes does not damage the logical or physical structure of the record and that any replacement records are valid.

A post exit may overlay the record buffer with a new record as outlined above for regular and record exits and then issue a return code of 1. This tells SMFUTIL to process the return record before reading another record from the current input data set. This has the advantage of giving full validation and duplexing facilities to the new record and is the recommended way of processing inserted data. Note that to insure the correct ordering of the resulting output data for subsequent processing, the date and time stamp of the created record should be identical to those in the record presented to the exit (the record just completing processing). This is most easily done by not overlaying the SMF header on the record but only altering the record type field.

An example exit is included in INSTLIB under the name RTE83EXT. It is written to run either as an IEFU83 SMF exit or an SMFUTIL exit. It is specifically for IDMS shops that have the IDMS/R product producing RTE data to the SMF data set. Transaction counts and CPU times from the RTE records for each known IDMS CV are accumulated as the RTE records go by. When a type 72 record (IBM RMF or BOOLE & BABBAGE CMF) record is encountered that matches the started task name for one of the CV's the totals for that CV are inserted in the 72 record TTX and TTM fields. This allows transaction response times to be automatically available to performance reporting and analytical modeling processing that uses the SMF data. Specific examples would be the Boole & Babbage Performance DATA Base (PDB) and CMF/Model products. Transaction counts and timings will be inserted automatically without the manual intervention previously required to collect and input this data. Comments in the source code for this exit will explain how to implement it.

12.2 COMPATIBILITY SUPPORT FOR IFASMFDP TYPE EXIT MODULES

SMFUTIL supports usage of modules designed to be invoked as exits under IFASMFDP. Each exit point is activated by specification of a keyword that supplies a load module exit name. The exit points and their activating keywords are:

USER1 Record read exit. Called after each record is read from an input data set but before the record is qualified as a selected record.

USER2 Record write exit. Called for each record to written to an output data set.

USER3 Data set closed exit. Called each time an output data set has been completed and closed.

These exit points are designed to emulate the corresponding exit point in the IFASMFDP utility program supplied by IBM.

When called, the exit receives a parmlist address in register 1. This parmlist is 3 words in length. For all exit points, word 1 points to a 3 word work area for use by the exit. This work area is common to all three exit points. For exit point 2 and three, word 2 of the parm list points to the current SMF record that was just read in (exit 1) or is about to be written out (exit 2). For exit 3, word 2 points to the DCB that was just completed and closed. Word 3 always points to a ddname. The input DD for exit 1 and the output dd for exits 2 and 3.

A return code of 0 (zero) means continue normal processing. From exit 2 and 3 a return code of 4 means skip this record and continue. For exits 1 and 2 greater than 4 and for exit 3 greater than 0 means the exit had a problem and should not be called again.. A messages is issued, calling of the exit is suspended and processing continues without the exit being invoked.

As an example, the RACF unload utility exits may be invoked as follows:

```
//RACFTTEST JOB  (ACCT,AAA,999),TEST,TIME=1439,MSGCLASS=X
/*JOBPARM LINES=9999
//TEST      EXEC  PGM=SMFUTIL,REGION=0K
//SYSPRINT  DD  SYSOUT=*
//ADUPRINT  DD  SYSOUT=*
//OUTDD     DD  DSN=BMCCDAW.RACF.DATAOUT,DISP=(,CATLG),
//           UNIT=SYSDA,SPACE=(CYL,(5,5)),
//           DCB=(LRECL=6000,BLKSIZE=0,RECFM=VB)
//SYSUDUMP  DD  SYSOUT=*
//SYSIN     DD  *
DETAIL
ESTAEOFF
SPIEOFF
INCLUDE(30,80,81,83)
DDASTART DDNAME(SYSUT1) DSN(SMF.INPUT.DATA) DDAEND
DDASTART DDNAME(SYSUT2) DUMMY DDAEND
USER2(IRRADU00)
USER3(IRRADU86)
/*
```

Note that the SMFUTIL output data set SYSUT2 is present and is DUMMY. This is because the only purpose of this run is to produce the output OUTDD data set. SIMULATE may not be specified because the USER2 and USER3 exits are called during output phase processing and this phase does not exist in a SIMULATE execution.

13. SMFUTIL REPORTS

SMFUTIL produces several required and optional reports that the quantity and type of input and output SMF data and the operations performed on the data. The following list details the reports, when they are produced and how they may be requested or suppressed:

REPORT: EXECUTION CONTROL DEFAULTS.

Produced: Before control images read in.

Purpose: To list the default execution control environment in place at program start.

Requested By: DFLTLIST on 'PARM=' on EXEC JCL card.
DFLTLIST=YES in SMFUDFLT module.

Suppressed By: NODFLTLI on 'PARM=' on EXEC JCL card
DFLTLIST=NO in SMFUDFLT module

NOTES: Cannot be suppressed at execution time via a SYSIN control statement because it is produced before these statements are read in.

REPORT: EXECUTION CONTROL STATEMENT PROCESSING.

Produced: While control images are read in.

Purpose: To list the control statements supplied by user and any errors encountered.

Requested By: REQUIRED.

Suppressed By: REQUIRED

NOTES: Cannot be suppressed.

REPORT: DYNAMIC ALLOCATION PROCESSING REPORT.

Produced: Inline when dynamic allocation or unallocation takes place.

Purpose: To show data set and volumes dynamically allocated.

Requested By: REQUIRED.

Suppressed By: REQUIRED.

NOTES: Cannot be suppressed.

REPORT: INPUT DATA SETS – DATA CONTENT REPORT.

Produced: At end of run.

Purpose: To show the first/last and high/low date/time spans and record counts of each input data set. A histogram is also produced to show the specific record types encountered for each data set.

Requested By: REQUIRED.

Suppressed By: REQUIRED.

NOTES: Cannot be suppressed.

REPORT: OUTPUT DATA SETS – DATA CONTENT REPORT.

Produced: At end of run.

Purpose: To show the first/last and high/low date/time spans and record counts of each output data set. A histogram is also produced to show the specific record types placed to each data set.

Requested By: REQUIRED.

Suppressed By: REQUIRED.

NOTES: Cannot be suppressed. The OUTPUTDETAIL parameter may be specified to request complete detailed statistics on each record type placed to each output data set.

REPORT: STATISTICAL RANGE OF DATA REPORT.

Produced: At end of run.

Purpose: Shows first, last, high and low date/time stamps encountered overall.

Requested By: REQUIRED.

Suppressed By: REQUIRED.

NOTES: Cannot be suppressed.

REPORT: TOTAL DATA MOVEMENT REPORT.

Produced: At end of run.

Purpose: Shows total blocks, records, and segments input and output plus total records selected for output. Also shows any error counts such as I/O errors, dropped segments, etc.

Requested By: REQUIRED.

Suppressed By: REQUIRED.

NOTES: Cannot be suppressed.

REPORT: DETAILED STATISTICS REPORT.

Produced: At end of run.

Purpose: To show the high/low date/time range of each record type plus the total input and out counts and the minimum, maximum, and average length.

Requested By: DETAIL keyword or DETAIL=YES in SMFUDFLT module.

Suppressed By: NODETAIL keyword or DETAIL=NO in SMFUDFLT.

NOTES: If this report is suppressed a simple listing of the record types and in and output counts for each type will be listed.

REPORT: ARCHIVE PROCESSING REPORT.

Produced: At end of run when ARCHIVE update activity has been done.

Purpose: To show all ARCHIVE database activity that occurred because of this run.

Requested By: Any archive update functions such as ARCHIVE, ARCHUSER or ARCHIVEINPUT.

Suppressed By: Not doing ARCHIVE updates during the run.

NOTES: Will always be produced if changes are made to the ARCHIVE. This report will also show if an automatic ARCHIVE database backup was performed.

REPORT: MANXALOC/DUMPCLEARALL ACTION MESSAGES.

Produced: During MANXALOC and DUMPCLEARALL data set selection processing.

Purpose: To show each SMF MANx data set that is being selected for processing. Each data set listed will show the beginning date/time stamp of the data found in the data set.

Requested By: MANXALOC or DUMPCLEARALL keywords

Suppressed By: REQUIRED if these keywords are used.

NOTES:

REPORT: DUMPCLEARALL – CLEAR PROCESSING REPORT.

Produced: During DUMPCLEARALL executions when the CLEAR phase is processing.

Purpose: To show each SMF MANx data set that is being cleared and the amount of clear blocks that were written to it.

Requested By: DUMPCLEARALL

Suppressed By: REQUIRED if DUMPCLEARALL is requested

NOTES: May be produced upon restart of previous run when previous dump had completed and clear was not completed.

REPORT: DELETE PROCESSING REPORT.

Produced: At end of successful run when DELETE specified on CATINPUT or GDGINPUT control cards.

Purpose: To list the data sets deleted as per the user's request.

Requested By: CATINPUT or GDGINPUT

Suppressed By: REQUIRED if requested.

NOTES: Data sets only deleted upon successful completion of execution.

REPORT: BYTE DETAILED STATISTICS.

Produced: At end of run.

Purpose: To show the bytes moved by record type. Both total and percentage are shown.

Requested By: BYTEDETAIL or BDETAIL=YES in SMFUDFLT module.
OUTPUTDATADISTRIBUTION (alias ODIST)

Suppressed By: NOBYTEDETAIL keyword or BDETAIL=NO in SMFUDFLT module.

REPORT: DATA DISTRIBUTION BY HOUR REPORT.

Produced: After each input DD statistical report and/or each output DD statistical report.

Purpose: A matrix report showing 24 columns of data. Each column, labeled 00 to 23 along the bottom, represents one hour of the day inclusive (i.e. 00 represents 00:00 thru 00:59).

Each row of the report represents a single record type from 0 to 255 in descending order. If no data was encountered for a particular record type that row is omitted.

Each intersection of a row and column contains the percentage of that record type for that hour.

If the report type is RECTYPE, the percentage represents the amount of that record type received during that hour compared to the number of that record type received during the 24 hour. Each row will sum to 100%.

If the report type is ALL, the percentage represents the amount of that record type received during that hour compared to the total number of all record type received during the 24 hour.

Both types of reports (RECTYPE and ALL) may be produced in the same execution.

Requested By: INPUTDATADISTRIBUTION (alias IDIST)
OUTPUTDATADISTRIBUTION (alias ODIST)

Suppressed By: Optional only, cannot be produced by default.

14. RETURN CODES

SMFUTIL uses return codes to indicate the result of the requested operation(s). The following table shows possible return code combinations and the meaning of each:

- | | |
|-----|---|
| 0 | Operation completed successfully. The output datasets contain the requested output data. |
| 1 | Operation completed successfully. A day split was encountered on an output data set. |
| 2 | Operation completed successfully. A week split was encountered on an output data set. |
| 3 | Operation completed successfully. A month split or a split on a user calendar boundary was encountered on an output data set. |
| 4 | Operation completed with possible lost data due to output block size specification or some data did not pass user specified validation criteria or no records were found for a user specified directed output DD name.

The user exit issued a return code greater than 4. Not all input data was processed in this case. |
| 5-7 | Same as 4 above with output data being split as described for 1, 2, and 3 above. |
| 8 | Operation failed during processing. Error messages in SYSPRINT give information on the reason(s) for the failure. |
| >8 | Operation failed at processing startup or fatal error during processing. Error messages in SYSPRINT or on JOBLOG give information on the reason(s) for the failure. |

Note that the return codes 1, 2, and 3 (and 5, 6 and 7) require careful consideration as to their meaning. These values represent a change from those in versions prior to release 5.0. This change was necessary so split indications from user TYPETODD data set could be incorporated. A 2 (week split) or a 3 (month split) would imply a 1 (day split) had also occurred. A 3 would not necessarily imply a 2 (week split) as a week may have a different boundary than an end of month. Typically, week split and month split operations should not be combined in a single step where the resulting return code for split is used to determine subsequent operations for both weekly and monthly datasets.

15. APPENDIX A - CONTROL & EXECUTION ERROR MESSAGES**SMFU000 SYSIN NOT SPECIFIED - DEFAULTS ONLY USED.**

No input control statement data set (SYSIN) is present in the JCL. The execution will proceed based on the default specifications listed in the output.

SMFU001 CALL ADVANCED SOFTWARE FOR SUPPORT.

Call SMFUTIL technical support for assistance with the problem encountered.

SMFU002 NO CONTROL CARDS ON SYSIN - DEFAULTS ONLY USED.

No control statement statements were present on the SYSIN control input data set. The execution will proceed based on the default specifications listed in the output.

SMFU004 KEYWORD xxxx IS UNDEFINED AND IS IGNORED.

The indicated keyword parameter is undefined to SMFUTIL. Check for a syntax or spelling error. Correct the indicated control card and rerun.

SMFU006 **WARNING - NO DATA COPIED

No data was copied to any output file. This could be caused by:

1. All input record types were EXCLUDED.
2. No data found in DATE or TIME range specified.
3. No CURRENTWEEK data found.
4. No PREVWEEK data found.
5. No data found for specified JOBNAME or USERID.
6. User exit deleted all records.
7. No data for WEEKEND selection.
8. No data for WORKDAY selection.
9. Site Security denied access to all requested records.
10. No data for specified day of week.

SMFU007 I/O ERROR - message.

This message is produced only when an input/output error is detected and SMFUTIL is operating in the XERROFF mode. It is generated by the SYNAD error processing routine and shows the DD name involved and more information about the error.

SMFU008 RECFM=VB SPECIFIED FOR DDNAME ddd. LRECL SET TO 4 LESS THAN MAXIMUM BLKSIZE FOR DEVICE.

This is not an error merely an informational message. The DCB specification for the indicated DD statement has RECFM=VB specified with no LRECL or BLKSIZE specified. SMFUTIL has set the BLKSIZE to the optimum maximum for the device and the LRECL to the same value less 4 for the block descriptor word.

SMFU010 RECFM=V OR VB SPECIFIED FOR DDNAME ddd. LRECL SET TO 4 LESS THAN USER SPECIFIED BLKSIZE.

This is not an error merely an informational message. The DCB specification for the indicated DD statement has RECFM=VB specified with a BLKSIZE but no LRECL. SMFUTIL has set the LRECL to the BLKSIZE value less 4 for the block descriptor word.

SMFU011 BLKSIZE FOR DDNAME ddd IS LARGER THAN MAXIMUM SUPPORTED BY DEVICE. IT HAS BEEN RESET TO MAXIMUM FOR DEVICE.

The DCB specification for the indicated DD statement has contained a user supplied BLKSIZE value that is invalid (too large) for the device type in use. SMFUTIL has reset the BLKSIZE to the largest value allowed on the device. This may not be the best value to use and may also result in message SMFU019.

SMFU012 ERROR - LOAD FAILED FOR MODULE lmodname.

The load module exit routine or table 'lmodname' specified via the EXIT keyword or MONTHCAL keyword was not found in STEPLIB, JOBLIB, LINKLIST, or LPA.

SMFU013 SMFUTIL IS TERMINATING DUE TO REQUEST BY USER EXIT xxxxxxxx.

The user exit routine xxxxxxxx returned a return code greater than 4. SMFUTIL is terminating before all input data has been processed. An execution return code of 4 plus any xSPLITxx return code will be issued.

SMFU014 INVALID OUTPUT RECFM SPECIFIED FOR DDNAME ddd. RESET TO VBS.

The output record format for SMF data must be variable (V, VB, or VBS). SMFUTIL detected a RECFM specification from the DD statement that was F or U. To avoid an ABEND, SMFUTIL has reset the RECFM to VBS (the optimum). This could be caused by specifying a RECFM in the DCB other than V/VB/VBS or by using a model DSCB for GDG data set that has a RECFM specification of F or U. This message is for information purposes only and will not affect the return code of the execution. The output file(s) will be valid VBS SMF data set.

SMFU015 ERROR - RDJFCB FAILURE FOR DDNAME ddd.

The JFCB for the specified DDNAME could not be located. Verify that the JCL is correct and a proper data set is supplied for the specified ddname.

SMFU016 BLKSIZE NOT FOUR GREATER THAN LRECL FOR DDNAME xxxxxxxx. LRECL RESET TO BLKSIZE MINUS 4.

The indicated DD statement contained RECFM=VB with an LRECL that was not at least 4 less than the BLKSIZE value. SMFUTIL has reset the LRECL to the BLKSIZE minus 4 for the block descriptor word.

SMFU017 RECFM=VB SPECIFIED FOR DDNAME xxxxxxxx. SPECIFIED LRECL INVALID FOR DEVICE. LRECL RESET TO MAXIMUM BLKSIZE MINUS 4.

The indicated DD statement contained RECFM=VB with an LRECL value and no BLKSIZE value. The LRECL value specified was larger than the largest physical block the output device supports. SMFUTIL has reset the LRECL to the maximum BLKSIZE minus 4 for the block descriptor word.

SMFU019 BLKSIZE FOR DDNAME xxxxxxxx IS LARGER THAN OPTIMUM. SPACE UTILIZATION WILL BE DEGRADED.

The BLKSIZE value in use for the indicated DD statement is larger than the largest block that the output device can store efficiently (i.e. 3380's are optimally BLKSIZE=23476. A BLKSIZE=32000 will result in this message). The BLKSIZE is unchanged and the space required to store the data set may be much larger than if an optimum BLKSIZE were used.

SMFU032 DATESCAN REQUESTED. DATA IS MISSING.. GAPS LARGER THAN DELTA FOUND. REPORT FOLLOWS:

This is a header message that precedes a series of one or more SMFU035 messages.

SMFU033 WARNING - ACTIVEOK CHANGED TO NOACTIVE TO PREVENT ACTIVE MAN FILE FROM BEING ARCHIVED.

ARCHIVING was requested with MANXALOC and ARCHMANX overrode the restriction of not archiving when MANXALOC is used. The ACTIVEOK specification was turned off to prevent the active SMF MANx data set from being archived more than once.

SMFU034 OUPUT DATA NOT IN SORTED ORDER. DATESCAN SUSPENDED.

DATESCAN operates best on sorted data. It will suspend operation when non-sorted input data is detected. The FORCE keyword may be specified to force DATESCAN to continue.

SMFU035 GAP BETWEEN ddd mm/dd/yyyy hh:mm:ss.th (TYPE rr) AND ddd mm/dd/yyyy hh:mm:ss.th (TYPE rr) IS xx MINUTES.

The DATESCAN function has detected a gap in the input data larger than the DATESCAN delta. The message indicates the dates and times and record types of the records that bound the gap. The gap must exceed the DATESCAN delta for this message to be produced.

NOTE: A gap larger than 24 hours will be printed as 0. An asterisk (*) will precede any entry caused by out of order data and the gap will be printed as 0.

SMFU038 INPUT SORT CHECKING REQUESTED. INPUT RECORDS ARE NOT IN DATE/TIME SORT SEQUENCE.

The CHKISORT function has detected output data that is not within the sort tolerance.

SMFU040 ERROR - DDNAME IS INVALID AS OBJECT DD FOR kkkk KEYWORD. MUST BE INPUT DATA SET WITH SYSUT1 PREFIX.

The specified keyword "kkkk" requires an input DDNAME (i.e. one with SYSUT1xx format) as it's object. The command is invalid.

SMFU042 NOTE - DSNAME dsn PREVIOUSLY ALLOCATED AS NEW. DISPOSITION CHANGED TO MOD FOR RESUSE.

An initial disposition of NEW was specified for the named data set. Out of order data caused the data set to be used more than once in this execution (freed and reallocated). The disposition is changed to MOD so that additional data may be added to the data set for this execution. The normal and conditional dispositions are left unchanged.

SMFU043 NOTE - DSNAME dsn DOES NOT CURRENTLY EXIST. DISPOSITION CHANGED FROM MOD TO NEW.

An initial disposition of MOD was specified for the named data set but the data set did not currently exist. The disposition is changed from MOD to NEW. In addition, the normal disposition is left as is and the conditional disposition is changed to DELETE. This alteration facilitates a restart in the event of an abnormal termination of the run.

SMFU044 MOD DATA SET CHECKPOINT RECORD dddddddd KEPT DUE TO RUN FAILURE. CORRECT ERROR AND RERUN.

The MOD data set checkpoint record for DDNAME 'dddddddd' was retained in the checkpoint data set because the run completed with a return code of 8 or greater or ended with an ABEND. Study the SYSPRINT for other error messages indicating what error caused the failure. Correct the error and resubmit the job. The output data set will be returned to it's original condition upon restart.

SMFU050 INVALID SEGMENTS - x.

The number of input record segments that did not meet spanned record structural requirements and were therefor discarded.

SMFU052 INVALID RECORDS - x.

The number of input records that did not meet SMF record structural requirements (i.e. invalid date fields in record header) and were therefor discarded.

SMFU053 OVERFLOW RECORDS - x.

The total number of times that an output record could not be placed to an output data set because the output DCB specifications would not allow it. An example would be a VB (non-spanned) data set with a small LRECL and/or BLKSIZE being the target for a record with a record length larger than the LRECL or BLKSIZE.

SMFU057 I/O ERRORS - x.

The number of input block reads that failed due to an I/O error. Each failing block is dropped and input begins with the next block successfully read. Any record currently under construction is discarded and record construction begins again with the starting record segment found in the next block.

SMFU059 OVERSIZE BLOCKS - x.

The number of input blocks that were larger than the input data set DCB block size indicate was the maximum. This message is issued in concert with message SMFU098.

SMFU061 ERRORS ENCOUNTERED:

This message is produced when unknown keywords are entered on a control statement. It will contain a '?' directly under each offending keyword or error. This message is not printed until after the entire control statement is scanned. There may be other intervening messages between this line and the control statement if errors occurred processing one or more keywords.

SMFU063 ERROR(S) ENCOUNTERED. REMAINING CONTROL CARDS WILL BE SYNTAX CHECKED ONLY.

This messages indicates that execution cannot proceed to process data. A major error has been found in the control statement set.

SMFU065 OPEN FAILED FOR CONTROL CARD DATA SET (SYSIN).

A SYSIN DD statement was present but open failed for the data set.

SMFU066 I/O ERROR READING CONTROL CARD DATA SET. EXECUTION TERMINATED.

The control statement data set could not be read.

SMFU067 ERROR - UNBALANCED PARENTHESIS DETECTED ON kkkkkkkk KEYWORD.

The specified keyword had a missing parenthesis or invalid syntax in the parmlist.

SMFU068 ERROR - PARMLIST REQUIRED FOR kkkkkkkk KEYWORD.

The keyword indicated requires a parameter list.

SMFU069 ERROR - RDJFCB-ARL FAILURE FOR DDNAME ddname. ATTEMPTING STANDARD RDJFCB.

The RDJFCB type x'13' failed. SMFUTIL will attempt to continue with a standard RDJFCB type x'07' and construct an internal ARL list.

SMFU071 ERROR - INVALID RANGE SPECIFIED IN kkkkkkkk PARMLIST - START GREATER THAN END.

When a range of record types is specified, the start must be less than the end.

SMFU072 ERROR - INVALID tttt VALUE IN kkkk PARMLIST - GREATER THAN maxvalue.

A numeric value was out of range for the specified keyword parmlist.

SMFU073 SMFUTIL IS RUNNING NON-AUTHORIZED ON SYSTEM sysid DYNAMIC ALLOCATION WILL NOT BE ABLE TO WAIT FOR VOLUMES OR DATA SETS.

The current execution is from a non-authorized environment (i.e. NOT from LNKLISTxx or APF authorized data set). Because of this SMFUTIL will not be able to request that dynamic allocation wait for 'inuse' devices, volumes, or data sets to become available.

SMFU074 ERROR - SMFUTIL TERMINATED DUE TO CONTROL CARD ERRORS.

An error was detected during control card processing. The preceding error messages describe the error(s). Correct the control card set and rerun the job.

SMFU075 ERROR - INVALID DATE STRING IN kkkk PARMLIST - reason.

A date value specified in the “kkkk” keyword was invalid for the reason stated by “reason”.

SMFU079 NOTE: DEFAULT ARCHSMF# HAS BEEN OVERRIDDEN. ARCHIVE UPDATE RECORDING INTO SMF RECORD IMAGES ARE NOW ENABLED. ARCHSMF MUST ALSO BE SPECIFIED.**WARNING: CURRENT ARCHSMF# DEFAULT VALUE IS NOT ZERO.**

The default ARCHSMF# value has been overridden to enable ARCHIVE update images to be recorded via SMF records. The ARCHSMF keyword must also be specified to request recording to begin. The WARNING message is issued if the current value of ARCHSMF# is non-zero. This could cause a subsequent ARCHIVERECOVER execution to not find all ARCHIVE update records as more than one SMF record value was used.

SMFU083 ERROR - INVALID TIME STRING IN kkkk PARMLIST - reason.

A time value specified in the “kkkk” keyword was invalid for the reason stated by “reason”.

SMFU086 ERROR - SUBSET NAME yyyyyy IS UNDEFINED.

The value for xxxxxx name indicated was not defined as a subset name.

SMFU088 ERROR - NO ssss ENTRY FOUND IN SUBSET TABLE. TABLE HAS BEEN MODIFIED - CORRECT AND RERUN.

The subset name ssss was not found in the subset table but was present when the product was shipped. The table has been locally modified and the name was deleted. Reinstall the name and rerun the job.

SMFU089 ERROR - PARMLIST NOT ALLOWED FOR kkkkkkkk KEYWORD.

The indicated keyword does not need a parmlist.

SMFU090 FIRST VOLUME BLOCK ID START: NOT AVAILABLE

The selected data set's archive record was created by a prior version of SMFUTIL and does not contain any Block ID information on where specific dates are located on the volume. Process the data set thru SMFUTIL as an input data set and specify ARCHINPT and SIMULATE to re-archive the data set with BID information stored in the archive record..

SMFU092 RESTART CLEANUP IN PROCESS. DASD DATA SET dsname ON VOLUME volser HAS BEEN SCRATCHED.

A previously aborted run has been restarted. The indicated disk data set that was created in the previous run has been scratched. The current run will recreated it.

SMFU095 OUTPUT SORT CHECKING REQUESTED. OUTPUT RECORDS ARE NOT IN DATE/TIME SORT SEQUENCE.**RECORD CAUSING OUT OF ORDER CONDITION WAS TYPE rrr.**

CHKOSORT was requested and the output data is not in sorted order within the sort tolerance. The first record found to be out of order was record type 'rrr'.

SMFU098 WARNING - OVERSIZE INPUT BLOCKS ENCOUNTERED. INPUT DATA SET LABEL/DSCB/DCB IS INCORRECT.

The input data set on SYSUT1 contained input blocks of data larger than the block size that was specified in the data set label (for tape), the format 1 dscb (for disk) or the user specified DCB. This can occur if a data set is improperly created or damaged. It may also occur when two or more data set are concatenated on SYSUT1 if the first data set in the concatenation has a block size smaller than one or more of the other data set. This is not an error. SMFUTIL can handle oversize input blocks without problems. SMFUTIL is only warning you that if this same input data set or concatenation of input data set is input to a program other than SMFUTIL an ABEND will likely occur. The OUTPUT data set of SMFUTIL will be correct and usable by other SMF processing programs.

SMFU099 ERROR - INPUT (ddname) NOT AVAILABLE.

The input DDNAME was not found in the JCL.

SMFU100 ERROR - OUTPUT (ddname) NOT AVAILABLE TO ACCEPT DATA.

The output DDNAME was not found in the JCL.

SMFU101 ERROR - OUTPUT (ddname) NOT AVAILABLE TO ACCEPT DATA IF REQUESTED SPLIT OCCURS.

The specified DDNAME is required for the requested split operation but was not found in the JCL.

SMFU104 ERROR - INVALID SYSTEM TYPE VALUE IN SYSTYPE. VALID TYPES ARE MVS, MXA, ESA, VS1, SVS, SP2, SP3, SP4, OR SP5.

Specify one of the supported types.

SMFU105 DDNAME ddn - NO DCB SPECIFIED FOR UNKNOWN DEVICE TYPE. SPECIFY DCB AND RERUN.

No DCB was specified for a device type unknown to SMFUTIL. Specify the DCB and rerun the job.

SMFU106 * - INDICATES THAT MISSING DATA RANGE WAS CAUSED BY OUT OF SORT INPUT DATA.

The FORCE keyword caused DATESCAN to continue operation even after out of sort input data was encountered. The '*' indicated data gaps caused by the unsorted areas of the input data. These may be caused by duplicated data blocks in the data.

SMFU109 OPEN FAILED FOR DDNAME ddn.

Specified DDNAME data set could not be opened. Verify that the data set is specified correctly in the JCL.

SMFU110 ERROR - INVALID name SPECIFIED IN kkkk KEYWORD - MORE THAN x CHARACTERS.

The specified keyword "kkkk" contained a value for the field "name" that was too long.

SMFU112 ERROR - UNBALANCED PARENTHESIS DETECTED ON kkkk KEYWORD.

The keyword "kkkk" contained unpaired parenthesis and could not be parsed successfully.

SMFU113 ERROR - DDNAME CANNOT BE USED AS TARGET FOR DIRECTED OUTPUT IN KEYWORD kkkkkkkk.

The DD names normally used by SMFUTIL are restricted and may not be used as targets in a TYPETODD command.

SMFU114 ERROR - DDNAME IS INVALID AS A DDNAME FOR DIRECTED OUTPUT IN KEYWORD kkkkkkkk.

The DDNAME specified must be valid as to IBM DDNAME specifications.

SMFU115 WARNING - SPECIFIED IVSTART VOLUME GREATER THAN VOLUME COUNT IN JFCB. LAST VOLUME USED.

The IVSTART specification was too large. The volume count for the data set is less than the IVSTART value. The last volume is used.

SMFU116 INVALID SYNTAX IN kkkk PARMLIST

The specified keyword contained invalid syntax.

SMFU119 ERROR - INVALID VALUE SPECIFIED IN kkkk KEYWORD - NOT ALL NUMERIC

A non-numeric field was specified where a number was expected.

SMFU122 **WARNING - NO INPUT DATA FOUND ON SYSUT1.

No input data was found to process. The archive search did not find any required data or the user supplied input data set was empty or all data was bypassed. If this is an expected condition, specify the COPY00K keyword to allow it.

SMFU124 DDNAME RECEIVED NO OUTPUT. DATA SET NEVER|WAS OPENED.

The DDNAME specified had not data for output. The data set was not opened and thus not created or was opened as a result of the CREATEU0 or CREATES0 parameter.

SMFU125 SMFUTIL LICENSE EXPIRED ON yyyy.ddd. CONTACT ADVANCED SOFTWARE AT 1-813-649-1548 FOR ASSISTANCE.

The program will not execute because the site license for SMFUTIL has not been renewed. Upon renewal a zap will be supplied to enable the program for the new license period.

SMFU126 SMFUTIL LICENSE WILL EXPIRE ON yyyy.ddd. CONTACT ADVANCED SOFTWARE AT 1-813-649-1548 FOR ASSISTANCE.

The license period for the program product is scheduled to expire on the Julian date specified. Upon execution of a renewal agreement, a code string and expiration date will be supplied to extend the license date. SMFUTIL will function normally until the specified expiration date occurs.

SMFU127 SMFUTIL LICENSE CODE IS INVALID. CONTACT ADVANCED SOFTWARE AT 1-813-649-1548 FOR ASSISTANCE.

An invalid license code was specified when the product was installed. Ensure that the code, date, and/or a serial number were not incorrectly entered in the INSTALL job. Validate the information from the shipping cover letter was correctly transferred to the INSTALL JCL. If required, call ASPG technical support for assistance.

SMFU128 SMFUTIL IS NOT LICENSED FOR THIS PROCESSOR (MODEL=mmmm, SERIAL=ssss). CONTACT ADVANCED SOFTWARE AT 1-813-649-1548 FOR ASSISTANCE.

An attempt was made to execute SMFUTIL on a processor that was not part of the processor CPU serial matrix supplied during installation. Ensure that the correct CPU's were designated in the contract license with Advanced Software. Contact ASPG Contract Administration at 1-813-649-1548 for assistance in updating the site records so a correct license code can be generated for all processors at the installation.

SMFU130 SYSOUT DD STATEMENT NOT AVAILABLE. PRINT FUNCTION NOT EXECUTABLE.

A SYSOUT DD statement must be supplied in the JCL stream for the PRINT function to put SYSOUT to.

SMFU131 CHECKPOINT RECORD FOUND FOR DDNAME ddn. BLOCK SEARCH IN PROGRESS.

SMFUTIL has detected the presence of a checkpoint record for the specified DD name. It is attempting to locate the restart point.

SMFU132 RESTART POINT FOUND FOR ddname. RESTART SUCCESSFULLY PROCESSED.

The restart point has been found for the specified DD name.

SMFU133 RESTART POINT NOT FOUND FOR DDNAME ddn. RESTART FAILED. DATA ASSUMED VALID AS IS.

The restart point could not be found for the specified DD name. The copy will proceed with the available data. This could be caused by changes made between the failure and the restart. Upon failure, the job must be rerun exactly as is with no changes.

SMFU134 OPEN FAILURE DURING RESTART FOR DDNAME ddn.

The DDNAME specified could not be opened for INOUT during the restart process. A security access of write only could cause this failure.

SMFU136 RESTART POINT DOES NOT MATCH PREVIOUS POINT FOR DDNAME ddname. INPUT HAS CHANGED. RERUN WITH ORIGINAL INPUT OR DELETE CKPT RECORD.

During a restart for the specified ddname, the first record to be output did not match the previous runs first output record (the previous checkpoint). The input data or the order of concatenation of input has been changed. Execution continues using the input data supplied.

SMFU139 SMFUTIL HAS NOT BEEN PROPERLY INSTALLED. YOU MUST RUN THE INSTALL JOB BEFORE EXECUTING THE SMFUTIL PROGRAM.

SMFUTIL must be installed via the INSTALL job stream supplied with the distribution tape in the INSTLIB data set. This job properly applies the license code and existing maintenance.

SMFU142 DSNNAME SUFFIX GENERATION VALID ONLY FOR TAPE OUTPUT. DDNAME ddname IS NOT ON TAPE DEVICE. USE DDA BLOCK TO SUFFIX DASD.

The DDNAME specified is not a tape device. Automatic data set name suffix generation (selected by GENSDND or GENSDNM) is only valid for tape data set. If data set name suffix generation is desired for this data set, move the data set to a tape device. The original data set name as specified on the DD card is used for the disk data set and processing continues. As an alternative, you may code a DDASTART/DDAEND block to allocate the data set and use the SUFFIX parameter within the block.

SMFU143 DSNNAME SUFFIX GENERATION VALID ONLY FOR NEW DATA SETS. DDNAME ddname IS NOT NEW. USE DDA BLOCK FORMAT.

The DDNAME specified does not specify a new data set. Automatic data set name suffix generation (selected by GENSDND or GENSDNM) is only valid for new tape data set. A disposition of MOD or OLD is not allowed. If OLD was used, delete the data set name (if cataloged) with a NOSCRATCH specification and rerun the job using a disposition of NEW. If MOD was used to add data to a previously created data set that was named via a GENSDNx in order to complete the data set you may 1) rerun the job omitting the GENSDNx specification and completing the data set name in the JCL statement, or 2) delete the data set and recreate it from scratch with a GENSDNx specification and all required data concatenated on SYSUT1. The DATE specification may be used to select a required date or date range.

SMFU144 ACB CONTROL BLOCK GENERATION FAILED FOR VSAM INPUT PROCESSING.

An internal problem has caused a VSAM processing failure. Call WOVEN SOFTWARE for assistance.

SMFU145 OPEN RETRY FAILED FOR VSAM INPUT DATA SET ON SYSUT1. ENSURE DISP=SHR ELSE INCREASE REGION SIZE.

A SMF VSAM data set was unreadable. Make sure that DISP=SHR is used as SMF always has the data sets allocated. Also increase the region size.

SMFU146 GET FAILED FOR BLOCK IN VSAM INPUT DATA SET.

RC=xxxx,FUNC=xxxx,FDBK=xxxx.

An SMF VSAM data set was unreadable. The FUNC (function) and FDBK (feed back) codes may be found in the VSAM Programmers Guide. If the data set is known to contain data, perform an IDCAMS verify (it may take up to eight attempts to clear the error) and retry the failing copy step. Call WOVEN SOFTWARE for assistance. This may be caused by a non-initialized SMF data set or by a failed CLEAR function that damaged the data set. The data in the data set may not be recoverable. It may be necessary to delete and reallocate the data set and then re-initialize it. Ensure that the data set is allocated correctly for SMF usage.

SMFU147 RPL CONTROL BLOCK GENERATION FAILED FOR VSAM INPUT PROCESSING.

An internal problem has caused a VSAM processing failure. Call WOVEN SOFTWARE for assistance.

- SMFU148 EXLST CONTROL BLOCK GENERATION FAILED FOR VSAM INPUT PROCESSING.**
An internal problem has caused a VSAM processing failure. Call WOVEN SOFTWARE for assistance.
- SMFU149 GENCB CONTROL BLOCK GENERATION FAILED FOR VSAM xxx DURING yyyy PROCESSING.**
An internal problem has caused a VSAM processing failure. Increase the region size and rerun. If the problem persists, call WOVEN SOFTWARE for assistance.
- SMFU151 OPEN FAILED FOR VSAM DATA SET OPENED FOR OUTPUT DURING CLEAR PROCESSING. RETRYING OPEN.**
A VSAM SMF data set did not open on the first try. This is usually caused by a data set that was not closed properly due to a cancel or system failure in a previous run. Retry should correct the situation.
- SMFU152 INVALID VALUE vvvv FOUND ON THE ssss SUBPARM FIELD OF THE kkkk KEYWORD. VALID VALUES: list**
The subfield “ssss” of the keyword “kkkk” contained an invalid value of “vvvv”. Only values in the “list” are valid for this subfield.
- SMFU153 PUT FAILED FOR BLOCK TO VSAM SYSUT1 DATA SET DURING CLEAR. RC=xxxx, FUNC=xxxx, FDBK=xxxx.**
An SMF VSAM data set was could not be written to clear it. The FUNC (function) and FDBK (feed back) codes may be found in the VSAM Programmers Guide. This error may be caused by a non-initialized SMF data set or by a failed CLEAR function that damaged the data set. If the data set is known to have already been used, perform an IDCAMS verify (it may take up to eight attempts to clear the error) and retry the failing CLEAR step. If this does not correct the problem, it may be necessary to delete and re-define the data set and then re-initialize it. Ensure that the data set is allocated correctly for SMF usage.
- SMFU154 SMF NOT ACTIVE IN SYSTEM. DATA SET WILL BE CLEARED. SMF NOT POSTED.**
SMF is not currently recording in the system. The data set will be cleared as requested but SMF will not begin to record to it until SMF is activated (via a ‘T SMF=xx’ command from the operators console).
- SMFU155 DATA SET dsname ON VOLUME vvvvvv NOT FOUND IN CURRENT SMF CONTROL BLOCKS. SMF NOT POSTED.**
The listed data set is not currently one of the SMF recording set. It will be cleared (using the format of the current operating system - SP or XA) but SMF will not be able to record to it on this system. This is regarded as a ‘foreign’ data set.
- SMFU156 DATA SET dsname ON VOLUME vvvvvv DOES NOT SHOW DUMP REQUIRED. DATA SET BYPASSED.**
SMF does not list the listed data set as requiring a dump at this time so the clear will be bypassed.
- SMFU157 DATA SET dsname ON VOLUME vvvvvv IS CURRENTLY THE ACTIVE SMF DATA SET. CLEAR ABORTED.**
SMF is currently recording on the listed data set. CLEAR will not be executed.
- SMFU158 DATA SET dsname ON VOLUME vvvvvv IS CURRENTLY ENQUEUED BY ANOTHER SMFUTIL JOB USING CLEAR. CLEAR ABORTED.**
SMFUTIL is already processing the listed data set. Only one run (copy or clear) may be executed at one time.
- SMFU161 INPUT DATA SET NOT VSAM. CLEAR INVALID AS USED.**
CLEAR only functions on valid, initialized, SMF data set that are VSAM format.

SMFU162 CLEAR SPECIFIED BUT ACTIVE CHECKPOINT RECORDS FOUND. RERUN INCOMPLETE COPY STEPS PRIOR TO CLEAR REQUEST.

A CLEAR execution detected checkpoint records present that may indicated an incomplete copy step. If the copy step was not completed, rerun it to correct completion and then re-submit the clear request. If the copy finished correctly, delete the checkpoint directory entries in the checkpoint PDS data set and rerun the clear job.

SMFU163 CLEAR IN PROGRESS CHECKPOINT RECORD FOUND BUT CLEAR NOT SPECIFIED. RERUN WITH CLEAR.

A copy execution detected a clear in progress checkpoint record when attempting to read the input SMF data set. A previous check pointed clear request did not complete. The data set is probably partially cleared and therefore unreadable. Rerun a clear against the data set if it is certain that SMF has not put additional data in it, else delete the clear checkpoint record and rerun the copy step.

SMFU164 CLEAR REQUIRED CHECKPOINT INDICATOR FOUND. RERUN ASSUMED. INPUT DATA SET WILL BE CLEARED.

No data movement may take place in a clear run. See the above text on CLEAR and the restrictions on it's use.

SMFU166 CLEAR FUNCTION MAY NOT BE SPECIFIED WITH ANY OTHER KEYWORDS. CLEAR IS A STANDALONE FUNCTION. USE CLEAR IN A SECOND STEP AFTER COPY COMPLETE.

No data movement may take place in a CLEAR execution run. See the above text on CLEAR and the restrictions on it's use.

SMFU168 SYSUT2 NOT VALID WHEN CLEAR SPECIFIED. NO DATA MOVEMENT ALLOWED. CLEAR FUNCTION NOT PERFORMED.

No data movement may take place in a clear run. See the restrictions on the use of the CLEAR command.

SMFU169 OPEN FAILED FOR VSAM INPUT DATA SET. RETRYING OPEN.

Indicates a problem opening a 'SYS1.SMF' data set. This message will be followed by either SMFU170, SMFU171, or SMFU172 indicating the status of the retry. If the retry is successful, this message can be ignored. If not, ensure that the DD statement involved has DISP=SHR and that the data set is not in use by other users. If the open fails again, perform an IDCAMS verify against the data set and try it again. Several verifies may be required to correct a damaged data set.

SMFU170 OPEN RETRY SUCCESSFUL FOR VSAM DATA SET ON SYSUT1.

The open retry was successfully done and processing will proceed normally. Ignore any VSAM messages in the JOBLLOG of the job.

SMFU171 ERROR - INVALID vvvv SPECIFIED IN kkkk KEYWORD - NOT xx CHARACTERS IN LENGTH.

The "vvvv" value in the "kkkk" keyword was not the correct length.

SMFU172 OPEN RETRY FAILED FOR VSAM DATA SET TO BE CLEARED. ENSURE DISP=SHR SPECIFIED.**RC=rrrr, ERROR CODE=cccc**

The open retry was NOT successfully done and processing will NOT proceed. The return code 'cccc' was issued by the OPEN SVC and the VSAM open error code 'cccc' was found in the VSAM access control block. The most common error is specifying "DISP=OLD" in the JCL or DDA block. Since SMF always has the data set allocated, a "DISP=SHR" is required. A code of '00A8' indicates the data set was currently open by another user and the system operator replied 'CANCEL' to WTOR message SMFU900. The job step should be re-executed after the data set is no longer in use by someone else. Other error codes are documented in the manual titled "VSAM ADMINISTRATION: MACRO INSTRUCTION REFERENCE" from IBM. In severe situations, it may be necessary to delete and reallocate the SMF file in question.

SMFU173 SMFUTIL NOT APF AUTHORIZED. SMF NOT POSTED. SMF WILL RECOVER DATA SET UPON TIMER WAKEUP. MOVE SMFUTIL TO AUTHORIZED LIBRARY FOR SMF POST TO BE FUNCTIONAL.

SMFUTIL determined that APF authorization was not present in the JSCB. SMFUTIL will not be able to post SMF to recover the cleared data set immediately. SMF will discover the empty data set the next time it wakes up automatically. In a critical SMF shortage condition (all data set full) this could cause additional lost data. It is much quicker to have SMFUTIL authorized to allow SMF to discover the data set sooner.

SMFU175 ABEND DETECTED DURING CLEAR PROCESSING. SMF DATA SET HAS BEEN CLOSED.

SMFUTIL detected an ABEND during clearing an SMF data set. The SMF data set was closed by an ESTAE exit. If the ABEND was internal to SMFUTIL the data set will be usable. Rerun the CLEAR step. If the ABEND was due to an operator cancel (S122 or S222) or of a system timeout (S322), the data set may be damaged. Run an IDCAMS verify on the data set. It may be necessary to run the verify as many as eight times to clear all the status bits in the catalog entry for the data set. If this does not repair the data set, it will be necessary to delete and reallocate it.

SMFU176 PUT FAILED FOR RECORD TO VSAM OUTPUT DATA SET ON DDNAME xxxxxxxx. RC=xxxx, FUNC=xxxx, FDBK=xxxx.

A VSAM data set could not be written to for output. Ensure that the data set is allocated correctly for SMF usage. The FUNC (function) and FDBK (feed back) codes may be found in the VSAM Programmers Guide. The most common error will be a FDBK=0028 which means that the output VSAM data set did not have enough space for the data to be written. Delete and redefine the data set with a larger space allocation.

SMFU180 DDNAME ddd DOES NOT DEFINE A VSAM DATA SET.

The data set defined by the specified DDNAME is not a VSAM data set and is therefore invalid in the VSAMMOD keyword.

SMFU181 CHECKPOINT RECORD NOT DELETED FOR DDNAME ddd. CORRECT I/O ERROR AND RERUN.

An I/O error occurred while writing to a check pointed output MOD data set. The return code will prevent and subsequent CLEAR step from executing. Correct the cause of the I/O error and re-execute the Job.

SMFU182 ddd HAD nnn RECORD PUT FAILURES DUE TO RECORD LENGTH GREATER THAN OUTPUT LRECL.

The specified DDNAME had nnn records that could not be written to it because it had a physical block size and/or record length that was smaller than the length of the record(s) that were to be output. This is applicable to user specified RECFM=VB only. The records were not placed to the specified output file and SMFUTIL continues but sets a minimum return code of 4. When creating output data set, do not specify a DCB and SMFUTIL will set LRECL=32756, BLKSIZE=32760 (for tape), RECFM=VBS. If you must use other parameters, ensure that you specify a record length large enough to handle the longest records. Ensure that the block size is at least 4 bytes longer than the record length for the block descriptor word.

SMFU185 ERROR - SUBTYPE KEYWORD AND SUBTYPE VALUES IN TYPETODD PARMLIST ARE MUTUALLY EXCLUSIVE.

The specification of a SUBTYPE keyword has been detected in the same execution as a TYPETODD keyword that has specific subtype values for a selected record type. These specifications are mutually exclusive and may not both be used within the same execution.

SMFU189 USER userid NOT AUTHORIZED TO USE SMFUTIL. RUN TERMINATED BY INSTALLATION [SECURITY EXIT] [AUTH USER TABLE].

Your site has implemented a security exit or table that is preventing you from executing SMFUTIL. Check with your installation Security Administrator or Systems Programmer for what is required to be authorized for SMFUTIL use.

SMFU190 DISASTER - ERROR IN PARMS RETURNED FROM USER SECURITY EXIT. TERMINATING WITH DUMP.

The site supplied security exit module (SMFUSECX) has returned an invalid buffer to the SMFUTIL program. Ensure that all restriction specifications that are turned on also have a valid buffer returned. The length value must not be 0.

SMFU193 USERx EXIT MODULE modulename CALLING SUSPENDED BY EXIT RETURN CODE REQUEST.

A USER1, USER2 or USER3 exit routine issues a return code indicating it should not be called again. Invocation of the named exit module is suspended and processing continues normally.

SMFU194 ERROR - INVALID x PARAMETER IN y. description.

The 'x' sub-parameter in the 'y' parameter was invalid for the stated reason 'description'.

SMFU195 ERROR - CICS VERSION PARAMETER v IS INVALID.

The 'v' version sub-parameter of the **CICSSPLIT** keyword was not recognized as a valid version identification. See the **CICSSPLIT** keyword syntax for supported versions.

SMFU196 CHECKPOINT RECORD FOUND FOR DDNAME d. BLOCK RESET PERFORMED IN EARLIER RUN - BYPASSED.

A checkpoint record was found for the indicated DDNAME but the record indicated that an earlier restart run had already been attempted. The earlier restart run had returned the output MOD data set for the specified DD to its original condition but the earlier restart run terminated before any new data was again added to the data set. This current restart run is bypassing the block reset function for this data set because it is not required. This is a normal condition if multiple restarts are attempted and terminate with other errors.

SMFU197 STOW ADD/REP FAILED FOR INPUT RECORD CHECKPOINT.

A STOW for the input record checkpoint failed during the CKPTINPT process. Ensure that the checkpoint data set specified on the SYSCKPT DD statement has sufficient directory blocks available and rerun the job. If the problem persists, call for technical support.

SMFU198 DUPLICATED|REGRESSED INPUT DATA DETECTED. CURRENT INPUT DATA HAS ALREADY BEEN PROCESSED. RESTART WITH PROPER INPUT DATA OR DELETE “INPTREC1” CHECKPOINT RECORD AND RERUN.

For DUPLICATED, the CKPTINPT processing feature has detected an input checkpoint record that matches the first input record of this execution. This indicates that the same input data is being processed as was processed in the previous execution. This may indicate a job was rerun in error. Ensure that current execution is not a duplication of a previous execution. If so, do not run the job. If not, delete the INPTREC1 checkpoint record and return the job.

For REGRESSED, the CKPTINPT processing feature has detected an input checkpoint record whose date and time stamp are younger than the date and time stamp of the current execution's first input data record by more than the SORTTOLERANCE amount. This indicates that the current input data may have already been processed in a previous execution. This may indicate that the job is being run with an improper input data set. Ensure that current execution is not a duplication of a previous execution. If so, do not run the job. If not, delete the INPTREC1 checkpoint record and return the job.

SMFU200 ERROR - INVALID v VALUE IN k PARMLIST - LESS THAN n.

The value “v” in keyword “k” was invalid because it was less than “n”

SMFU201 DATA SET dsname ON VOLUME x IS CURRENTLY THE ACTIVE SMF DATA SET. COPY DISALLOWED DUE TO ‘NOACTIVE’ SPECIFICATION.

An attempt was made to read in the active SMF data set when ‘NOACTIVE’ was in effect. This is probably a job sequence error. ‘NOACTIVE’ is used primarily to prevent the active SMF data set from being dumped in a production DUMP/CLEAR run. The CLEAR step would fail and the possibility of copying the same data twice would exist. ‘NOACTIVE’ is used to prevent this. If this is not a DUMP run prior to a clear, remove the “NOACTIVE” specification and rerun. If it is a DUMP and CLEAR run, verify that the job is processing the correct data set and rerun.

SMFU203 ABEND xxx-xx DETECTED DURING DATA COPY. ABEND INFORMATION SNAP FOLLOWS:

A possibly recoverable ABEND has been detected by the ESTAE exit in SMFUTIL. A brief snap of ABEND information follows. Please contact Product Support and give them the information.

SMFU204 ABEND LIMIT EXCEEDED. USER ABEND SCHEDULED TO ABORT EXECUTION.

SMFUTIL has detected multiple ABENDs beyond the ABEND limit threshold. The execution is terminated, reports are generated for data copied to this point, and the run is aborted with a USER 999 ABEND with a DUMP. Any checkpoint records are left in place. Please contact Product Support for assistance.

SMFU205 ABENDS DETECTED - XXX.

ABENDs were detected in the data input copy phase. Message SMFU203 should also appear. Report the problem to the Product Support group.

SMFU206 USER EXIT ABEND DETECTED. USER ABEND SCHEDULED TO ABORT EXECUTION. SPECIFY NOESTAE TO OBTAIN SYSUDUMP.

A user-supplied exit routine has ABENDED. SMFUTIL has intercepted the ABEND and is terminating the run. To debug an exit routine, specify the NOESTAE keyword to disable SMFUTIL's ESTAE and allow a dump to be taken.

SMFU211 ABEND DETECTED. NO SDWA AVAILABLE. RETRY WILL BE ATTEMPTED.

SMFUTIL detected an ABEND but no SDWA (diagnostic work area) was available to describe the ABEND. The region size is probably too small to allow the SDWA to be constructed. A retry will be attempted. Specify a larger region and rerun to obtain diagnostic information. If the problem persists, contact Product Support for assistance.

SMFU214 CHECKPOINT RECORD FOUND FOR DDNAME x. NO DATA ADDED IN PREVIOUS RUN. RESTART SKIPPED.

SMFUTIL encountered a checkpoint record that indicated a 'MOD' disposition data set had been opened in a previous run but had never actually received any output. This is usually caused by an S80A ABEND at open time. A restart on this DDNAME is not required and will be bypassed. The output data set is still in the condition as it was in the original execution.

SMFU216 DSNAMES SUFFIX GENERATION BYPASSED FOR ddname. DSNAMES IS GDG FORMAT.

The DDNAME specified had a data set name structure of a generation data group (GDG). The appending of a generated suffix to this data set name is bypassed and processing continues.

SMFU218 DSNAMES ON DDNAME x IS TOO LONG FOR GENDSN SUFFIX GENERATION. RUN ABORTED.

The DDNAME specified had a data set name that is too long to allow the requested suffix to be added. The execution is abandoned.

SMFU219 SYSCKPT INDICATES PREVIOUS RUN NOT COMPLETED. THIS RUN IS NOT PROPER INPUT DATA SET.

The previous execution of SMFUTIL using this checkpoint data set was not completed successfully. The current execution should be a rerun attempt but the input data set is not the same as the previous execution. Rerun with the proper input data set or delete the "INPTREC2" and/or "INPTREC1" checkpoint records and rerun as is.

SMFU220 EXCESSIVE DATE/TIME GAP DETECTED BETWEEN THIS INPUT AND FINAL RECORD OF LAST RUN. PROBABLE CAUSE IS DUMPING DATA SETS OUT OF SEQUENCE. RERUN WITH CORRECT INPUT DATA SET.

The "ORDERCHK" function detected a time gap between the previous run's final record DATE/TIME stamp and the current run's first record DATE/TIME stamp that was larger than that specified (or defaulted) by "ORDERGAP". The execution is aborted. This usually indicates that a 'SYS1.SMF' was being dumped out of sequence and that another data set contains data prior to the current data set and it should be dumped first. Rerun with the older input data first and then execute with the current data set. This message may also be caused by selecting a value for "ORDERGAP" that is too small or by a system outage that causes a time lag between final record of one SMF data set and the first record of the next man data set. In this case, select a larger value for "ORDERGAP" or specify "NOORDER" to override order sequence checking and rerun.

SMFU222 DDNAME DD FAILED TO OPEN. name FUNCTION NOT AVAILABLE.

The DDNAME specified could not be opened. An output data set required for the requested function "name" is not available so the requested function (i.e. PRINT, BADSNAP, DUPSNAP, etc.) will not be performed. Processing continues.

SMFU225 ERROR - ONLY ONE REGULAR "TIME" SPECIFICATION ALLOWED PER EXECUTION. USE "XTIME", "ITIME", OR "DTRANGE" TO SPECIFY REQUIRED TIME SLICES.

The "TIME" parameter may only be specified once in a run and may only contain one pair of start:end values. The "TIME" parameter is used to select a basis start and end time, not time "slices". The "ITIME" and/or "XTIME" parameters may be used to select time slices to be included or excluded (respectively) from the date range being processed or the "DTRANGE" parameter may be used to select a specific date/time span of data to be selected.

SMFU226 ERROR - RDJFCB FAILED FOR DDNAME ddname DURING RESTART. JCL HAS PROBABLY BEEN CHANGED. RESTART JOB WITH ORIGINAL JCL OR DELETE THE CHECKPOINT RECORD WITH THIS DDNAME.

A restart required checkpoint record was found for the indicated ddname. A RDJFCB macro was issued to test for the presence of the required DDNAME in the JCL and it failed indicating the DDNAME was not present. The current execution JCL is different from the JCL originally used in the prior run that failed. Restart the execution with the correct JCL or delete the checkpoint record with the indicated name from the SYSCKPT PDS data set.

SMFU228 ERROR - MULTIPLE START:END PAIRS NOT ALLOWED IN "TIME" PARAMETER LIST. USE "XTIME", "TIME", OR "DTRANGE" TO SPECIFY REQUIRED TIME SLICES.

The "TIME" parameter may only contain one start:end time specification pair. The "TIME" parameter is used to select a basis start and end time, not time "slices". The "ITIME" and/or "XTIME" parameters may be used to select time slices to be included or excluded (respectively) from the date range being processed or the "DTRANGE" parameter may be used to select a specific date/time span of data to be selected.

SMFU230 ERROR - SMFUTIL ARCHIVE VSAM BASE CLUSTER NAME TOO LONG. CORRECT ARCHIVE DATABASE NAMING AND RESTART.

The archive database cluster name is limited to 31 characters in length. If a longer one is allocated, you must have modified the suffix index and path names to accommodate it. SMFUTIL will not be able to allocate the archive database elements required to process. Rerun the ARCHDEFN job to allocate a cluster name of 31 characters or less.

SMFU231 CHECKPOINT RECORD FOUND FOR DDNAME x AND DATA SET WAS NEWMOD. RESET TO EMPTY.

The previous failed run left a checkpoint record for DDNAME "x". The data set specified as MOD but was really NEW. The data set was reset to empty for the restart.

SMFU232 SEARCH FOR SYSARCH UCB FAILED. CALL TECHNICAL SUPPORT.

The search of the for the UCB for the TIOT entry for SYSARCH failed. This is an internal error. Call SMFUTIL technical support.

SMFU233 ERROR - x BLOCK IN PROCESS. KEYWORD y INVALID IN CURRENT BLOCK TYPE.

A keyword was encountered that was not valid for the block command structure in process. The block must be terminated with an 'x' END statement prior to specifying normal keyword commands or beginning another type of block structure.

SMFU235 ERROR - SYSTEM ID s WAS PREVIOUSLY TARGETED TO DDNAME d. IGNORED.

The specified system id was already targeted to a DDNAME in a previous SIDSPLIT statement. It cannot be re-targeted to another DDNAME via an additional SIDSPLIT statement.

SMFU236 ERROR - ILLEGAL xxxSTART COMMAND. PREVIOUS BLOCK COMMAND MUST BE COMPLETED BEFORE BEGINNING A NEW BLOCK.

A block start command of type xxx was encountered when another block was already in process. The previous block must be terminated with a xxxEND statement prior to beginning a new block.

SMFU238 ERROR - INVALID BLOCK COMMAND xxx. COMMAND BLOCK NOT IN PROCESS. xxxSTART MUST BE SPECIFIED FIRST.

A block type keyword was encountered when a block of the required type was not in process. The xxxSTART keyword must be specified first to begin the required block structure.

SMFU239 ERROR - DDNAME x IS ALREADY ALLOCATED. DDA BLOCK FOR THIS DDNAME IS IGNORED.

The DDNAME specified is already in use, either via a JCL specification or a previous DDA block specification.

SMFU241 USER EXIT WAS IN CONTROL AT TIME OF ABEND.

A user exit routine caused an ABEND. Correct the exit logic.

SMFU242 ARCHIVE RECORD IGNORED: VOLUME=vvvvvv, DSNNAME=dsname, START DATE/TIME UNKNOWN FOR MOD DATA SET.

The data set on the specified volume was disposition MOD but a record did not currently exist in the ARCHIVE database for this volume. SMFUTIL does not have sufficient information to create an archive record for the volume. To archive the existing data on this volume use the ARCHINPT function in a separate job specifying this volume as an input data set.

SMFU244 MEMBER SPECIFICATION HAS INVALID SYNTAX OR CHARACTERS.

The member name specification within the data set name is invalid. A member name specification must be 8 characters or less and begin with an alpha character. It may not contain special characters other than the national characters of \$ and #.

SMFU245 RELATIVE GDG SPECIFICATIONS HAS INVALID CHARACTERS. MUST BE "0" OR +/- FOLLOWED BY NUMERICS.

The specified relative GDG specification in the data set name is invalid. A relative GDG must be all numeric or a '-' or '+' sign followed by numeric.

SMFU246 INVALID ITEM xxx IN DEFAULTS TABLE. [reason].

The defaults member verb 'xxx' is in error. It is either misspelled (unknown verb) or the specified value is incorrect for as defined by the 'reason' text.

**SMFU247 ERROR - DYNAMIC ALLOCATION FAILED FOR DDNAME x WITH DSNNAME y.
SVC 99 RETURN CODE xx, ERROR CODE xx, INFO CODE xxxx.**

The DDA block dynamic allocation failed with the indicated return, error, and information codes. If the DDNAME begins with SYSARCH, ensure that the archive database name component suffixes were not incorrectly modified during creation. For other DD names, consult the IBM manual on SVC 99 return codes for more information as to the reason of the failure.

SMFU250 ERROR - type SUFFIX GENERATION IS NOT ALLOWED UNLESS rrrr.

For data set name suffix generation utilizing a date from the output data, allocation must be deferred until data is available. Change to DEFER in the DDAEND statement. Data set name suffix generation is also invalid for input data sets and output temporary data sets.

SMFU251 ERROR - DDA TEMPORARY DATA SET MUST HAVE DISP(NEW,DELETE,DELETE) SPECIFIED OR DEFAULTED.

If a data set name is omitted (default to temporary) or specified as a temporary data set name structure, the disposition must be as indicated.

SMFU252 ERROR - A DDNAME MUST BE SPECIFIED FOR DYNAMIC DATA SET ALLOCATION.

The DDNAME keyword parameter was omitted from a DDA block definition. A DDNAME is required for allocation. Specify a ddname.

SMFU253 ERROR - PERMANENT DATA SET NAME IS REQUIRED WHEN OLD, MOD, OR SHR IS SPECIFIED AS INITIAL DISPOSITION.

The initial disposition is incompatible with the data set name specified.

- SMFU254 ERROR - DDA TEMPORARY DATA SET MAY NOT BE SPECIFIED FOR INPUT USAGE.**
A DDA block defining a DDNAME of SYSUT1xx must specify a permanent data set with a disposition of OLD or SHR. Temporary data set names are invalid.
- SMFU257 ERROR - DDA AUTOMATIC SPLIT DATA SET GENERATION REQUIRES DSN SUFFIX GENERATION OR RELATIVE GDG.**
A DDAEND keyword indicated that multiple output data set were to be prepared to receive split data but no provision was made for altering the data set name for each data set. Either a relative GDG specification must be indicated or a SUFFIX must be generated to ensure that each created data set is uniquely named.
- SMFU258 ERROR - DDA AUTOMATIC SPLIT DATA SET GENERATION REQUIRES DDNAME OF 5 OR MORE CHARACTERS.**
The DDAEND keyword specified a multiple split request for a user output data set but the DDNAME parameter specified a DDNAME of less than 5 characters. Auto split requires at least 5 characters as the sixth character is overlaid with a unique character to form a unique DDNAME for each output data set.
- SMFU259 ERROR - DDAEND SPLIT SUB-PARAMETER INVALID FOR INPUT (SYSUT1xx) DATA SET.**
An input data set DDNAME (SYSUT1xx) was specified and the DDAEND keyword specified a split request. This is invalid.
- SMFU260 ERROR - DDNAME ddname GENERATED FOR AUTO SPLIT REQUEST IS ALREADY IN JCL OR DDA LIST.**
The DDAEND parameter specified a requested auto split but one or more of the generated DD names were already specified in the JCL or a previous DDA block.
- SMFU261 ERROR - DDNAME IS AUTO GENERATED DDNAME FOR SPLIT OUTPUT. INVALID FOR DIRECTED OUTPUT.**
The specified DDNAME was already generated for receiving split output from a user defined DDNAME via the DDA end keyword. This DDNAME cannot be used as the direct target of a TYPETODD keyword. The primary user DDNAME (specified in the DDNAME keyword within the DDA block structure) should be the target of the TYPETODD keyword.
- SMFU262 ERROR - THE MONTHCAL TABLE SUPPLIED DOES NOT SPAN ALL DATES IN INPUT DATA. ABEND SCHEDULED.**
The MONTHCAL table must contain an entry lower than the lowest data date encountered and an entry higher than the highest data date encountered. Define a table with sufficient entries and re-submit the job.
- SMFU263 ERROR - THE x TABLE y DOES NOT HAVE PROPER STRUCTURE. IGNORED.**
The module loaded as a result of a MONTHCAL, MNTHPCAL or WEEKPCAL keyword specification is improperly structured and is invalid as an SMFUTIL calendar. The load module must be generated via the SMFUMCAL (for MONTHCAL) or SMFUPCAL (for MNTHPCAL or WEEKPCAL) macros in the INSTLIB data set. Ensure the module has been assembled and link edited correctly.
- SMFU266 ERROR - VSAM FUNCTION FAILURE DURING x. FEEDBACK CODE=f. ABEND SCHEDULED.**
A VSAM request block has failed. Call SMFUTIL technical support for assistance.

SMFU267 ERROR - ARCHIVE AND SEARCH FUNCTIONS REQUIRE THE SYSARCH DATABASE TO BE AVAILABLE.

The ARCHIVE and/or ARCHINPT function was requested but the SMFUTIL archive database is not available to the run. Specify the SYSARCH DD card in the JCL or use the ARCHIVDB keyword parameter to inform SMFUTIL of the name of the archive cluster.

SMFU268 ERROR - NO RECORDS FOUND IN ARHIVE DATA SET FOR REQUESTED DATE/TIME VALUES. RUN ABORTED.

A data retrieval request was made but no records were found in the archive to satisfy the request. Ensure that the requested date/time range is valid.

SMFU269 ERROR - DATE REGRESSION DETECTED FOR DDNAME ddn DURING SPLIT CHECKING. DATA IN DATA SET IS OUT OF ORDER.

During a split check, a date/time stamp was encountered older than the previous records date/time stamp. This may indicate that data has been presented out of sequence to SMFUTIL. A split is not taken. This is an informational message only and does not stop the execution. The output data set indicated may need to be reprocessed to remove the out of sequence data to a correct data set for it.

SMFU270 ERROR - SPLIT TYPE MUST BE LOGICALLY GREATER THAN SUFFIX TO BE GENERATED. (I.E. DAY SPLIT CANNOT HAVE MONTH SUFFIX.)

A split function was requested with a suffix generation request that was improper. Performing the requested suffix generation may result in duplicate data set name generation across multiple splits. Specify a suffix generation appropriate for the split requested.

SMFU278 ERROR - INVALID x SPECIFIED IN k KEYWORD - LESS THAN n CHARACTERS.

The “x” parameter in the “k” keyword was too short. It must be at least “n” characters in length.

SMFU279 DUPLICATE KEYWORD SPECIFICATION. KEYWORD k WAS PREVIOUSLY ENCOUNTERED .

The specified keyword was encountered twice in the same run. It may only be specified one time.

SMFU280 SYSARCH DD ALREADY PRESENT IN JCL. ARCHIVDB SPECIFICATION IS NOT ALLOWED.

The ARCHIVDB parameter may not be specified if the SYSARCH DD statement is already in the execution JCL.

SMFU281 ERROR - DDNAME x IS INVALID FOR DDA SPECIFICATION.

The specified DDNAME may not be dynamically allocated (i.e. SYSPRINT, BADSNAP, SYSIN, etc.).

SMFU282 WARNING - AMBIGUOUS TYPETODD REQUESTS. DDNAME d WILL RECEIVE ALL SUBTYPES OF RECORD TYPE n.

A record type of ‘n’ has been targeted to the indicated DDNAME directly and also with subtype specification limitations. This is ambiguous and is ignored. All subtypes of the record will go to the indicated ddname.

SMFU283 WARNING - ARCHINPT REQUEST INVALID FOR INPUT VSAM DATA SET. REQUEST IGNORED.

The ARCHINPT keyword may only be specified when the input data source is not a VSAM data set. For example, you cannot archive the input SMF files.

SMFU285 ERROR - CKPTDB MAY ONLY BE SPECIFIED ONCE PER EXECUTION.

The CKPTDB keyword may only be specified a single time per execution of SMFUTIL.

SMFU286 ERROR - CKPTDB DYNAMIC ALLOCATION FAILED.

The dynamic allocation requested by CKPTDB was not successful. Check the other error messages for the information and reason codes.

SMFU289 ERROR - xxx BLOCK IN PROCESS AT END OF FILE. NO xxxEND STATEMENT ENCOUNTERED.

After the final control statement was processed, SMFUTIL determined that a command block structure of type xxx was not terminated. An xxxEND statement must precede the end of file to terminate the command block.

SMFU291 ERROR - VOLSER MISMATCH IN EOVS EXIT FOR DDNAME d. ARCHIVING HAS BEEN SUSPENDED FOR THIS FILE.

During EOVS (End Of Volume) processing, SMFUTIL compared the volume serial extracted from the JFCB that should be the volume just completed with the volume serial in the UCB being used and they did not match. This is probably a user error. Do not specify multiple units for output data set being archived or for input data set if ARCHINPT is being used. Do not specify a VOLSER list for MOD output data set that contain volumes not yet used (not currently part of the data set). Allow 'PRIVAT' scratch volumes to be used to extend that data set.

SMFU292 ERROR - ARCHINPT IS MUTUALLY EXCLUSIVE WITH IVSTART. RUN ABORTED.

The IVSTART keyword may not be specified in the same execution with the ARCHINPT keyword. Input data set to be archived must be processed in their entirety.

SMFU293 ERROR - DYNAMIC ALLOCATION FAILED DURING CHECKPOINT RESTART. RUN ABORTED.

During a restart of a previously aborted execution, a dynamic allocation failure occurred while attempting to allocate a data set required for the restart. The JCL or input data may have been changed. A restart requires the same JCL and input data as the original failing run. If they have not been altered, note the error codes in the previous messages and refer to the appropriate IBM documentation for information on the failure reason.

SMFU296 ERROR - THE MONTHCAL TABLE SUPPLIED DOES NOT CONTAIN SUFFICIENT DATE SPAN TO PROCESS REQUEST.

The MONTHCAL table supplied by the installation must contain sufficient date points to be able to select data. Ensure that at least two month points exist before and after the month you are attempting to process. Add more elements to the table, if required, reassemble it and rerun the extract with the new MONTHCAL table.

SMFU297 ERROR - RETPD AND EXPDT PARAMETERS ARE MUTUALLY EXCLUSIVE. ONLY ONE MAY BE USED IN A DDA BLOCK.

The RETPD and EXPDT keywords may not both be specified in the same DDA block structure.

SMFU298 WARNING - RECORD GET FAILED FOR SMF DATA SET dsname DURING MANXALOC PROCESSING. DATA SET BYPASSED.

During MANXALOC processing, a get was done against an SMF data set to retrieve the first record for testing date sequence to allocate the SMF data set in. The get failed. The data set is omitted from the run. Call technical support for assistance.

SMFU299 WARNING - OPEN FAILED FOR SMF DATA SET dsname DURING MANXALOC PROCESSING. DATA SET BYPASSED.

During MANXALOC processing, an open was done against an SMF data set. The open failed. The data set is omitted from the run. Call technical support for assistance.

SMFU300 ERROR - MANXALOC AND kkkk ARE MUTUALLY EXCLUSIVE. RUN ABORTED.

The MANXALOC and “kkkk” keywords may not both be specified in the same execution. For example, ARCHINPT may not be specified as VSAM data sets may not be archived and MANXALOC allocates the SMF (VSAM) data set as input data set(s).

SMFU301 ERROR - DUMPCLEARALL INVALID IF OTHER INPUT DATA SETS ARE PRESENT.

If DUMPCLEARALL is specified, no input data set may be supplied by the users. All input data will be determined by DUMPCLEARALL processing by examining the current SMF data sets.

SMFU306 ERROR - DUMPCLEARALL EXECUTION ALREADY IN PROGRESS. THIS EXECUTION ABORTED.

Only one execution of DUMPCLEARALL may be in progress at a time on a system.. The second one started will terminate.

SMFU307 WARNING - DUMPCLEARALL FOUND NO DATA SETS TO DUMP AND CLEAR. THIS EXECUTION TERMINATED.

The automatic SMF switch within DUMPCLEARALL processing failed and no input SMF data sets could be found to process. Call support.

SMFU308 ERROR - DUMPCLEARALL REQUIRES ACCESS TO ALL DATA SETS MARKED DUMP REQUIRED. ABEND SCHEDULED. CALL SUPPORT.

The allocation of an SMF data set failed during DUMPCLEARALL processing. See the associated dynamic allocation failure messages for more information.

SMFU309 ERROR - ARCHIVE PROCESSING FAILURE DURING DUMPCLEARALL EXECUTION. DUMP SUCCESSFUL BUT ARCHIVE NOT COMPLETE.

ARCHIVE processing was in error during the DUMPCLEARALL run. The DUMPCLEARALL processing was successfully done but the ARCHIVE may not properly reflect the content of the output data sets. See associated messages for more information about the ARCHIVE failure.

SMFU310 WARNING - INPTXXX2 CHECKPOINT RECORD RETAINED DUE TO RUN FAILURE. CORRECT ERRORS AND RERUN.

The run has failed. The input data checkpoint record was retained in it alternate format so a restart can be facilitated.

SMFU311 ERROR - GDG CKPT RECORD xxxx WAS NOT FOUND. RUN ABORTED. ENSURE CORRECT GDGCKPTI NAME SPECIFIED.

The requested GDGCKPTI definition was not located in the SYSCKPT data set. Ensure the correct name is being specified and that it coincides with a CKPTGDG specification from a DDA block in a previous run.

SMFU313 WARNING - NO DATA SETS FOUND FOR INDEX gdgindex .

The GDG index level specified in the GDGINPUT parameter did not contain any data sets in the catalog.

SMFU315 GDGCKPTI WARNING - GDG DATA SET IN CKPT RANGE NOT FOUND IN CATALOG. DSN=dsname

A generation of a GDG index being input was not valid. This means the GDG skips a data set. The data set was probably un-cataloged and deleted during a rerun of some type. The data set name is skipped and processing continues with the next generation.

SMFU317 DELETE PROCESSING - DATA SET NOT DELETED DUE TO RETURN CODE OF RUN GREATER THAN x.

The requested disposition of DELETE for a CATINPUT or GDGINPUT dataset was ignored because the return code for the run was greater than the DELETERC maximum return code of “x”. Correct any problems and rerun, if required, or delete the dataset manually.

SMFU318 ERROR - CKPTGDG NOT ALLOWED FOR INPUT DATA SET.

The CKPTGDG DDA block parameter may not be used in conjunction with an input data set definition. It is ignored.

SMFU319 ERROR - CKPTGDG INVALID FOR NON RELATIVE GDG DSNAME STRUCTURE.

The data set name parameter for a DDA block containing a CKPTGDG request must be in the relative data set name format (i.e. GDG.DATA SET(+1)).

SMFU320 ERROR - xxxx PROCESSING REQUIRES SYSCKPT CHECKPOINT DATA SET TO BE AVAILABLE.

The indicated type of processing requested required that SYSCKPT data set be present.

SMFU321 ERROR - GDG INDEX START NOT LOCATABLE IN DSN STRUCTURE DURING CKPTGDG PROCESSING. CALL SUPPORT.

The scan of the GDG data set name failed to locate the GDG suffix. Call support.

SMFU322 ERROR - CKPTGDG ID=xxx INDEX NAME DOES NOT MATCH CURRENT OUTPUT GDG INDEX NAME. RUN ABORTED. CKPT DSN=xxxx DDA DSN=yyyy.

The GDG index name in the existing SYSCKPT CKPTGDG record does not match the current output data set name index. The run is aborted. Ensure the DDA specification has not be incorrectly modified. The existing SYSCKPT record may need to be deleted and the CKPTGDG collection of index levels restarted.

SMFU324 GDG PROCESSING - GDG DATA SET xxxx FAILED. RC=xx, DSN=ddddddddd.

A scratch, locate, or un-catalog for the specified GDG data set failed with the indicated return code.

SMFU326 GDG PROCESSING - GDG CKPT RECORD NOT xxxx DUE TO RETURN CODE OF RUN: DSN=

The indicated function was not performed on the GDG CKPT SYSCKPT record because the run failed with a return code indicator. Correct the problems and rerun.

SMFU328 WARNING - FAST DATE SCAN PROCESSING OF INPUT DATA SKIPPED ALL INPUT DATA BLOCKS. ALL BLOCKS BEGAN WITH DATA LOWER THAN LOWEST REQUIRED DATE/TIME. LAST BLOCK WILL BE USED AS FIRST INPUT.

During initial input scanning of the input data set, no data blocks were found that started with a date/time stamp higher than the lowest data/time value requested before an EOD condition was encountered. If a specific end date AND time value was being selected, it is likely the final block of the data set contained the only required data. Input will begin with the final data block of the data set.

SMFU330 WARNING - FAST DATE SCAN PROCESSING DID NOT SKIP ANY INPUT DATA BLOCKS. LOWEST DATE REQUIRED IS LOWER THAN FIRST DATE FOUND IN FIRST INPUT DATA SET.

During initial input scanning, the first block found contained a date higher than the first required input date. Some required input may be missing.

SMFU335 ERROR - xxxx DSNNAME yyyy NOT FOUND IN CATALOG.

The indicated data set name of type xxxx was not located and is required.

SMFU336 ERROR - MODEL DSNNAME xxxx NOT FOUND ON VOLUME vvvvvv AS INDICATED IN CATALOG.

The model data set name required for GDG process was not on the volume indicated in the catalog.

SMFU337 WARNING - GDG MODEL|LIKE DSNNAME dsn HAS DEFINED DCB VALUES. POOR I/O PERFORMANCE MAY RESULT.

The DCB characteristics associated with the named GDG model or LIKE data set name may cause degraded SMFUTIL performance due to excessive I/O because of small blocking factors.

SMFU338 WARNING - GDG MODEL|LIKE DSNNAME dsn HAS INVALID RECFM VALUE FOR SMF DATA. DCB CHARACTERISTICS WILL BE RESET TO OPTIMUM VALUES.

The specified GDG MODEL or LIKE data set name had existing DCB characteristics that could not be used for SMF data. These are overridden with optimum valid values.

SMFU339 WARNING - USER DEFINED BLKSIZE FOR DDNAME dddddddd IS MUCH SMALLER THAN OPTIMUM. OUTPUT PERFORMANCE WILL BE DEGRADED.

The block size chosen by the user is too small to be efficient. Excessive I/O overhead will be generated. The block size remains as specified.

SMFU344 ERROR - DDNAME d HAS ALREADY BEEN SPECIFIED WITH AN OUTPUT DATE/TIME RANGE. IGNORED.

The specified DDNAME has already been used in conjunction with an output date/time range limitation. Only one output range limitation may be specified for an output DDNAME.

SMFU346 ERROR - INVALID PARM SPECIFIED IN x PARMLIST -

The indicated keyword is an invalid sub-parameter for the 'x' parameter.

SMFU347 ERROR - DDNAME d WAS NOT FOUND FOR PARAMETER kkkk.

The indicated keyword "kkkk" specified a DDNAME of "d" which was not found in the JCL or a DDA block specification. Specify a correct DDNAME or move the DDA block for this data set before the keyword specification in the control stream.

SMFU350 DUMPCLEARALL ACTION - SMF DATA SET ON VOLUME CLEARED IN PREVIOUS RUN. BYPASSED.

A restart of a failed DUMPCLEARALL run is in progress in the CLEAR phase. The indicated data set was successfully cleared in the aborted run and does not need to be cleared again. It is bypassed.

SMFU351 DUMPCLEARALL ERROR: SMF CONTROL BLOCKS NO LONGER AVAILABLE. RUN ABORTED.

The SMCA control blocks could not be located. Ensure SMF is still active on the system.

SMFU352 DUMPCLEARALL CAUTION: SMF DATA SET dsnname ON vvvvvv WAS PARTIALLY CLEARED IN PREVIOUS RUN AND IS NO LONGER MARKED DUMP REQUIRED. SMF HAS CONTROL OF DATA SET.

The previous DUMPCLEARALL run was interrupted during the execution of a clearing process. The SMF system has subsequently marked the partially cleared data set as 'AVAILABLE'. SMFUTIL no longer has access to the data set to complete the clear process. This normally will not cause a problem unless the data set is switched off of AFTER SMF has begun to overwrite the un-cleared area and BEFORE SMF completely fills the data set. This could result in previously dumped data being dumped again. To prevent this, ensure that SMF data sets are normally allowed to fill completely prior to invoking a DUMPCLEARALL run. The recommended method is to use the IEFU29 exit to begin the run and to NOT perform a manual 'I SMF' or a 'Z EOD' shutdown after receiving this message until the current data set is filled.

SMFU354 DUMPCLEARALL ACTION - SMF DATA SET ddd ON vvvvvv IS NO LONGER IN RDS CHAIN. ABORTING.

The SMF SMCA control blocks have changed, probably due to a 'T SMF' command from the operator console or an IPL with a different SMF PARMLIB member. Reset to the correct SMF member or delete the \$DUMPALL SYSCKPT record and the SYSDCKPT data set.

SMFU355 DUMPCLEARALL ACTION - SMF DATA SET ddd ON vvvvvv HAS HAD CLEAR RESTARTED.

The clear of the specified MANx data set was restarted in the current execution.

SMFU356 DUMPCLEARALL ERROR: DELETE OF DUMPCLEARALL CKPT DATA SET FAILED. RUN ABORTED. CALL SUPPORT.

The SYSDCKPT data set could not be deleted. Correct the problem and rerun the job.

SMFU357 ERROR: CLEAR AND DUMPCLEARALL ARE MUTUALLY EXCLUSIVE. RUN ABORTED.

CLEAR and DUMPCLEARALL may not both be specified in the same execution. DUMPCLEARALL implies a CLEAR anyway so if you really want to do a DUMPCLEARALL run, remove the CLEAR statement and rerun the job.

SMFU358 ERROR - SYSDCKPT DATA SET COULD NOT BE rrrr. RUN ABORTED.

The specified operation 'rrrr' could not be performed on the SYSDCKPT data set. Check the SYSLOG for additional messages and correct any errors causing the problem and re-submit the job.

SMFU359 DUMPCLEARALL "\$DUMPALL" SYSCKPT RECORD FOUND. DUMPCLEARALL RESTART IN PROGRESS.

A previous DUMPCLEARALL run was interrupted and is being restarted automatically.

SMFU360 DUMPCLEARALL RESTART - PREVIOUS RUN HAD STARTED|COMPLETED xxxx PHASE.

This message is used to indicated the progress that the interrupted DUMPCLEARALL run has made. The restart will begin at the appropriate point.

SMFU361 DUMPCLEARALL RESTART - PREVIOUS RUN COMPLETED SUCCESSFULLY. RESTART BYPASSED.

The previous run has completed successfully but the \$DUMPALL checkpoint record was not successfully deleted. The record is delete and the restart is bypassed.

SMFU362 DUMPCLEARALL RESTART - RESTARTING DUMP|CLEAR PHASE.

This indicates where the interrupted DUMPCLEARALL will be restarted. Note that if the DUMP phase is restarted and additional SMF data sets are now available, they will be dumped also. If the CLEAR phase is restarted, it will be performed and than a DUMP phase will be performed followed by an additional CLEAR phase.

SMFU363 DUMPCLEARALL WARNING - MOD RESTART RECORD REMOVED FOR DDNAME ddd. SEE EXPLANATION IN DOCUMENTATION.

The interrupted DUMPCLEARALL run had completed the DUMP phase and but all output data sets may not have been successfully closed. MOD restart is no longer possible as some of the MOD restart records may have been deleted. Processing proceeds as if the previous run's DUMP phase was completed and the restart begins with the CLEAR phase. A small amount of output data may be missing from one or more output MOD data sets. This data was contained in the final output buffers for the data sets that were not closed successfully during the interrupted run. The actual amount of data that is lost is dependent on the number of output buffers associated with the DD. If the previous run was interrupted via an ABEND, the data is probably complete as TASK termination would have successfully completed outstanding I/O's. Data will probably only be missing if the interrupted run was because of a processor or system failure (i.e. hardware failure of the CPU). Note that the data is only 'lost' for the indicated ddname. It most likely is still present on other data sets and can be recovered. If you need more information or assistance please call technical support.

SMFU364 ERROR - COULD NOT READ rrrr SYSCKPT RECORD. RUN ABORTED. CALL SUPPORT.

A BLDL macro for the indicated SYSCKPT record failed. Call support.

SMFU365 STOW FAILURE DURING CKPT REPLACE FOR RECORD rrrr. RUN ABORTED.

A STOW macro for the indicated SYSCKPT record failed during a replace. Ensure sufficient space exists in the SYSCKPT data set directory. Call support.

SMFU367 DUMPCLEARALL ERROR - SWITCH OF SMF FAILED. RUN ABORTED BY OPERATOR.

The system operator replied 'ABORT' to WTOR message SMFU902. Check SYSLOG to see why the SMF switch requested by the DUMPCLEARALL run failed. Restart the job.

SMFU368 DUMPCLEARALL WARNING - MORE THAN 36 SMF DATA SETS ARE MARKED "DUMP REQUIRED". THIS RUN WILL PROCESS THE FIRST 36 FOUND.

DUMPCLEARALL processing is limited to 36 data sets. The remainder will be processed in the next run.

SMFU369 DUMPCLEARALL ERROR - CLEAR PHASE BYPASSED DUE TO RETURN CODE DURING DUMP PHASE.

The DUMP phase ended with a return code indicating an error. Check for messages indicating the problem. Correct and rerun the job.

SMFU370 DUMPCLEARALL ERROR - NO DATA SETS MARKED DUMP REQUIRED AND NONE ARE AVAILABLE TO ACCEPT SWITCH. RUN ABORTED.

SMF is currently processing with only one data set (the active one) and no other data set are currently valid. Switch SMF processing to a different set of data sets and then manually dump the data set just that was active.

SMFU371 DUMPCLEARALL WARNING - PRIMARY SMF DATA SET IS CURRENTLY ACTIVE AND ALL OTHERS ARE MARKED DUMP REQUIRED. UNABLE TO LEAVE PRIMARY AS AVAILABLE IN CASE OF "Z EOD" AND SUBSEQUENT IPL.

If the MAN1LEAV defaults parameter is set to "YES", SMFUTIL will attempt to leave the primary SMF data set in an available condition after DUMPCLEARALL processing has completed. This is done by ensuring that the primary is marked "DUMP REQUIRED" by issuing up to two "I SMF" commands to cause SMF switching. If the primary SMF data set is currently active (receiving SMF data) when DUMPCLEARALL begins processing and all other SMF data sets are already marked "DUMP REQUIRED", SMFUTIL cannot honor the MAN1LEAV specification because no data set is available to accept processing after a switch. DUMPCLEARALL processing proceeds and ignores the MAN1LEAV specification. It will be honored at the next run. When the current run completes, you may run an additional DUMPCLEARALL execution (restart the same job) to cause the primary data set to be switched off of and dumped and cleared.

SMFU373 DUMPCLEARALL RESTART WARNING - PREVIOUS RUN DID NOT COMPLETE CLOSE PROCESS FOR OUTPUT DATA SETS. MOD RESTART NOT POSSIBLE. SOME OUTPUT FILES MAY BMISSING A SMALL AMOUNT OF DATA. RESTARTING WITH CLOSE PHASE.

After DUMPCLEARALL processing has completed the data movement phase it closes all output data sets. During close processing each MOD data set has it's checkpoint record deleted. If the run aborts prior to completion of the close processing, some data sets may not be successfully closed and thus may be missing a small amount of data remaining in the final output buffer. A run restart is not possible as one or more MOD data set checkpoint records may have already been deleted. Restart processing continues with close and clear.

SMFU375 ERROR - appl VERSION NOT FOUND IN applTABL FOR process. CALL SUPPORT.

The version identification could not be made for application "appl" (i.e. DB2 or CICS) during process "process" (i.e. DB2SPLIT or CICSSPLIT). The table "applTABL" is used to determine what version of the application produced the current record. This is required to correctly locate the started task name or applid so that a correct split can be done. If a new version of the application has recently been implemented, call technical support for assistance in updating the table to support it.

SMFU377 ERROR - PROCESS NAME pppppppppppppppp DOES NOT MATCH CHECKPOINT \$PROCESS NAME cccccccccccccc. RUN ABORTED.

The \$PROCESS checkpoint record already in the checkpoint data set did not match this run's process name (from PROCESS keyword or TIOCSSTEP field in TIOT). Ensure this checkpoint data set is valid for this usage. If not, specify the correct checkpoint data set. The \$PROCESS checkpoint is a checkpoint usage validation check. It is designed to ensure a checkpoint data set is used for one and only one process. If this is the correct checkpoint data set but the step name was changed, delete the \$PROCESS checkpoint record and rerun the job.

SMFU378 ERROR - PROCESS SYSTEM ssss DOES NOT MATCH CHECKPOINT \$PROCESS SYSTEM NAME cccc. RUN ABORTED.

The \$PROCESS checkpoint record already in the checkpoint data set did not match this run's system name (from SMCA control block or PROC SYS keyword). Ensure this checkpoint data set is valid for this usage. If not, specify the correct checkpoint data set. The \$PROCESS checkpoint is a checkpoint usage validation check. It is designed to ensure a checkpoint data set is used for one and only one process. If this is the correct checkpoint data set but the step name was changed, delete the \$PROCESS checkpoint record and rerun the job. If a single process may be run on more than one CPU, you must specify the PROC SYS keyword to inform SMFUTIL that this is a CPU independent process.

SMFU379 WARNING - NOTIFY OF USER uuuuuuuu SKIPPED. ENVIRONMENT NOT AUTHORIZED.

SMFUTIL must be in an authorized environment for the NOTIFY function for TSO Userid's to be usable. Move SMFUTIL to a LNK LSTxx data set or place the STEPLIB data set into the IEAAPFxx list.

SMFU380 WARNING - PROC SYS pppp INVALID FOR DUMPCLEARALL RUN. CHANGED TO ssss.

The PROC SYS keyword may not be specified for a DUMPCLEARALL execution. DUMPCLEARALL, by its very nature, is CPU specific. The SMF system ID must be used for the DUMPCLEARALL run to ensure this checkpoint data set is not used by another system.

SMFU381 ARCHIVE VOLSER vvvvvv FOR DATA SET dddddddd OVERRIDDEN BY CATALOG VOLSER ccccc.

The DASD data set 'ddddddd' from the SMFUTIL archive is no longer on volume 'vvvvvv' as indicated by the SMFUTIL archive. The system catalog not indicates it is on volume 'cccccc'. SMFUTIL overrides its archive volume indication and uses the system catalog under the assumption that HSM (or an OEM product) has archived and recalled the data set to a different volume.

SMFU384 NON-TARGETED CICS APPLID(S) - applname

CICSSPLIT was specified. This message is a list of one or more CICS applid names that were found in the input stream but were not specifically directed to a user data set. The records containing these applid's have been written to the standard output data sets (i.e. SYSUT2/SYSUT2D).

SMFU385 NOTIFICATION OF RUN FAILURE SENT TO userid/SYSTEM OPERATOR.

A notification of the run failure was sent to the requested TSO user or the System Operator as specified in the NOTIFY keyword.

SMFU395 DAY/WEEK/MONTH TYPE SPLIT BOUNDARY ENCOUNTERED DURING OUTPUT FOR DDNAME ddn. NO ADDITIONAL SPLIT DATA SET AVAILABLE. DDNAME WILL CONTINUE TO RECEIVE DATA.

The indicated DDNAME had more split boundaries than were allowed for. Unless this was intentional, you should increase the split number in the DDAEND keyword for the specified DDNAME to allow for more splits to occur.

- SMFU397 EXECUTION CPU xxxxxxxx NOT FOUND IN LICENSE TABLE. PRODUCT USAGE APPROVED BASED ON SITE LICENSE.**
- The indicated processor serial number was not found in the table of licensed processors. The execution continues on the basis of the site license. You should notify ASPG Contract Administration as soon as possible as to the change in processor serial number.
- SMFU398 prcstype PROCESSING FOUND NO DATA SET NAMES UNDER INDEX iiiiii. PREFIX IGNORED.**
- The specified index level had no data set under it in the catalog. No input data for this index will be read in.
- SMFU401 prcstype PROCESSING FOUND ONLY 1 DATA SET NAME UNDER INDEX iiiiii BUT SKIPLAST SPECIFIED. PREFIX IGNORED.**
- SKIPLAST was specified for the index but only one data set was found under it. No input data will be read for this index level.
- SMFU404 INPUT DDNAME ddn WITH DSNAME dsn BYPASSED BY USER SPECIFIED OPEN EXIT MODULE.**
- The user open exit routine requested that the input data set “dsn” associated with the input DDNAME “ddn” be skipped.
- SMFU405 ALL INPUT DATA SETS BYPASSED BY OPEN EXIT.**
- The user open exit routine skipped all available input data sets. Run was halted.
- SMFU406 ALL INPUT DATA SETS BYPASSED BY OPEN EXIT.**
- The user open exit skipped all input data sets. No input data was available to process.
- SMFU407 ERROR - CONSECUTIVE ERROR LIMIT REACHED ON INPUT DATA SET. EXECUTION TERMINATED.**
- The IOELIMIT error limit was reached. The run is aborted.
- SMFU408 LICENSE INFORMATION - PRODUCT =**
CODE =
EXPIRATION DATE = XXXX.XXX (X/XX/XXXX)
CPU # =
- This information is listed as a result of the LISTLICENSE keyword and shows current license information.
- SMFU413 RESTART CLEANUP IN PROCESS FOR DATA SETS CREATED IN PREVIOUS RUN.**
- A previously aborted run is being restarted. The data sets created in the aborted run are being deleted and uncataloged by the restart.
- SMFU414 DATA SET dsn ON VOLUME volser HAS BEEN DELETED / UNCATALOGED .**
- The listed data set on the specified volume and been deleted and/or uncataloged as part of the restart cleanup.
- SMFU415 RESTART CLEANUP COMPLETE FOR DATA SETS CREATED IN PREVIOUS RUN.**
- The restart is complete and all data sets created in the previous run may now be recreated by this run.
- SMFU416 NEW DATA SET CHECKPOINT RECORD rrr FOR DSNAME dsn KEPT DUE TO RUN FAILURE. CORRECT PROBLEMS AND RESTART RUN. IF IT STILL EXISTS, THIS DATA SET WILL SCRATCHED AND UNCATALOGED UPON RESTART OF RUN.**
- The listed data set is retained because the run failed.

- SMFU418 ERROR: RELATIVE GDG “xx” WAS NOT VALID FOR DATA SET NAME dsn. VERIFY THIS IS A GDG BASE.**
The specified data set name “dsn” contained a relative GDG number (i.e. “+1”) but it does not appear to be a GDG type data set name index in the system catalog.
- SMFU419 WARNING: KEYWORD kkkk SHOULD BE SPECIFIED BEFORE ANY oooo ARE SPECIFIED.**
The keyword “kkkk” must be specified in the control card set before the “oooo” specifications. Reorder the control card set.
- SMFU420 NOTE: RELATIVE GDG “0” DOES NOT CURRENTLY EXIST FOR GDG BASE . CHANGED TO “+1” AND DISP=NEW.**
The specified relative index of “0” does not currently exist for this GDG base. The index is reset to “+1” and the disposition of MOD is changed to NEW.
- SMFU430 ARCHIVE BACKUP BYPASSED BECAUSE ALLOCATION FAILED FOR SEQUENTIAL BACKUP DATA SET.**
The backup data set could not be allocated for the archive backup function. The backup is not taken.
- SMFU433 ARCHIVE BACKUP BYPASSED BECAUSE SUFFIX COULD NOT BE ATTACHED TO BACKUP DATA SET NAME.**
The specified backup data set name is too long to attach the required date suffix. Shorten the dsname.
- SMFU434 ARCHIVE UPDATE PROTECTION IS NOT AVAILABLE BECAUSE SMF RECORD NUMBER NOT DEFINED.**
The SMFUDFLT module specified “0” for the SMF record number to be used to record archive update protection records. This disables update protection. If backup protection is required, change the value to a valid record type between 129 and 255 that is reserved for the exclusive use of SMFUTIL and then reinstall the product.
- SMFU435 ARCHIVE RECOVERY REQUESTED. ALL ARCHIVE UPDATE IMAGES FOUND WILL BE APPLIED TO ARCHIVE DATABASE.**
ARCHIVERECOVER was specified and the ARCHIVE database is being recovered using input SMF records of previous updates.
- SMFU436 ARCHIVERECOVER IS MUTUALLY EXCLUSIVE WITH ANY DATA MOVEMENT AND ARCHINPT. SIMULATE REQUIRED.**
ARCHIVERECOVER is a special function process. No other data movement is allowed as only the recovery records will be processed.
- SMFU438 USER uid NOT AUTHORIZED TO USE KEYWORD kkkk. EXECUTION ABORTED.**
The specified userid “uid” was not found in the authorized users list for the restricted keyword “kkkk”. Consult your SMF administrator for authorization to use the indicated keyword.
- SMFU439 DSNNAME “dsn” FOR DDNAME ddname CONTAINS AN INVALID SUBSTITUTION PARAMETER. CHECK THE SYNTAX.**
The specified data set name contains an invalid or undefined substitution parameter that begins with “&”. The text that follows is not recognizable as a valid substitution string for a date overlay or system identifier overlay. Ensure the correct string is specified.

SMFU440 DSNNAME “dsn” ON VOLUME volser FAILED TO OPEN IN UPDATE MODE FOR SHORTCLEAR PROCESSING. RETRYING IN SEQUENTIAL MODE.

The indicated MANx data set was to be cleared using the SHORTCLEAR processing method but the VSAM open for “UPD” failed with a feedback code of x'E0' indicating the data set had never been initialized. “UPD” is not valid for initial loading of a data set after allocation via IDCAMS. Processing continues by disabling SHORTCLEAR and retrying the open in “SEQ” mode. Clear processing should then proceed normally. Specify NOSHORTCLEAR when attempting to clear a new SMF MANx data set prior to placing it into production.

SMFU446 ERROR – “ALLOCATE” NOT ALLOWED WHEN SYMBOLIC DATE VALUES ARE PRESENT IN THE DSNNAME.

As specified, the symbolic date string (i.e. &YYYYDD) is mutually exclusive with the request for immediate allocation of the data set(s). This would result in multiple allocations with the same dsnname.

SMFU447 WARNING - FUTURE DATED RECORDS HAVE BEEN DISCARDED AS REQUESTED VIA SKIPHIDT.

The SKIPHIGHDATE keyword was specified and future dated records were found and discarded as requested.

SMFU453 DMPCLALL - CLEAR PHASE BYPASSED DUE TO "OLIMIT" SPECIFICATION FOR MOD RESTART.

The OUTPUTLIMIT keyword was specified during a DUMPCLEARALL restart execution. The CLEAR phase has been bypassed for safety as this is probably an interim restart step to reset the output datasets only. Ensure restart succeeded then remove the OLIMIT keyword and proceed with normal operations or other recovery efforts as required.

SMFU454 NOTE: rename CHECKPOINT RECORD RETAINED DUE TO "OLIMIT" RESTART.'

The OUTPUTLIMIT keyword was specified during a DUMPCLEARALL restart execution. The specified checkpoint record was kept for the future restart phase as this is probably an interim restart step to reset the output MOD datasets only. Ensure restart succeeded then remove the OLIMIT keyword and proceed with normal operations or other recovery efforts as required.

SMFU455 ERROR: DUMPCLEARALL AND OLIMIT GREATER THAN ZERO ARE MUTUALLY EXCLUSIVE. RUN ABORTED.

An OUTPUTLIMIT greater than zero (0) may not be specified during a DUMPCLEARALL restart. The OUTPUTLIMIT(0) is only used to perform an interim restart to return the MOD output datasets to their original conditions prior to other recovery efforts.

SMFU457 CHECKPOINT RECORD rename FOR DATASET dsnname ON VOLUME volser HAS BEEN REMOVED. DATASET COMPLETED IN PREVIOUS RUN.

The specified dataset is complete and will be retained. The restart deletion checkpoint record for it has been removed for safety.

SMFU458 WARNING: SYSLOG DD PRESENT BUT SYSLOG DCB OPEN FAILED. SYSLOG IGNORED.'

Ensure the SYSLOG DD dataset has valid DCB characteristics for an output SYSPRINT type dataset.

SMFU459 DSSPACE VALUE IS ZERO. CALL SUPPORT.

The calculated value of the available space in the MANx vsam cluster is zero. This is an internal error and required SMFUTIL support assistance to resolve.

SMFU460 CISIZE VALUE IS ZERO. CALL SUPPORT.

. The calculated value of the control interval size of the MANx vsam cluster is zero. This is an internal error and required SMFUTIL support assistance to resolve.

SMFU461 ERROR: DDNAME d INVALID IN k KEYWORD.

The DDNAME 'd' in keyword 'k' is invalid as used. Refer to the appropriate keyword documentation for valid values.

SMFU462 ERROR: TOO MANY POSSIBLE SPLITS REQUESTED FOR DDNAME d. DDNAME SUFFIX TABLE EXHAUSTED.**SHORTED DDNAME TO 6 CHARACTERS.**

More than the allowable number of splits have been requested in the DDA block DDAEND statement for a DDNAME that is 7 or 8 characters in length. Reduce the number requested or shorted the DDNAME to 6 characters or fewer so 2 digit suffixes may be used.

SMFU467 ERROR: CLEAR TASK FAILED FOR DDNAME d. RETURN CODE WAS r.

An asynchronous CLEAR task failed with the return code of 'r'. Check for further messages as to the reason for the failure.

SMFU468 ERROR: ONE OR MORE CLEAR TASKS FAILED. ABEND SCHEDULED.

One or more of the CLEAR tasks has failed. An ABEND will be scheduled to abort the current execution with a DUMP.

SMFU74 DATASET d USAGE UNKNOWN BECAUSE CLEAR RESTART IN PROCESS. FULL CLEAR WILL BE DONE.

During a DUMPCLEARALL restart, a CLEAR is being done for a MANx dataset dumped in the previous execution. Since the block count is not available, a full clear will be done.

SMFU475 THE KEYWORD k IS NO LONGER SUPPORTED. REFER TO THE CURRENT DOCUMENTATION.

The specified keyword 'k' is no longer supported. Check the current documentation for a replacement.

SMFU476 ERROR - SKIPLAST SPECIFIED FOR GDGCKPTI NAME g BUT ONLY ONE GENERATION IS PRESENT. RUN ABORTED.

The SKIPLAST sub-parameter was specified in the GDGCKPTI keyword for the GDG index 'g' but only a single generation of the dataset is present. There are no previous generations to read so the run is aborted.

SMFU481 WARNING - SMF ID "x" IN RECORD DOES NOT MATCH PROCESSING ID "y".

The SMFIDCHECK function has encountered an SMF record that contains an SMF ID of 'x' but the \$PROCESS checkpoint record contains an id of 'y'. The return code will be set according to that defined for the SMFIDCHECK function.

SMFU482 ERROR - SYSTEM SMF ID "x" DOES NOT MATCH CHECKPOINT \$PROCESS SMF ID "y" FOR DUMPCLEARALL RUN. RUN ABORTED.

The SMFIDCHECK function has encountered an execution system SMF ID of 'x' but the \$PROCESS checkpoint record contains an id of 'y'. The checkpoint dataset is being used for the wrong system. The run is aborted. Define the correct checkpoint dataset or delete the \$PROCESS record and restart.

SMFU484 NOTE - NO DATA DATE AVAILABLE. SUFFIXES WILL USE CURRENT SYSTEM DATE.'

One or more output dataset(s) with no output data directed it requires allocation because of a CREATEU0 or CREATES0 request. Any suffix requiring date insertion or date overlay will be created using the current system (execution) date). This message is produced if the CRECDATE keyword in the SMFUDFLT module is set to 'NO' or if no input data was processed at all and thus no data date is available.

SMFU485 NOTE - RETURN CODE IS GREATER THAN 8. NO EMPTY DATASET CREATION WILL BE PERFORMED.'

When a run completes with a return code of more than 8, any CREATEU0 or CREATES0 requests will be ignored and empty output datasets will not be created.

SMFU486 ERROR - GETMAIN FAILURE IN MODULE modname DURING REQUEST FOR STORAGE. RUN ABORTED.'

There was insufficient private area region available for the execution of SMFUTIL. Increase the REGION parameter on the EXEC card. "REGION=0M" is highly recommended.

SMFU490 ERROR - SPECIFIED VOLSER vser1 DOES NOT MATCH EXISTING CATALOG VOLSER OF vser2.

The user supplied a volume serial list of one or more volumes for a MOD or OLD disk dataset that is currently cataloged. The volume serial "vser1" does not match the corresponding volume serial "vser2" currently in the cataloged volume list for the dataset. If you are trying to extend a disk dataset onto additional volumes, you must specify all existing volumes the dataset resides on plus any new volume required for the dataset. Before re-cataloging the dataset, ensure the dataset physically exists on each of the newly specified volumes.

SMFU491 ERROR - SPECIFIED VOLSER LIST TOO SHORT. DOES NOT CONTAIN EXISTING CATALOGED VOLUME vser.

The user supplied a volume serial list of one or more volumes for a MOD or OLD disk dataset that is currently cataloged. The supplied volume list does not contain all the volumes that are contained in the catalog entry for the dataset. The displayed volume serial "vser" is the next volume in the catalog list. Re-specify the VOLUME list for the dataset, including all volumes that the dataset currently resides on. If the catalog entry contains volumes the dataset does not reside on, it must be corrected before the job is restarted.

The following messages are issued in the JOBLOG or SYSLOG or on the Operator's console:

SMFU777 FAILED SCRATCH IS BEING RETRIED. DSN=dsname

A scratch request for the indicated data set failed with a return code indicating the scratch could be retried. After waiting the prescribed amount of time as specified in the SCRATCHRETRY keyword or the SCRWAIT parameter in the SMFUDFLT defaults module, the scratch request is attempted again. This process continues until the scratch succeeds or the retry limit is reached (SCRETRY value). If the retry limit is reached the scratch attempt is abandoned and a minimum return of 8 is issued.

Note that and IBM message of IEC614I is issued in the JOBLOG for each failed scratch attempt. There is no way for SMFUTIL to suppress this message. The SMFU777 and SMFU778 messages WTO messages in the JOBLOG are intended to explain what is happening. If the SMFU778 message is issued the IEC614I message can be safely ignored.

SMFU778 SRATCH RETRY WAS SUCCESSFUL.

The retried scratch request was successfully completed. Normal processing continues.

SMFU779 FAILED SCRATCH DATA SET IS NO LONGER CATALOGED DURING SCRATCH RETRY. USER DELETION ASSUMED.

The data set for which a failed scratch was to be retried was no longer found in the catalog. It is assumed that the data set was deleted manually by a user. The error is ignored and normal processing continues.

SMFU900 OPEN OF SMF MAN DATA SET FOR CLEAR PROCESSING HAS FAILED. DATA SET IN USE BY ANOTHER USER.**REPLY “R” TO RETRY OR “CANCEL” TO ABORT EXECUTION.**

During CLEAR processing, SMFUTIL must open the SMF SYS1.SMF data set for “CNV,RST”. This means Control Interval Processing with Reset requested. VSAM only allows this type of open if no other users have the data set open at the time. During the open, VSAM returned a x'A8' error code to SMFUTIL indicating another user currently had the data set open. This can happen if another user had the data set open for sequential input. An example of this would be a security officer doing ad-hoc reporting against the SMF MAN files. When this return code is returned by VSAM, SMFUTIL will wait briefly and then retry the open. It continues to do this for approximately 5 minutes. At the end of this time, this message is issued and a reply from the system operator is required. A reply of “R” will cause SMFUTIL to continue to try the open for an additional 5 minutes. A reply of “CANCEL” will terminate the execution with a return code indicating clear failure. Determine what user has the data set open, instruct them to free it and reply “R” to instruct SMFUTIL to retry the open process. Reply “CANCEL” only if instructed to do so by site technical support.

SMFU901 CLEAR OF MAN DATA SET HAS FAILED. CALL TECH SUPPORT.

The SMF MAN data set to be cleared could not be opened properly. The execution is terminated and the return code for the run is the VSAM return code from the open failure.

SMFU902 SWITCH OF ACTIVE SMF DATA SET HAS NOT OCCURED WITHIN TIME LIMIT. REPLY “WAIT” TO RETRY WAIT, “GO” IF SWITCH HAS OCCURED, OR “ABORT” TO ABORT EXECUTION.

The DUMPCLEARALL run has issued a ‘I SMF’ switch request and is waiting for it to complete. SMF has not responded within the time limit. If the system is very busy, reply ‘WAIT’ to continue the wait for switch completion. If you see that the switch has physically occurred, reply ‘GO’ to bypass the wait. Reply ‘ABORT’ to kill the execution only if there is a problem with the SMF system and SMFUTIL DUMPCLEARALL should not continue.

SMFU910 SMFUTIL J=jobname P=processname, S=smfid COMPLETED/FAILED(?) RC=xx.

This message is sent to the system operator and/or TSO users as requested via the NOTIFY parameter. Return codes from 0 to 3 are considered normal completion's. These messages will not be sent to NOTIFY user ID's flagged with the '(F)' parameter. Return codes 4 and up are considered failed runs and are so flagged in the message. The message text field 'jobname' is the name the job is running under, 'processname' is the process name taken from the TIOCTSTEP field in the TIOT or supplied via the PROCESS keyword, and 'smfid' is the executing system SMF identification taken from the SMCA control block or supplied via the PROCSYS keyword.

SMFU920 SMFUTIL J=jobname P=processname, S=smfid FAILED, ABND CODE xxxx.

This message is sent to the system operator and/or TSO users as requested via the NOTIFY parameter. It indicates the identified run has ended abnormally with ABEND code 'xxxx'. The message text field 'jobname' is the name the job is running under, 'processname' is the process name taken from the TIOCTSTEP field in the TIOT or supplied via the PROCESS keyword, and 'smfid' is the executing system SMF identification taken from the SMCA control block or supplied via the PROCSYS keyword.

SMFU925 VSAM FAILURE DURING ARCHIVE PROCESSING.

This message is sent to the system operator to the JOBLOG. It indicates that the SMFUTIL execution ended successfully but the archive updates were not completed. Examine the SMFUTIL SYSPRINT output for messages indicating the reason code for the VSAM failure. An example would be if there were insufficient space in the ARCHIVE base cluster to add a new record. Correct the problem. **DO NOT RERUN THE JOB THAT ISSUED THE MESSAGE.** It successfully copied all the data required. One or more output data sets will not be properly reflected in the ARCHIVE database. Run an SMFUTIL ARCHIVEINPUT execution against all volumes that have data added to them in the previous run.

SMFU999 OPEN FAILED FOR ddname.

The DDNAME specified could not be opened. This message is written to the JOBLOG via a WTP macro and is usually for SYSPRINT. The SYSPRINT output data set is required.

16. APPENDIX B - EXECUTION WARNING/INFORMATION MESSAGES

The following messages are not numbered. They are informational in nature and are produced as part of the EXECUTION LOG of processing for each run of SMFUTIL. Most messages are printed only when they have data to display. The messages are presented here in alphabetical order with the exception of the first one which start with a variable text field defined in the message.

DDNAME CONTAINS n RECORDS FOR DATES ccyy.ddd hhmmss THRU ccyy.ddd hhmmss.

This is the number of records written to the specified DDNAME and spanning the indicated times.

DDNAME CONTAINS NO OUTPUT. CREATEU0 SPECIFIED. DATA SET IS EMPTY.

The specified DDNAME was opened due to the CREATEU0 request but the created data set is empty.

DDNAME CONTAINS NO OUTPUT. CREATES0 SPECIFIED. DATA SET IS EMPTY.

The specified DDNAME was opened due to the CREATES0 request but the created data set is empty.

DDNAME RECEIVED NO OUTPUT DUE TO OUTPUTLIMIT(0).

The DDNAME specified had no data written to it because 'OUTPUTLIMIT(0)' was specified. This was probably a reset execution for 'MOD' disposition data set.

**xxxxxxx TYPE SPLIT BOUNDARY ENCOUNTERED DURING OUTPUT FOR DDNAME dddddddd.
DDNAME sssssss NOW RECEIVING DATA.**

The requested split was performed. A split boundary was detected while output to DDNAME 'ddddddd' was in process. The previous file is placed in idle status and output has begun to the split data set on DDNAME 'sssssss'. If SYSCKPT is not present, the previous file is closed (and deallocated if DDA defined). If SYSCKPT is present, the previous file is retained in an open state until the entire run is completed.

ALL OUTPUT VSAM DATA SETS WILL BE EXTENDED (MOD).

The VSAMMOD keyword was specified without a parameter list. All VSAM output data set (SYSUTx and/or user ddnames) will be extended. This means that data from this copy execution will be added to the end of the data set as if a MOD disposition had been specified.

ARCHIVE RECORD action: VOLUME=vvvvvv, FILESEQ=XXXX, VOLSEQ=XXXX,

DSNAME=ddddddd

LOW DT=YYYY.DDD-HH:MM:SS:TH, HIGH DT=YYYY.DDD-HH:MM:SS:TH, SETNAME=X

BLOCK COUNT= RECORD COUNT=

These messages indicate the action taken in the archive database as a result of an ARCHIVE, ARCHINPT, or ARCHUSER keyword request.

BLOCKS IN - n

This the total number of physical blocks read from the input data set(s).

BLOCKS OUT - n

The total physical blocks written is indicated.

**COPY COMPLETED FOR DATE/TIME RANGE SPECIFIED. DATA IS SORTED.
END OF DATA ASSUMED BEFORE END OF FILE.**

The requested date and/or time range has been copied. Since the data is in sorted order, or SHORTCPY was specified, SMFUTIL will terminate early because all requested data has been copied. If SHORTCPY or CHKISORT was specified and no data was copied, the first input record encountered was past the date/time specification requested. If IVSTART was used, specify a lower value for it to access an earlier volume, otherwise, the data set may not contain the requested data. Delete the CHKISORT specification or specify GOTOEOF to force SMFUTIL to continue to the end of the input file if the input is suspected to be out of sequence or have duplicated data blocks.

COPY SIMULATION. NO RECORDS WERE ACTUALLY COPIED.

SIMULATE in effect. Run is for data scanning purposes only. No data will be copied.

DATA SET dsname ON VOLUME vvvvvv BEING CLEARED FOR REUSE.

All checks are complete and the requested data set clear function will be performed. Ensure that a completion message follows this message.

DATA SET dsname ON VOLUME vvvvvv HAS BEEN CLEARED FOR REUSE. CLEARED BLOCK COUNT WAS nnn.

The clear function completed successfully for the specified SMF data set. The number of blocks written during the CLEAR function for this data sets was 'nnn'.

DATESCAN REQUESTED. NO MISSING DATA DISCOVERED.

No gaps between records larger than the DATESCAN delta were detected.

DATE RANGE ccyy.ddd TO ccyy.ddd SELECTED

Shows the Julian dates that will be selected for processing.

DELDUP IN EFFECT. DUPLICATE RECORDS WERE DELETED.

DELDUP was requested and duplicate records were found and deleted.

DELDUP IN EFFECT. NO DUPLICATE RECORDS FOUND.

DELDUP was requested but no duplicate records were detected.

DSPLIT REQUESTED AND CHANGE FOUND. RECORDS WERE WRITTEN TO SYSUT3.

DSPLIT was specified and a DAY split boundary was encountered in the primary output. Data was written to SYSUT3.

DSPLITDP REQUESTED AND CHANGE FOUND. RECORDS WERE WRITTEN TO SYSUT3D.

DSPLITDP was specified and a DAY split boundary was encountered in the duplex output. Data was written to SYSUT3D.

DSPLIT REQUESTED. NO CHANGE FOUND.

DSPLIT was specified but no change in DAY was found. No split was performed. SYSUT3 was not used.

DSPLITDP REQUESTED. NO CHANGE FOUND.

DSPLITDP was specified but no change in DAY was found. No split was performed. SYSUT3D was not used.

DUPLICATE RECORDS - n

This is the number of records deleted by the DELDUP function.

DYNAMIC ALLOC(nnnn) SUCCESSFUL FOR DDNAME dddddddd WITH DSNAME ssssssss.

The file DDNAME dddddddd was successfully allocated to DSNAME ssssssss with disposition nnnn. The value 'nnnn' will be 'SHR', 'OLD', 'MOD', 'MODNW' or 'NEWMD'. A value of 'NEWMD' indicates a non-specific (no volume serial) data set disposition was changed to new by allocation (SVC 99) since the data set did not currently exist and had to be created. A value of 'MODNW' indicates a MOD disposition for a data set targeted to a specific DASD volume was changed to from 'MOD' to 'NEW' by SMFUTIL as the data set does not currently exist on the volume. This avoids a 'data set not found' allocation error. If no space parameters were supplied, a default space value of (CYL,15,15) will be inserted.

FIRST DATE STAMP - ccyy.ddd hh:mm:ss.th

This is the date/time stamp from the first input record encountered.

FIRST VOLUME BLOCK ID START: DEC=ddd, HEX=hhhhhh

The selected data set's archive record supplied a BLOCKID pointer for the first date requested from the first volume of the set. The listed hex value will be used to perform a POINT macro to perform a high-speed cartridge positioning of the tape to the first block containing data that is required.

HIGH DATE STAMP - ccyy.ddd hh:mm:ss.th

This is the highest date/time stamp encountered in the output data processed.

I/O ERRORS - n

This is the total number of I/O errors that occurred on the input DDNAME (SYSUT1xx).

INPUTLIMIT RECORD COUNT LIMIT REACHED. SMFUTIL TERMINATING DUE TO LIMIT.

The specified record limit for input records was reached.

INPUT SORT CHECKING REQUESTED. INPUT RECORDS ARE IN SORTED ORDER BY DATE AND TIME.

All input records were within sort tolerance with respect to date/time values in the record header.

INSERTED RECORDS - CCC,CCC,CCC

The indicated number of records were inserted into the output data stream as a result of a user POSTEXIT requesting a record insertion.

INVALID RECORDS - n

This is the total number of input records that were invalid and discarded for any structural reason.

INVALID SEGMENTS - n

This is the total number of invalid record segments found.

LAST DATE STAMP - ccyy.ddd hh:mm:ss.th

This is the date/time stamp from the last record processed.

LOW DATE STAMP - ccyy.ddd hh:mm:ss.th

This is the lowest date/time stamp encountered in the output data processed.

MANXALOC ACTION - SMF DATA SET dsname SELECTED FOR INPUT. START DATE/TIME IS CCYY.JJJ HH:MM:SS.TH.

The specified SMF data set was selected for input processing because MANXALOC was specified and the data set contained data.

MSPLIT REQUESTED AND CHANGE FOUND. RECORDS WERE WRITTEN TO SYSUT3.

MSPLIT was specified and a month boundary was encountered in the primary output. Data was written to SYSUT3.

MSPLITDP REQUESTED AND CHANGE FOUND. RECORDS WERE WRITTEN TO SYSUT3D.

MSPLITDP was specified and a day boundary was encountered in the duplex output. Data was written to SYSUT3D.

MSPLIT REQUESTED. NO CHANGE FOUND.

MSPLIT was specified but no change in month was found. No split was performed. SYSUT3 was not used.

MSPLITDP REQUESTED. NO CHANGE FOUND.

MSPLITDP was specified but no change in month was found. No split was performed. SYSUT3D was not used.

PRINT SPECIFIED AND SYSUT2 NOT AVAILABLE. PRINT ONLY RUN ASSUMED. SIMULATE DEFAULTED.

The PRINT function was specified without a SYSUT2 DD statement to receive output. SMFUTIL assumes that the print of the records selected is all that is required and defaults to a simulate run with no records actually being copied to an output data set.

OUTPUTLIMIT RECORD COUNT LIMIT REACHED. SMFUTIL TERMINATING DUE TO LIMIT.

The specified record limit for output records was reached.

OUTPUT SORT CHECKING REQUESTED. OUTPUT RECORDS ARE IN SORTED ORDER BY DATE AND TIME.

All output records were within sort tolerance with respect to date/time values in the record header.

OVERFLOW RECORDS - n

This is the total number of records that could not be placed to the output data set because they had logical record lengths longer than the block size of the output data set and the record format was VB. An "SMFU182" message will be produced for each DDNAME that was affected.

RECORDS IN - n

This is the total number of input records.

RECORD TYPES IN DATA SET:

This message precedes a spot graph indicating the record types in the data set.

SEGMENTS DROPPED - n

This is the total number of out of place record segments found.

SEGMENTS IN - n

This is the total number of record segments found.

SMFUTIL IS RUNNING AUTHORIZED ON SYSTEM sysid DYNAMIC ALLOCATION WAITS FOR VOLUMES OR DATA SETS WILL BE ALLOWED.

The current execution is from an authorized environment (i.e. from LNKLSSTxx or an APF authorized data set). Because of this SMFUTIL will be able to request that dynamic allocation wait for 'inuse' devices, volumes, or data sets to become available.

SMFUTIL COMPLETE: RETURN CODE=xx RUN DURATION=hh:mm:ss.

Processing has completed with the specified return code. The wall clock duration of the run is as shown.

SYSUT2D PRESENT. SYSUT2 HAS BEEN AUTOMATICALLY DUPLEXED.

The SYSUT2D DD statement was supplied and a duplex split was not requested. SMFUTIL has automatically duplexed the contents of SYSUT2 to the data set specified by SYSUT2D.

SYSUT3D PRESENT. SYSUT3 HAS BEEN AUTOMATICALLY DUPLEXED.

The SYSUT3D DD statement was supplied and duplex split was not requested. SMFUTIL has automatically duplexed the contents of SYSUT3 to the data set specified by SYSUT3D.

TIME RANGE hhmmssst TO hhmmssst SELECTED

Shows the time boundaries in effect.

TIME RANGE IS EXCLUSIVE. ONLY RECORDS OUTSIDE RANGE WILL BE COPIED.

The XTIME keyword was used. All records that have a time within the range will be excluded.

TIME RANGE IS INCLUSIVE. ONLY RECORDS WITHIN RANGE WILL BE COPIED.

The ITIME keyword was used. All records that have a time outside the range will be excluded.

TIME RANGE IS NORMAL: RECORDS WILL BE COPIED FROM START TIME ON FIRST ELIGIBLE DATE TO END TIME ON LAST ELIGIBLE DATE.

The TIME keyword was used. All records from the start time on the first eligible date to the end time on the last eligible date will be copied. Note that all times are copied for days between the start and end date.

TOTAL RECORDS OUT - n

This is the total count of records written to all output data set including records written more than once.

TOTAL BLOCKS OUT - n

This is the total count of physical blocks written to all output data set.

TOTAL SEGMENTS OUT - CCC,CCC,CCC

This is the total count of record segments written to all output data set.

TYPE nnn RECORDS - IN -x OUT - y

For record type nnn, x is the number of records input, and y is the number of records output.

VSAMMOD SPECIFIED OUTPUT VSAM DATA SETS WILL BE EXTENDED. ALL OTHERS WILL BE OVERWRITTEN.

The VSAMMOD keyword was specified with one or more ddnames specified in the parameter list. All VSAM output data set who's ddnames are identified in the list will be extended. This means that data from this copy execution will be added to the end of the data set as if a MOD disposition had been specified. Any VSAM output data set not specified will be overwritten (i.e. DISP=OLD).

WARNING - NO INPUT DATA FOUND ON SYSUT1.

The data set specified by SYSUT1 was empty.

WSPLIT REQUESTED AND CHANGE FOUND. RECORDS WERE WRITTEN TO SYSUT3.

WSPLIT was specified and a week boundary was encountered in the primary output. Data was written to SYSUT3.

WSPLITDP REQUESTED AND CHANGE FOUND. RECORDS WERE WRITTEN TO SYSUT3D.

WSPLITDP was specified and a week boundary was encountered in the duplex output. Data was written to SYSUT3D.

INDEX

&

&C	197
&D	197
&DD	197
&DID	
system id substitution from output data	13, 14, 129, 182, 196
&DMMYY	197
&DMMYY	197
&JJJ	197
&M	197
&MDDYY	197
&MM	197
&MMDD	197
&MMYY	197
&MMYYYY	197
&SID	
system id substitution from executing system	7, 13, 14, 39, 129, 196
&YJJJ	197
&YYY	197
&YYYJJJ	197
&YYYYMM	197

A

ABEND	
S002-18	4
S413-40	125
S713-04 - preventing on MOD restart	161
U999 - caused by security exit	210
Abends	
B37/D37	164
ACCOUNT	23
location of within record	154
recognition	153
source of	153
ACTIVEOK	23
AEXPIRE	120
ALL	24
ALLOW2	24
ALLOW3	24
ALTUNITS	25
APF authorization	142
ARCBACKUP	26
ARCFORCE	27
ARCHBIAS	27
ARCHINPT	28
ARCHIVDB	28, 179
ARCHIVE	29
ARCHIVE DATABASE	
backup	184
Block ID retention	181
defining and initializing	176
GRS Management of reserve during update	177
informing SMFUTIL of name	179
maintenance	186
maintenance via ISPF	187
recovering data	179
reporting, batch	190
retrieving smf data using	191
serialization of updates	177
update sequence	182
updates protection via smf records	178

ARCHIVE ISPF INTERFACE

access control	157, 187
INSTALLATION	158, 187
OPERATION	158, 188
ARCHIVEACTIVE	30
ARCHIVERECOVER	30
ARCHIVEUSER	32
Archiving	
INPUT data	181
multiple sid handling	182
OUTPUT data.....	180
rules for use of.....	183
SMF Data	175
user data sets.....	180
archsid.....	30
archsmf	31
ARCHSMF#	179
archsmfrecnum	31
ARCHSMFRECNUM	179
archsrch.....	31
ARCHSRCH.....	31
ASETNAME.....	120
ASMFIDLN.....	32
AUTHUSER	207

B

B37/D37 Abends

Recovering from.....	164
BADSNAPLIMIT.....	32
BGUSER	33
BIDBYHOUR.....	33, 121
BIDINDEX	34
BIDPOINT	34
BIDSTART	121
BLKSIZE.....	122
BUFNO.....	122
bypassing previously processed data	163
BYTEDETAIL	34

C

CATALOG DATA SET INDEX LEVEL INPUT	170
CATINPUT.....	35
AUTOMATIC INPUT OF DATA SETS UNDER A CATALOG INDEX.....	170
CATLG	
use of with split data sets	166
CHKINPT	163
checkpoint data set	
uses	157
Checkpointing	
dumping sequence	162
input data	162
MOD processing	
high speed positioning for mod output.....	163
restart for new data sets	161
restart of DUMPCLEARALL.....	162
restart point for mod data sets.....	157, 160
CHKISORT	36
CHKOSORT	36
CICSSPLIT	37
CICSSUBC	38
CICSTYPE	39

CKPTDB	39
CKPTGDG	123, 124
CKPTINPT	40
CLEAR	1, 3, 7, 8, 41, 139, 219
CLEARING SMF DATA SETS	219
concatenation of unlike devices	7
CONTROL KEYWORDS	
DDA BLOCK	120
Normal	22
COPY0OK	41, 216
COPYBOTH	11, 42, 99, 139, 150
COPYVALUEBOTH	43
CREATE\$0	44
CREATEU0	44
CURRENTMONTH	45
CURRENTWEEK	45

D

DARETRY	46
DASD	
use of with split data sets	166
DATA HANDLING	
INPUT DATA	7
OUTPUT DATA	9
DATA HANDLING METHODOLOGY	7
DATA SET	46
data set names	
selecting by	155
DATA SPLITTING	166
DATACLASS	124
DATE	47, 103, 140, 152
DATE INSERTION INTO OUTPUT DSNAME	197
date range	
defaults	103
DATESCAN	47
DB2SPLIT	48
DCACHECHECKPOINT	49
DCB	125
DCBABENDEXIT	49
DCBRESET	50
DDA BLOCK	
coding examples	195
dsname suffix generation	196
dsname symbolic date value substitution	197
dsname symbolic system id substitution	196
split request	195
DDACOPY0RC	127
DDACREATE	128
DDAEND	126
DDALARBEBLOCK	126
DDANOCREATE	128
DDANOLARBEBLOCK	126
DDASTART	126
DDEXIT	50
DDNAME	128
DDRETAIN	52
DECR	52
DEFAULT LISTING	
Enabling at execution time via PARM	206
Suppressing at execution time via PARM	206
DEFAULT OVERRIDE EXCLUSIONS	149
defaults	
for control keywords	199

DEFAULTS	
OVERRIDE AT EXECUTION TIME	147
DELDUP	53, 156
design purpose	156
DELDUPRV	53, 156
DELETERC	54
DETAIL	54, 139
DISKONLY	55
DISP	129
for mod data set	160
for split data sets	160
DMPCLALL	57
checkpoint data set required	217
DMPCLALL PROCESSING	217
DATA SET SWITCHING	218
DSCB	
format 1 on disk	160
DSNAME	129
DATE INSERTION INTO	197
dsname suffix	
automatic generation	167
DSNAME SUFFIX GENERATION	
DAILY	167
DATE and TIME	167
MONTHLY	167
WEEKLY	167
DSPLIT (Obsolete)	55
DSPLITDP (Obsolete)	55
DTRANGE	56
DTVALOFF	57
DUMMY	130
DUMPCLEARALL	57
DUMPING	
MULTIPLE DATA SET DUMPING	216
SINGLE DATA SET DUMPING	216
dynamic allocation of data sets	195

E

EDATE	58
EOF370	58
ERROR MESSAGES	
EXECUTION AND CONTROL KEYWORD ERRORS	233
normal log messages	265
ESTAEOFF	58
ESTAEON	58
Examples	
CONTROL STATEMENTS	139
EXCLUDE	21, 59, 99, 139, 169
excluded record data set	4
EXCONLY	59, 99
execution control	21
introduction to control keywords	21
EXECUTION LOGGING	5
EXECUTION PARM	
DFLTLIST - Print the Default listing	206
NODFLTLI - Suppress the Default listing	206
Execution Region Size Requirements	146
EXIT	59, 224
EXITS	
IFASMFDPCOMPATIBILITY	225
SECURITY	210
USER	223, 227
EXPDT	130

F

FASTMOD	60
FORCE	60
FRI	90

G

GDATE	60, 140
GDG DATA SET	
SPECIAL PROCESSING CAPABILITIES	170
GDG DATA SET INPUT	
VIA CKPT RECORDS	171
GDG DATA SETS	
INTERNAL CONTROL OF RELATIVE LEVELS	171
GDGCKPTI	61
GDGINPUT	62
GENDSND	63
GENDSNM	63, 169
GENDSNW	64
GOTOEOF	7, 64, 94
GRS (Global Resource Serialization)	
Management of reserve during update to archive database	177

H

High-speed cartridge positioning	
for data retrieval	173
for MOD processing	174
High-speed positioning of cartridge devices	173

I

IBMCLEAR	100
IDIST	67
IDMS/R	
example of out of sort data	156
product producing RTE data	225
IDTRANGE	65
IEAAPFxx	3
IEC145I	
with return code 40	125
IEC148I	
with return code 04	161
IECIOSxx	
Changing the MIH timing for tape devices	174
IEFU29	163
INSTALLATION	19
IFASMFDP	1, 169, 216
IFASMFDP TYPE EXIT POINTS	225
implementation	11
INCLUDE	66, 99, 139, 150, 152
INONLY	66, 99, 139, 150
INDTVOFF	66
INITEXIT	67, 224
initial testing	
IVP Job	11
INPUT DATA	
ENTIRE CATALOG INDEX	172
ENTIRE GDG INDEX	172

VIA THE CATALOG	172
INPUT DATA SET	
DELETION AT END OF RUN	173
Input data sets	3
INPUT DATA SPLITTING.....	166
INPUTDATA DISTRIBUTION	67
INPUTDETAIL1	68
INPUTLIMIT	68
installation	15
INSTALLATION	
Ensuring MOD tape restart capability	19
INSTALLATION JOB STREAM	16
Introduction.....	1
IOELIMIT.....	68
ISSETNAME	69
ISMF	69
ISPF	
ARCHIVE ISPF INTERFACE.....	187
ISPFUSRS	207
ITIME	70, 103
ITIMERANGE	71
IUCB.....	72
IVOLSER	72
IVOLUME	72
IVSTART	73

J

JCL requirements	
execution	3
unloading tape	15
job name	
masking	152
other record types	152
type 30 records	152
JOBLIB.....	3, 224
JOBNAME	139
JOBNAME	73, 139
jobname id	
recognition.....	152

K

keywords	
ACCOUNT.....	23
ACTIVEOK	23
ALL	24
ALLOW2	24
ALLOW3	24
ALTUNITS	25
ARCBACKUP:	26
ARCFORCE.....	27
ARCHBIAS.....	27
ARCHINPT	28
ARCHIVDB	28
ARCHIVE	29
ARCHIVEACTIVE.....	30
ARCHIVERECOVER.....	30
ARCHIVEUSER	32
ARCHSRCH	31
ASMFIDLN	32
BADSNAPLIMIT	32
BGUSER	33

BIDINDEX	34
BIDPOINT	34
BYTEDETAIL	34
CATINPUT	35
CHKISORT	36
CHKOSORT	36
CICSSPLIT	37
CICSSUBC	38
CICSTYPE	39
CKPTDB	39
CKPTINPT	40
CLEAR	41
COPYOOK	41
COPYBOTH	42
COPYVALUEBOTH	43
CREATES0	44
CREATEU0	44
CURRENTMONTH	45
CURRENTWEEK	45
DARETRY	46
DATA SET	46
DATE	47
DATESCAN	47
DB2SPLIT	48
DCACHECKPOINT	49
DCBABENDEXIT	49
DCBRESET	50
DDA BLOCK	
AEXPIRE	120
ASETNAME	120
BIDBYHOUR	33, 121
BIDSTART	121
BLKSIZE	122
BUFNO	122
CKPTGDG	123, 124
DATACLASS	124
DCB	125
DDACOPYORC	127
DDACREATE	128
DDAEND	126
DDALARGEBLOCK	126
DDANOCREATE	128
DDANOLARGEBLOCK	126
DDASTART	126
DDNAME	128
DISP	129
DSNAME	129
DUMMY	130
EXPDT	130
LABEL	130
LIKE	131
LRECL	131
MANAGEMENTCLASS	131
MODEL	132
RECFM	132
RETPD	132
RETRY	133
SPACE	133
SPLITDISP	134
STORAGECLASS	134
SUFFIX	135
TRTCH	137
UNIT	137
VOLCOUNT	137
VOLNUM	138
VOLUME	138

DDEXIT	50
DDRETAIN	52
DECR	52
DELDUP	53
DELDUPRV	53
DELETERC	54
DETAIL	54
DISKONLY	55
DMPCLALL	57
DSPLIT (Obsolete).....	55
DSPLITDP (Obsolete).....	55
DTRANGE	56
DTVALOFF	57
DUMPCLEARALL.....	57
EDATE	58
EOF370	58
ESTAEOFF	58
ESTAEON.....	58
EXCLUDE	59
EXCONLY	59
EXIT	59
FASTMOD	60
FORCE	60
FRI.....	90
GDATE	60
GDGCKPTI.....	61
GDGINPUT	62
GENDSND	63
GENDSNM	63
GENDSNW	64
GOTOEOF	64, 94
IBMCLEAR	100
IDIST.....	67
IDTRANGE	65
INCLUDE	66
INCONLY	66
INDTVOFF	66
INITEXIT	67
INPUTDATADISTRIBUTION	67
INPUTDETAIL.....	68
INPUTLIMIT	68
IOELIMIT	68
ISETNAME.....	69
ISMF	69
ITIME.....	70
ITIMERANGE	71
IUCB	72
IVOLSER	72
IVOLUME	72
IVSTART	73
JOBNAME	73
LARGEBLOCK	74
LBI	74
LINESPERPAGE.....	74
LISTALL.....	75
LISTLICENSE	75
LOCATE	80
MANILEAVE.....	75
MANXALOC	76
MON	90
MONTHCAL	76
MONTHPCAL	76
MQSPLIT.....	77
MSPLIT (Obsolete).....	77
MSPLITDP (Obsolete).....	78
MULTICLEAR	78

NIORATIO	79
NOACTIVE	23
NOALL	24
NOALLOW2	24
NOALLOW3	24
NOALTUNITS	25
NOARCFORCE	27
NOARCHDB	28
NOARCSID	79
NOBIDPOINT	34
NOBYTEDETAIL	34
NOCKPTIN	40
NOCOPY0	41
NOCOPYBOTH	42
NOCOPYVALUEBOTH	43
NODCBABENDEXIT	49
NODCBRESET	50
NODETAIL	54
NOFASTMOD	60
NOFORCE	60
NOLARGEBLOCK	74
NOLBI	74
NOLOCATE	80
NOMULTICLEAR	78
NOORDERCHECK	82
NOOUTPUTDETAIL	84
NOSHORT	94
NOSHORTCLEAR	93
NOSMFIDCHECK	96
NOSUCLEAR	100
NOSYMBOLS	100
NOSYSUT1	80
NOTERSE	102
NOTIFY	80
ODIST	85
ODTRANGE	81
ORDERCHECK	82
ORDERGAP	82
OSYNADOFF	83
OTIMERANGE	83
OUTPUTDATADISTRIBUTION	85
OUTPUTDETAIL	84
OUTPUTLIMIT	84
POSTEXIT	85
PREEXIT	86
PREVIOUSWEEK	87
PREVMNTH	86
PRINT	87
PROCESS	88
PROCSYS	88
RECEXIT	89
REPORTFIXED	89
RVALEXIT	89
SAT	90
SCLRTHLD	92, 93
SCRATCHRETRY	91
SDB	92
SEARCHARCHIVE	90
SETNAME	91
SHORTCLEAR	93
SHORTCOPY	94
SIDSPLIT	94
SIMULATE	95
SKIPHIDT	95
SKIPOLD	95
SMFIDCHECK	96

SORTTOLERANCE	96
SPIEOFF	97
SPINBACKUP	97
SRBUFF	97
STOP	64
SUBSET	98
SUBTYPE	99
SUBUFF	99
SUCLEAR	100
SUN	90
SWBUFF	100
SYMBOLS	100
SYNADOFF	100
SYSID	101
SYSTYPE	101
TAPEONLY	101
TERMEXIT	102
TERSE	102
TEXTSCAN	102
THSUFFIX	103
THU	90
TIME	103
TODAY	104
TSOUSER	104
TUE	90
TYPETODD	105
UCB	72
USER1	106
USER2	106
USER3	107
USERID	107
VALUEONLY	108
VCBUFF	108
VERBOSE	102
VRBUFF	108
VSAMMOD	109
VSINGLE	109
VWBUFF	109
WED	90
WEEKEND	110
WEEKPCAL	110
WEEKSTART	110
WORKDAY	111
WSPLIT (Obsolete)	111
WSPLITDP (Obsolete)	112
XACCOUNT	112
XBGUSER	113
XDATASET	113
XERROFF	114
XJOBNAME	114
XSYSID	115
XSYSTYPE	115
XTIME	115
XTSOUSER	116
XUCB	117
XUSERID	117
XVOLSER	118
XVOLUME	118
YEARFIRST	118
YEARFIRSTM	118
YEARFIRSTW	119
YESTERDAY	119
YFIRST	118
YFIRSTM	118
YFIRSTW	119

L

LABEL	130
Large Block Interface Support	141
LARGEBLOCK	74
LASTREC1	163
LBI	74
LBI Support	141
LIKE	131
LINESPERPAGE	74
LISTLALL	75
LISTLICENSE	75
LNKLSTxx	3
LOCATE	80
LPALIB	3
LRECL	131

M

MANAGEMENTCLASS	131
MANx	139
MANX DATA SET SWITCHING	218
MANXALOC	76
messages	
unnumbered	265
MIH TIMING	174
MOD	2, 160, 216
MODEL	132
model dscb	
and invalid DCB values	4
MODIFICATION	199
EXECUTION ACCESS CONTROL	207
EXECUTION DEFAULTS	199
KEYWORD ACCESS CONTROL	208
RECORD ACCOUNTING TABLE	209
SECURITY EXIT INSTALLATION	210
SUBSET DEFINITIONS	208
MON	90
month split	
special installation month calendar	211
MONTHCAL	76
MONTHPCAL	76
MQSPLIT	77
MSPLIT (Obsolete)	77
MSPLITDP (Obsolete)	78
MULTICLEAR	78
multiple volume input data sets	4
MVS/370	219
MVS/ESA	219
MVS/SP	219
MVS/XA	219

N

NIOTATIO	79
NOACTIVE	23
NOALL	24
NOALLOW2	24
NOALLOW3	24
NOALTUNITS	25
NOARCFORCE	27
NOARCHDB	28

NOARCSID	79
NOBIDPOINT	34
NOBYTEDETAIL	34
NOCKPTIN	40
NOCOPY0	41
NOCOPYBOTH	42
NOCOPYVALUEBOTH	43
NODETAIL	54
NOFASTMOD	60
NOFORCE	60
NOLARGEBLOCK	74
NOLBI	74
NOLOCATE	80
NOMANILEAVE	75
NOMULTICLEAR	78
NOORDERCHECK	82
NOOUTPUTDETAIL	84
NOSHORT	94
NOSHORTCLEAR	93
NOSMFIDCHECK	96
NOSUCLEAR	100
NOSYMBOLS	100
NOSYSUT1	80
NOTERSE	102
NOTIFY	80

O

ODIST	85
ODTRANGE	81
operating system type	
specifying for data exclusion	115
specifying for data selection	101
operational considerations	141
ORDERCHECK	82
ORDERGAP	82
OSYNADOFF	83
OTIMERANGE	83
output data set i/o errors	164
Output files	4
OUTPUTDATADISTRIBUTION	85
OUTPUTDETAIL	84
OUTPUTLIMIT	84

P

PARM	
DFLTLIST - Print the Default listing	206
NODFLTLI - Suppress the Default listing	206
PARMLIB	
DEFAULT OVERRIDE EXCLUSIONS	149
LICENSE MEMBER VALIDATION	150
SPECIFYING DEFAULTS AND LICENSE IN A SINGLE MEMBER	149
SPECIFYING VIA A JCL DD STATEMENT	147
SPECIFYING VIA SMFUDFLT	147
SPECIFYING WITH MEMBER NAME VIA JCL	149
PARMLIB MEMBER NAMES	
SPECIFYING VIA EXECUTE PARM STRING	148
SPECIFYING VIA SMFUDFLT	147
POSTEXIT	85, 224
PREEXIT	86
PREVIOUSWEEK	87
PREVMNTH	86

PRINT	2, 87
PROCESS	88
PROCSYS	88
production phase in	
data moving	11
dumping & clearing	13

R

RACF	151
RECEXIT	89, 224
RECFM	132
automatically reset to valid value	4
record security exit	210, 211
record subtypes	150
selecting by	150
selection criteria	151
RECOVERING FROM AN OUTPUT I/O ERROR DURING MOD TO TAPE	165
Recovering from B37/D37 abends	164
REMOVING MOD DATA FROM A BAD TAPE	165
REPORTFIXED	89
RETPD	132
RETRY	133
return code	
effect on by keywords	22
final user exit call ignored	225
from post exit	224
from record security exit	211
from regular and record exit	224
from security exit	210
from user supplied duplicate record exit	50
minimum after exit termination	224
preventing due to empty data set	216
use in multiple step jobs	216
RETURN CODE	
SMFUTIL	231
RTASMLK	209
RTE record	156
RVALEXIT	89

S

S002-18 ABEND	4
SAFE CLEAR PROCESSING	220, 221
SAT	90
SCLRTHLD	92, 93
SCRATCHRETRY	91
SDB	92
SEARCHARCHIVE	90
SECURITY exit	210
Serialization of execution	143
SETNAME	91
SHORTCLEAR	93
SHORTCOPY	94
SHORTCPY	7
SIDSPLIT	94
SIMULATE	95
SKIPHIDT	95
SKIPOLD	95, 163
SMF DATA	
COLLECTION AND ARCHIVING PROCESS	215
COLLECTION METHODOLOGIES	215
DUMPING FREQUENCY	221

SAFE CLEAR PROCESSING	220, 221
smf data order checking	156
SMF EXITS	
IEFU29	216
IEFU83	225
SMFDUMP	
replacing with DMPCLALL	216
SMFIDCHECK	96
SMFSSASM	208
SMFSSTBL	98, 208
SMFUARPT - ARCHIVE DATABASE	
reporting, utility	190
SMFUDFLT	199, 207
SMFUSECX	210, 211
SMFUTIL DEFINED EXIT POINTS	223
SMFUTLRT	209
SMFWTM	156
SMS	
ABENDs caused by invalid DCB values	4
and automatic DCB selection	4
sorting smf data	156
SORTTOLERANCE	96
SPACE	133
SPIEOFF	97
SPINBACKUP	97
SPLITDISP	134
splitting data	
output data	166
SPLITTING DATA	
BY DAY	166
BY MONTH	166
BY WEEK	166
SPLITTING OUTPUT DATASETS	9
SRBUFF	97
STEPLIB	3, 224
STOP	64
STORAGECLASS	134
SUBSET	21, 98, 99, 139, 152, 208
SUBSETS	
DEFINING	98
SUBTYPE	99, 105, 150
SUBUFF	99
SUCLEAR	100
SUFFIX	135
SUFFIX GENERATION	
DAILY	167
DATE and TIME	167
MONTHLY	167
WEEKLY	167
SUN	90
SWBUFF	100
SYMBOLIC SYMBOL SUBSTITUTION	146
SYMBOLS	100
USE IN DEFAULT DEFINITIONS IN PARMLIB MEMBER	150
SYNADOFF	100
SYSCKPT	105, 160, 161
SYSCKPT data set	160
SYSID	101
SYSIN	105
SYSLOG	
PURPOSE AND USAGE	5
SYSOUT	105
SYSPRINT	105
output to dynamically allocated data set	144
SYSTEM ID	
symbolic substitution into dsnames	196

SYSTYPE.....	101
SYSUT1	
PURPOSE AND USAGE	7
SYSUT2	160, 168
PURPOSE AND USAGE	9
SYSUT2D.....	160, 168
PURPOSE AND USAGE	9
SYSUT2X	
excluded record data set	4
PURPOSE AND USAGE	9
SYSUT3	160, 166
PURPOSE AND USAGE	9
use in data splitting.....	166
SYSUT3D.....	160, 166
PURPOSE AND USAGE	9
SYSUTxx	99, 105

T

tape	
product distribution	
contents	15
unloading	15
TAPEONLY	101
tapes	
importance of deferred mounting	166
TERMEXIT.....	102, 224
TERSE.....	102
TEXTSCAN	102
THSUFFIX	103
THU.....	90
TIME	103, 152
time range	
exclusive.....	115
inclusive	70
normal	103
TLMS	
checkpoint restart for mod tapes.....	161
TMS	
checkpoint restart for mod tapes.....	161
TODAY	104
TRTCH.....	137
TSO	104, 116
TSOUSER	104
TUE	90
type 72 records	225
TYPETODD	99, 105, 139, 150, 156

U

UCB.....	72
UNIT	137
unit names	
IDENTIFYING TAPE DEVICES.....	213
modifying the unit name table.....	213
used for dynamic allocation	193
unnumbered messages	265
USER EXITS.....	223, 227
user id	
and other record types	152
and security systems.....	151
masking	152
recognition	151

source of	151
type 30 records	151
USER1	106
USER2	106
USER3	107
USERID	107, 150

V

validation	
input data blocks	1
input data records	1
VALUEONLY	108
VCBUFF	108
VERBOSE	102
VOLCOUNT	137
VOLNUM	138
VOLUME	138
VRBUFF	108
VSAM	1
VSAM buffering defaults	206
VSAMMOD	109
VSINGLE	109
VWBUFF	109

W

WED	90
week split	
special installation week calendar	211
WEEKEND	110
WEEKPCAL	110
WEEKSTART	110
wildcard characters	
use in buffers in security exit	210
use in job name	152
use in SYSID	101
use in TSOUSER	104
use in USERID	107
use in XSYSID	115
use in XTSOUSER	116
use in XUSERID	117
WORKDAY	111
WSPLIT (Obsolete)	111
WSPLITDP (Obsolete)	112

X

XACCOUNT	112
XBGUSER	113
XDATASET	113
XERROR	114
XJOBNAME	114
XSYSID	115
XSYSTYPE	115
XTIME	103, 115
XTSOUSER	116
XUCB	117
XUSERID	117
XVOLSER	118
XVOLUME	118

Y

YEARFIRST	118
YEARFIRSTM.....	118
YEARFIRSTW.....	119
YESTERDAY	119
YFIRST	118
YFIRSTM.....	118
YFIRSTW.....	119