

Creating and Implementing the ACC / SRS Rules

A Setup Guide for Allocation Control Center and Space Recovery System



Copyright 2011
DTS Software, Inc.
1818 Lakefield Ct. SE
Conyers, GA 30013

Table of Contents

Creating and Implementing the ACC / SRS Rules.....	1
A Setup Guide for Allocation Control Center and Space Recovery System.....	1
Table of Contents.....	2
Purpose of the ACC/SRS Rules.....	3
Purpose of this Guide.....	3
Location of the ACC/SRS Rules.....	4
How ACC / SRS Rules are Activated.....	4
Activating Rules at Product START.....	4
F DIF,START ACC MYRULES.....	4
REFRESHing Rules.....	4
F DIF,REFRESH SRS SRSRULES.....	4
What Rules are Active Now?.....	5
F DIF,STATUS.....	5
F DIF,SET productname RULELIST ON.....	5
Creating an ACC / SRS Rule Set.....	6
Rules Language Syntax.....	6
Rules Language Architecture.....	6
Comments, Readability, and Maintainability.....	6
Continuation.....	6
Nested IF's.....	7
THEN SET &DISKPOOL = PRODPOOL.....	7
Flow of Control.....	7
A Sample Rule Member.....	8
Defaults and Initialization Section.....	8
DEFPROD – Set Product-Wide Defaults.....	8
SMF Recording.....	8
Wildcards and Masking.....	8
Sharing ACC / SRS Pools.....	8
DEFENV – Processing Defaults for Environment.....	9
DEFFILT – Define Filter Lists.....	9
DEFDMYx – Process Obsolete Units / Volumes.....	9
Bypass-Product-Processing Section.....	10
Bypass SRS ADDVOL Checks Section.....	10
Standards Enforcement Section.....	10
Setting RLSE.....	10
Controlling Tape Allocations.....	10
Controlling RENAME processing.....	11
Other Installation Standards.....	11
Pooling Section.....	11
Pool Definition and Recovery Criteria Section.....	11
Messages Section.....	12
Troubleshooting.....	12
Syntax Errors.....	12
Logic Errors.....	12
Technical Support.....	12
Appendix A – The Sample ACC / SRS Rules.....	13

Purpose of the ACC/SRS Rules

Allocation Control Center and Space Recovery System are powerful, flexible systems software products that can control datasets as they are created, extended, and used in modern enterprise systems environments, including OS/390 and z/OS, as well as in Hitachi and Fujitsu systems. Allocation Control Center (ACC) provides a means to examine, test, and if necessary, dynamically alter many of the parameters that define how and where datasets are created and used, without the need to modify JCL, IDCAMS control statements, or SMS constructs. Space Recovery System (SRS) uses sophisticated processing techniques to effectively eliminate all out-of-space conditions that would otherwise occur in a large-systems environment, whether the cause is incorrect or obsolete size estimates, job scheduling conflicts, end-of-period processing load, or other, unpredictable types of failures. ACC and SRS are designed to meet the need for centralized, systemwide policies that are easily understood, enforced, and maintained. The DIF (Dynamic Install Facility) Rules Language is the key to the power and flexibility of ACC and SRS. DIF provides a common rules syntax and architecture across all of the DTS products. ACC and SRS run under control of DIF, which provides all of the services ACC and SRS need to accomplish their tasks. DIF commands allow the ACC / SRS administrator to start, stop, and modify product operation, including which rules are to be used.

Purpose of this Guide

This guide provides an overview of how the ACC/SRS rules are used, along with an example system which uses ACC and SRS for standards enforcement, dataset management, and out-of-space error prevention. The example system shows only some of the most common uses of ACC and SRS. Many more powerful capabilities, along with detailed explanations of the DIF rules language syntax, are described in the Dynamic Install Facility User's Guide and the Allocation Control Center / Space Recovery System User's Guide. Both of these documents, along with product load libraries and maintenance fixes, can be obtained from the DTS Software web site, www.DTSSoftware.com, or by calling DTS Software at 770-922-2444.

Location of the ACC/SRS Rules

ACC and SRS may use a single, common set of rules, or both products may use different rules. The rules consist of a series of statements in 80-character records, which must be contained in a partitioned dataset member. The partitioned dataset must be named in the //PARMLIB DD statement of the DIF cataloged procedure.

How ACC / SRS Rules are Activated

Activating Rules at Product START

When a MODIFY DIF,START command is issued in order to instruct DIF to start ACC or SRS, the appropriate member of the PARMLIB dataset is read and the rules are dynamically compiled and loaded into common storage for use by ACC or SRS. If a rule name (the name of a PDS member) is specified on the START statement, that member is used. If no rule name is specified, then a default name is used. For ACC and SRS, the default name is ACCRULES. START commands are normally placed in the STARTxx member of the DIF PARMLIB dataset so that they are executed automatically when DIF starts. An example of the START command is shown below. In this example, the member MYRULES in the PARMLIB dataset is used, rather than the default of ACCRULES.

```
F DIF,START ACC MYRULES
```

REFRESHing Rules

The REFRESH command may be issued to DIF in order to activate a new set of rules or to activate an existing set of rules which has been updated. The REFRESH command may specify the product name (ACC or SRS) and the rule name (the PDS member to be used). If no rule name is specified, then the name used is the name of the rule set that is *currently in use*. Note that this differs from the case where no rule name is specified on the START command. In that case, the default rule name (ACCRULES) is used. For the REFRESH command, however, omitting the rule name indicates that the existing rules are to be reread from the PARMLIB dataset and the updated copy used. An example of the REFRESH command is shown below. The rule name is specified so that there is no question about which member of the PARMLIB dataset should be used. The rule name specified may be the same as the name of the rules currently in use.

```
F DIF,REFRESH SRS SRSRULES
```

What Rules are Active Now?

The STATUS command instructs DIF to display which products are currently active, what release of each product is running, and what rule member is currently in use. The date and time at which the rule member was activated are also shown. For ACC and SRS, this date and time may be the most recent time at which a disk or tape device was varied online. This occurs because ACC and SRS rules are automatically REFRESHed whenever a device is varied online in order to update the pool definitions. The figure below shows an example of a STATUS command and the resulting output:

```
F DIF,STATUS  
DIF01180-I PRODUCT  VERSION  STATUS  RULES NAME  DATE      TIME  
DIF01181-I SRS      5.1.13  ACTIVE  SRSRULES    08/23/11 11:25  
DIF01182-I SRSVSAM  OPTION  ACTIVE  
DIF01181-I ACC      5.1.13  ACTIVE  ACCRULES    08/23/11 11:26  
DIF01182-I ACCVSAM  OPTION  ACTIVE  
DIF01181-I DIF      5.1.14  ACTIVE
```

It is often useful to know what the contents of the rule member(s) were at the time ACC / SRS was started or when the rules were last REFRESHed. This can be accomplished via a SET command. If the following command is issued to DIF:

```
F DIF,SET productname RULELIST ON
```

then the contents of the rule member will be written to the SYSPRINT dataset allocated to the DIF started task when the product is activated or when the rules are next REFRESHed. The 'productname' above may be either ACC or SRS, or may be '*' for all products. The SET command should be placed in the STARTxx member of the DIF PARMLIB dataset, so that it is executed automatically whenever DIF starts.

Creating an ACC / SRS Rule Set

Rules Language Syntax

In general, the syntax of the DIF rules language is obvious. There are some special constructs, such as the DEFPOOL statement, which specifies the characteristics of a disk or tape pool, and the DEFENV statement, which is used to modify defaults for the types of processing that may take place at various points during dataset allocation or space error recovery. However, most processing takes place as result of a series of IF / THEN / ELSE statements which occur following a DEFRULE statement. DEFRULE provides the name for a rule. One or more IF / THEN / ELSE statements follow a DEFRULE statement. These statements conclude with an action clause, usually a SET statement of the form 'SET DISKPOOL=poolname' to assign the dataset to an ACC allocation pool or SRS recovery pool. Complete syntax can be found in in the Dynamic Install Facility User's Guide and the Allocation Control Center / Space Recovery System User's Guide.

Rules Language Architecture

The sample rule member shown is divided into a number of sections. In reality, most statements, such as DEFFILT (FILTLIST), DEFPOOL, and DEFMSG may be placed anywhere within the rules. However, the sequence of statements in the sample rule member is similar to that widely used by DTS customers and provides a proven, easy-to-understand view of the rules logic. Also, as the rules language is executed top-to-bottom through the rules (DEFRULE statements) and the IF / THEN / ELSE statements that occur under the DEFRULEs, it is best to place the rules which apply to all datasets first, followed later by more narrowly-targeted rules, as shown in the sample.

Comments, Readability, and Maintainability

Although the DIF rules language is mostly free-form, it is of course best to follow standard coding conventions. That is, indentation and comments always make for easier-to-read and more-maintainable code. Comments in the DIF rules language are indicated by either an asterisk in the first column, which indicates that the entire statement following is a comment, or by a '/* */' pair, which encloses the comment. Blank lines are ignored by DIF rules processing, so they may be used as needed to provide clarity.

Continuation

Statements may be continued on the next line by breaking at a comma and continuing the statement on the next line, as shown in the example below. The quoted text fields in DEFMSG statements must be continued by placing a plus (+) sign at the end of the statement and continuing with the rest of the statement, in quotes, on the next line, as shown in the second example.

```
DEFPOOL PRDPOOL3
```

```
  VOLSER= (PRD001 , PRD002 , PRD003 ,  
          PRD004 , PRD005 , PRD6*)
```

```
DEFMSG MYMSG1 'THIS IS A LONG MESSAGE' +  
              'THAT IS CONTINUED ON A SECOND LINE'
```

Nested IF's

The DIF rules language allows nested IF statements. However, any nested IF *must* be contained within a DO-END pair, or the results will be unpredictable. The example below shows a valid nested IF. There is no limit to the amount of nesting, but more than 2 or 3 levels is difficult to understand and likely to lead to logic errors.

```
DEFRULE SOMERULE
  IF &QUAL2 = PROD THEN
    DO
      IF &JOBNAME = ABC*
        THEN SET &DISKPOOL = PRODPPOOL
    END
  ELSE DO
    IF &JOBNAME = ABC*
      THEN SET &DISKPOOL = TESTPOOL
    END
```

Flow of Control

There are a few important points to remember about the flow-of-control through the DIF rules language.

- 1.) The flow-of-control is from top-to-bottom unless altered by a CONTINUE parameter
- 2.) Rule processing is terminated (the rules are exited) *whenever the first IF / THEN / ELSE condition is satisfied* unless a CONTINUE parameter is present on the DEFRULE or on the THEN or ELSE clause.
- 3.) Once a pool has been assigned to a dataset via a SET DISKPOOL= clause, the rules are not executed again for that dataset unless RULES(ALWAYS) is specified on the DEFENV statement.

In order to continue rule processing after a successful IF / THEN / ELSE statement, the CONTINUE parameter must be present. It may be present on the rule definition (the DEFRULE statement), or it may be specified as part of the IF / THEN / ELSE, as shown in the examples below.

```
DEFRULE SKIPRULE CONTINUE(NEXTIF)
  IF &DDNAME = (SORTOU*, SORTOF*) THEN SET &SKIP_EXCP = YES
```

```
DEFRULE RLSERULE
  IF &ENV = DD
    &DSORG = (PS, ' ')
    &DISP1 = NEW
  THEN SET &RLSE = YES
    CONTINUE SCANNING AT NEXTRULE
```

A Sample Rule Member

The sample rule member shown in Appendix A provides a typical set of rules for ACC and SRS. It is designed to show the syntax needed for some of the most common uses of ACC and SRS. Because it is merely an example, it is unlikely that you will want to use it unchanged at your installation. There are many ACC and SRS functions which are not shown in the sample. However, you may want to use the sample rule member as a model for your own rules, after adapting it to fit your installation's needs for naming conventions, volume definitions, and execution requirements.

The sample rule member is divided into a number of sections. The following pages describe each of the sections of the sample rule member and the function of each statement.

Defaults and Initialization Section

The first section of the rules provides defaults and initialization values that will be used later. This section includes defaults for items such as the SMF record number to be used by SRS, the type of pattern matching to be used in the rules, and how often the rules are to be processed.

DEFPROD – Set Product-Wide Defaults

The DEFPROD statement provides defaults which are used unless overridden by subsequent statements such as DEFENV, DEFPOOL, and DEFMSG. In the sample rule set, there are three parameters present on the DEFPROD statement, each of which is described below.

SMF Recording

The first parameter, **SMF**, indicates that SRS is to write record type 221 (decimal) whenever SRS processing is performed for a dataset. The **LEVEL** subparameter indicates that SMF records are to be written for all SRS messages (I = 'informational'), rather than just action ('A'), warning ('W'), or error ('E') messages. These SMF records may be analyzed and reported on by the SRSSMF program included with SRS. SRSSMF provides reports on the number and types of out-of-space errors prevented by SRS, including the name of the dataset and job encountering the error. Of course, SMF recording must be enabled for the record type specified in order for SRS to successfully write SMF records.

Wildcards and Masking

The DIF rules support wildcards and masking characters. The default pattern matching characters are very simple and are adequate for most types of pattern matching. However, the defaults used by DIF are slightly different from those used in the DFSMS ACS routines. The default, which is explicitly specified in the sample rule set, is indicated by **OPTION(NOSMSMASK)**. In this case, ACC and SRS will use the default pattern matching values. The defaults are:

- *' – An asterisk indicates any sequence of characters. This may include multiple qualifiers for dataset names.
- \?' – A question mark indicates any single character.

If **OPTION(SMSMASK)** is specified, the above patterns are used for all variables except dataset names or VSAM cluster or component names. For these variables, the same pattern matching scheme used by the DFSMS ACS routines is used. That is, for dataset names, the following pattern matching characters are used:

- *' – A single asterisk indicates one or more characters or a single qualifier in a dataset name
- **' – A double asterisk indicates one or more characters or multiple qualifiers in a dataset name
- \%;' – A percent sign indicates a single character in a dataset name.

Sharing ACC / SRS Pools

ACC and SRS may share the same rule set, or each product may use a different set of rules. The rule set used by each product is specified on the **START** command issued to DIF when the product is started, or on the **REFRESH**

command when the rules are refreshed after a change. If ACC and SRS each use different rule sets, it is possible that the names of the pools (specified on the DEFPOOL statements in the rules) may overlap; that is, the ACC rules might define a pool called PROD, and the SRS rules might also define a pool called PROD. In this case, the **POOLNAME(INDEX(n))** parameter must be specified in order to allow ACC and SRS to determine which pool should be used when processing the dataset. In the example, the default value, POOLNAME(INDEX(1)) is specified. If a different, additional set of rules is used for SRS, and some of the pool names in the SRS rules are the same as those in the ACC rules, then POOLNAME(INDEX(2)) should be specified in the SRS rules.

DEFENV – Processing Defaults for Environment

The DEFENV statement provides additional default values for processing that is to take place in a particular ACC or SRS ‘environment’. ACC and SRS may receive control from the operating system at several points during processing of a dataset. For example, ACC receives control early in allocation processing (the ‘DD environment’) to examine or alter JCL or IDCAMS characteristics such as primary space or unit name. ACC may receive control again later (during ‘DADSM environment’, for example), in order to reject undesirable disk volumes, or at end-of-step processing (the ‘EOFMARK environment’) to write an end-of-file mark for an unopened non-SMS dataset. The DEFENV statement indicates whether the ACC / SRS rules are to be processed in a given environment and what types of datasets (tape or disk, new or old) are to be processed.

Environments may be active or inactive. If an environment is active, ACC or SRS will perform rule processing and take the actions specified in the rules for the dataset. In general, the necessary environments for a product are active by default. Therefore, the DEFENV statement is not a required statement. Once a pool has been assigned to a dataset by the rules, rule processing is not normally performed in other environments unless the RULES(ALWAYS) parameter is specified.

In the sample rule set, three DEFENV statements are specified. The first, **DEFENV DD PROCESS(TAPE)**, indicates that the rules are to be processed for newly allocated tape datasets as well as disk datasets. This statement is necessary because a later rule is used to reassign new values for certain types of tape units specified in the JCL.

The second DEFENV statement enables the DADSM environment via **MODE(ACTIVE)** and insures that rule processing is performed in this environment via **RULES(ALWAYS)**. This statement is necessary because a later rule is used to check whether a dataset being renamed meets installation standards. Since this environment is not normally active, it must be turned on via MODE(ACTIVE), and since the rules are not normally processed once a pool has been assigned to a dataset, we must insure that rule processing is performed for this dataset even if a pool was already assigned, and therefore RULES(ALWAYS) is specified.

The third DEFENV statement activate the INCSEC (Increase Secondary) environment for SRS and forces rule processing to occur in this environment. This is necessary because a later rule examines the number of extents occupied by a dataset and alters the secondary space amount if an excessive number of extents are in use. In order for this to occur, SRS must get control and must process the rules every time a dataset obtains a new extent, and thus RULES(ALWAYS) is necessary.

DEFFILT – Define Filter Lists

The DEFFILT statement provides a means of referring to a list of datasets or volumes in later rules. In the sample rule set, there are two filter lists. The first, INACTDSN, specifies a list of dataset name masks that will be used in a later rule. The second filter list, BVOLS, specifies a list of volumes which will later be used in a pool definition. Note that the BVOLS list includes all volumes beginning with the characters ‘BAX’ or ‘BAY’, except for two particular volumes, BAX001 and BAY001. FILTLIST may be used as a synonym for DEFFILT.

DEFDMYx – Process Obsolete Units / Volumes

These two statements allow ACC to intercept JCL or IDCAMS requests for volumes or units which no longer exist on the system, and convert them as necessary to allow the job to run. The first statement, DEFDMYU indicates that if no UNIT name is specified (\$NONE), or if an invalid UNIT name is specified (\$CHECK), then UNIT=SYSALLDA is to be used. This prevents jobs from going into allocation recovery processing or receiving JCL errors if UNIT names are missing or invalid. The second statement DEFDMYV, is used to prevent a problem

which can occur if an IDCAMS DEFINE statement contains an invalid or obsolete volume serial number. Normally, if such an IDCAMS DEFINE is specified, and DEFINE statement happens to include the UNIQUE parameter, then IDCAMS will attempt to dynamically allocate the specified volume, resulting in allocation recovery processing or an IDCAMS error. The \$CHECK parameter allows ACC to check for a nonexistent volume serial number and convert the unnecessary allocation to DUMMY, thus preventing the error.

Bypass-Product-Processing Section

Once defaults have been established, the first section of rules is specified. The DEFRULE statement defines the rule, which may be followed by one or more IF statements. Usually, the rules which have the widest scope, i.e., those which encompass all jobs or all datasets, should be specified first. In the sample rule set, some jobs and datasets are to be excluded from ACC and SRS control. The first IF statement indicates the jobs for which ACC/SRS processing is to be bypassed, and the EXIT parameter on the THEN clause indicates that rule processing is complete. The second statement uses a filter list to indicate datasets which are not to be processed. Note that the filter list name must include an ampersand to distinguish it as a variable. Finally, the third IF statement bypasses processing for SORTWK datasets, as allocation and use of these datasets is normally controlled by the system sort.

Bypass SRS ADDVOL Checks Section

The next section of the sample rules allows SRS to add volumes to some types of datasets that would normally not be eligible for ADDVOL processing. By default SRS does not attempt to add volumes to datasets which are written via the EXCP access method or which are processed with the NOTE or POINT macros. Processing is bypassed in these instances because SRS cannot be certain that the applications accessing the datasets support multiple volumes. In nearly all modern applications, however, multivolume support is automatic, and these restrictions can be bypassed. In the sample rule set, ADDVOL processing is allowed for sort and copy utility output datasets and for DB2 utilities, since it is certain that these types of datasets can be accessed if they are multivolume. Note the presence of the CONTINUE(NEXTIF) parameter on the DEFRULE statement. Since rule processing normally ends when the first IF / THEN / ELSE statement is satisfied, the CONTINUE parameter is required in order to process the remaining rules after setting the SKIP_EXCP and SKIP_NOTE or SKIP_POINT flags.

Standards Enforcement Section

One of the most important uses of ACC and SRS is to enforce installation standards. The next few rules provide standards for testing. Jobs or datasets are examined to see if they are in compliance with the installation's requirements, and actions such as issuing messages or altering allocation values are taken if necessary. Only a few very simple examples of standards enforcement are shown here. For more information and examples, the ACC / SRS User's Guide should be consulted.

Setting RLSE

The first rule, SETRLSE, allows idle, unused space at the end of a dataset to be released when the dataset is CLOSED. This is done by first testing to see if the dataset is being newly allocated (&ENV = DD and &DISP1 = NEW) by a batch job (&JOBTYPE = JOB) and if it is one of a particular group of datasets with a desired high level qualifier. If so, then the &RLSE variable is set to 'YES'. This has the same effect as coding 'RLSE' in the JCL.

Controlling Tape Allocations

The next rule, SELTAP2, is used to redirect allocations of tape datasets to a different type of unit. In this case, if a tape dataset is being allocated, and UNIT=3480 or UNIT=STAP was specified, the allocation is directed via the SET &TAPEPOOL statement to a 3490 unit and a message is written to the JES log of the job indicating the action that was taken. In this example, the message and the tape pool are specified by a DEFMSG statement and a DEFPOOL statement that immediately follow the DEFRULE. In other rules, the DEFMSG and DEFPOOL statements are not placed adjacent to the rule, but are placed in a group at the end of the rule set. Either means of coding is acceptable.

The ENDTAPE rule checks to see if the dataset being processed is a disk dataset, and if so, processing is allowed to continue. Otherwise, the rules are exited, preventing any subsequent rules from doing any processing of tape datasets.

Controlling RENAME processing

The next rule, TSTRENAM, enforces installation standards for the renaming of non-VSAM datasets. First, a test is made to see if a dataset is being renamed, i.e., if the DADSM environment is in effect and if the subenvironment name (&ENVS) is RENAME. If so, then a second, nested IF statement (placed within a DO-END group, as is required for nested ifs) checks to see whether the old and new high-level qualifiers of the dataset name are different. If so, the rename is disallowed via the SET &JCLFAIL = YES statement. A message is then issued indicating the reason for the failure. The MSGNOAL parameter on the WRITEMSG statement refers to the name of a DEFMSG statement specified at the end of the rules. If the old and new high-level qualifiers match, the rules are exited, as there is no need for any further processing to be performed in the DADSM environment.

Other Installation Standards

The last rule in this section, SELNOAL2, issues warnings if certain non-SMS datasets are being allocated on particular volumes. A check is made for the specified high-level qualifiers and for their SMS status, and a message issued if a standards violation is found.

Pooling Section

Following the Standards Enforcement section, the next few rules provide pooling, that is, volume selection, for dataset. Both SMS and non-SMS managed datasets may be assigned to pools; however, the volume serial numbers specified in the pool definitions are used only if the dataset is non-SMS-managed, or if the volumes are in the SMS storage group assigned to an SMS-managed dataset. In addition to specifying volume selection, the pool definitions indicate what types of SRS error prevention may be performed. Once a pool has been assigned to a dataset via a SET DISKPOOL statement, the rules are not processed again for the dataset unless RULES(ALWAYS) has been specified on a DEFENV statement.

The first rule in the pooling section, ALLOWX, allows only particular RACF userids to allocate to volumes in the pool called POOLX. A check is made for the volume serial number specified and for the RACF userid, and if a match is found, the pool is assigned. The pool definition contains the RESTRICT(NEW) parameter, indicating that this is a RESTRICTED pool. No other users may allocate any datasets in this pool.

The second rule, ALOCPool, simply uses the first IF statement to assign any dataset with a high level qualifier of 'PROD' to the PRODUCTION pool. This pool is defined later in the rules via a DEFPOOL statement.

The second IF statement in the ALOCPool rule assigns datasets with a second-level qualifier of TEST to the TEST pool, which is also specified by a DEFPOOL statement later in the rules. In addition, if a 'TEST' dataset has reached 50 or more extents, then the size of the secondary space requested is doubled. This is the reason for the DEFENV INCSEC statement earlier in the rules.

Pool Definition and Recovery Criteria Section

When a dataset has been assigned to a pool via the SET &DISKPOOL= or SET &TAPEPOOL= statements, the characteristics of the pool determine how the dataset is allocated.

The first pool definition is for the XPOOL. This is a restricted pool which contains only a few volumes. Any job which attempts to allocate a new dataset on volumes in this pool will fail, even if the job is otherwise uncontrolled by ACC, unless the pool is assigned via a SET &DISKPOOL statement. If the pool was not restricted (i.e., RESTRICT(NEW) was not specified, then other jobs not under ACC control, could allocate datasets on volumes in the pool.

The second pool is the TEST pool. If there is not enough room to allocate a dataset on a volume in the test pool, then an alternate pool, BPOOL, is used. Several types of SRS recovery (add-volume, set-secondary, and reduce-secondary) are allowed for datasets allocated to volumes in these pools.

The third pool is the PROD pool. Any volume beginning with the characters 'PR', 'X', or 'ZY' belongs to this pool. When allocating datasets in this pool, ACC will prefer the volume with the largest percentage of free space due to the ALG(MAXSPACE) parameter. In addition to the other types of SRS recovery, primary space reduction is allowed in this pool, and SRS may add more volumes to datasets in this pool than in the other pools. (20 as opposed to 15). If a duplicate-DSN situation is encountered, SRS will scratch the original dataset due to RECAT(SCRATCH).

Messages Section

This section includes the user-defined message which are issued by ACC or SRS as a result of an ISSUE WRITEMSG statement. The message text must be enclosed in quotes. The values of variables (prefixed by an ampersand) will be dynamically filled in at the time the message is issued.

Troubleshooting

Syntax Errors

Syntax errors are easily distinguished in the DIF rules language. Syntax errors are noted whenever the rules are dynamically compiled due to a START or REFRESH command. If a syntax error is discovered, message DIF90550-E is issued and the specified rules are not used. If a REFRESH command was issued, the existing rules continue to be used. The example below shows the messages issued as a result of a syntax error. Note that the DIF90550-E message indicates the line on which the syntax error was found. If there are INCLUDE statements in the rules, which cause other members to be included in line during rule processing, then the line number includes the lines in the INCLUDED rule member(s).

```
F DIF,REF ACC
DIF90550-E SYMBOLIC FIELD (JBNAME) INVALID ON STMT DEFRULE OPERAND EXPRESSION
LINE NUMBER 4
DIF20191-I RULE-CONVERSION MODULE DIFCRL HAD A RETURN CODE 12
```

Logic Errors

Logic errors cannot be determined by DIF. If datasets are being assigned to the wrong pools or are receiving the wrong allocation characteristics, or SRS error recovery is not being performed, then the problem is most likely a logic error in the way the rules are coded. Probably the most common type of logic error is failure to assign a pool to a dataset via a SET DISKPOOL= statement. If a logic error is suspected, an //ACCTRACE DD DUMMY statement should be added to the JCL. This statement will cause ACC or SRS rules processing to display the result of each statement in the rules when a dataset is processed. Examining the trace, which is written either to the JES SYMSGS or the IDCAMS SYSPPRINT dataset, will usually turn up the error.

Technical Support

DTS Software provides expert technical support directly from the developers of the products. Normal daily hours are from 9:00AM EST to 5:00PM EST Monday through Friday, with 24-hour support always on-call for emergencies. You can reach DTS Software Technical support by phone at 919-833-8426 or by email to support@dtssoftware.com.

Appendix A – The Sample ACC / SRS Rules

```
*****
* DEFAULTS AND INITIALIZATION SECTION *
*****

/*-----*/
/* DEFPROD STATEMENT PROVIDES PRODUCTWIDE DEFAULTS */
/* DEFENV STATEMENT PROVIDES ENVIRONMENT DEFAULTS */
/*-----*/
DEFPROD SMF(221 LEVEL(I)) POOLNAME(INDEX(1)) OPTION(NOSMSMASK)

DEFENV DD PROCESS(TAPE)
DEFENV DADSM MODE(ACTIVE) RULES(ALWAYS)
DEFENV INCSEC MODE(ACTIVE) RULES(ALWAYS)

/*-----*/
/* FILTER LISTS */
/*-----*/
FILTLIST INACTDSN
    INCLUDE(*.SYS*,HSM*,SYSA*)

FILTLIST BVOLS INCLUDE(BAX*,BAY*)
    EXCLUDE(BAX001,BAY001)

/*-----*/
/* PROCESSING FOR INVALID UNITS AND VOLUMES IN JCL/IDCAMS */
/*-----*/
DEFDMYU ($CHECK,$NONE) UNIT(SYSALLDA)

DEFDMYV ($CHECK)

*****
* BYPASS-PRODUCT-PROCESSING SECTION *
*****

DEFRULE BYPRULE
    IF &JOBNAME = (SYSX3*,ABFQ%1) THEN EXIT
    IF &DSNAME = &INACTDSN THEN EXIT
    IF &DDNAME = SORTWK* THEN EXIT

*****
* BYPASS SRS-ADDDVOL CHECKS SECTION *
*****
DEFRULE SKIPS CONTINUE(NEXTIF)
    IF &DDNAME = (SORTOU*,SORTOF*) THEN SET &SKIP_EXCP = YES
    IF &PGM = (SORT, SYNCSORT, ICEMAN, IGHRCO00, IERRCO00,
        IEBGENER, ICEGENER, FASTVSAM) THEN SET &SKIP_EXCP = YES
    IF &PGM = DSNUTILB
        THEN SET &SKIP_NOTE = YES
        SET &SKIP_POINT = YES

*****
* STANDARDS-ENFORCEMENT SECTION *
*****
*
```

```

* ENABLE IDLE SPACE RELEASE FOR SOME DATASETS
*
DEFRULE SETRLSE CONTINUE (NEXTIF)
  IF &ENV = DD
    &JOBTYPE=JOB
    &DISP1 = NEW
    &HLQ=(ZF,NV,AZ,CO,VZF,VNV,VAZ,VCO)
    THEN SET &RLSE = YES

*
* CONVERT ALL UNIT=3480 OR UNIT=STAP TO UNIT=3490
*
DEFRULE SELTAP2
  IF &ENV = DD
    &UNITNAME = (3480,STAP)
    THEN SET &TAPEPOOL = ACTTAP1
        ISSUE WRITEMSG (TAPEMSG)

DEFPPOOL ACTTAP1
  &UNITNAME = 3490

DEFRULE ENDTAPE
IF &UNITTYPE NE DASD THEN EXIT /* NO FURTHER PROC OF TAPES */

*
* ALLOW PROD RENAME ONLY IF HLQ OF NEW DATASET MATCHES OLD HLQ
*
DEFRULE TSTRENAM
  IF &ENV = DADSM
    &ENVS= RENAME
    THEN DO
      IF &HLQ NE &NEWDSN_HLQ
        THEN SET &JCLFAIL = YES
        ISSUE WRITEMSG (RENAMER)
      ELSE EXIT
    END

*
* WARN ABOUT ALLOCATION OF SOME HLQS ON CERTAIN VOLUMES
*
DEFRULE SELNOAL2
  IF &HLQ = (VZF,VNV,VAZ)
    &SMS = NO
    &VOLSER=(M08089,M08085,M08132)
    THEN SET &JCLFAIL = YES
        ISSUE WRITEMSG (MSGNOAL)

*****
* POOLING SECTION *
*****
*
* ALLOW ONLY SOME RACF USERS TO ALLOCATE TO NONPROD 'X' VOLS
*
DEFRULE ALLOWX
  IF &RUSER = PGR*
    &VOLSER = (X00001,X00006,X00009)

```

```

        THEN SET &DISKPOOL = XPOOL

*
* ALLOCATE DATASETS TO PRODUCTION OR TEST POOLS AND SET SRS RECOVERY
*
DEFRULE ALOCPOOL
  IF &HLQ = PROD
    THEN SET &DISKPOOL = PRODPOOL

  IF &QUAL2 = TEST
    THEN DO
      SET &DISKPOOL = TESTPOOL
      IF &DATASET_EXTENT GT 50 THEN SET &SSPACE = &SSPACE*200%
      ISSUE WRITEMSG(XTINCRD)
    END

*****
* POOL DEFINITION AND RECOVERY CRITERIA SECTION          *
*****
DEFPOOL XPOOL RESTRICT(NEW)
  &VOLSER = (X00001,X00006,X00009)

DEFPOOL TESTPOOL ALTPPOOL(BPOOL)
ADDVOL(YES MAXVOL(15))
SETSEC(PERCENT(50))
REDUCES(YES PERCENT(10) LIMIT(1))
VOLSER=TST*

DEFPOOL BPOOL
ADDVOL(YES MAXVOL(15))
SETSEC(PERCENT(50))
REDUCES(YES PERCENT(10) LIMIT(1))
VOLSER=&BVOLS

DEFPOOL PRODPPOOL (ALG(MAXSPACE))
  SETSEC(PERCENT(100))
  REDUCEP(PERCENT(10) LIMIT(1))
  REDUCES(YES PERCENT(10) LIMIT(1))
  ADDVOL(YES MAXVOL(20))
  RECAT(SCRATCH)
  &VOLSER = (PR*,X*,ZY*)

*****
* MESSAGES SECTION          *
*****
DEFMSG TAPEMSG 'TAPE UNIT &UNITNAME SPECIFICATION OVERRIDDEN'
DEFMSG MSGNOAL 'WARNING: ALLOCATION OF &DSNAME INVALID ON &VOLSER'
DEFMSG XTINCRD 'EXCESSIVE EXTENTS FOR &DSNAME - SECONDARY INCREASED'
DEFMSG RENAMER '&DSNAME MAY NOT BE RENAMED TO &NEWDSN'

```