# StarTool® I/O Optimzer

## User's Guide

# Contents

# Welcome to StarTool IOO

Product Description
: Thank you for choosing StarTool® IOO (I/O Optimizer), the intelligent buffer management tool for IBM z/OS mainframes. StarTool IOO optimizes total system throughput based on a proven knowledgebase of performance rules, custom user override settings, and real-time detection of actual file usage. StarTool IOO dynamically manages both VSAM and non-VSAM I/O buffering in batch and on-line processing environments.

Document Objectives
: This document describes the functions and features of StarTool IOO and provides user operation instructions. It will help you:

- Start, stop, or query the status of StarTool IOO.

- Understand the default optimization tables and their intended workloads.

- Tailor buffer management settings dynamically during IOO execution.

- Customize optimization rules if necessary.

Audience
: The information in this document is intended for IBM mainframe systems programmers and systems engineers with responsibility for managing global z/OS performance.

Change Bars
: Change bars in the left margin identify substantive changes that have been made to this manual in this release.

# Before You Begin

Review the Readme file for new information and corrections that became available after this document was published. You can download the Readme file and other documentation in the IOO documentation suite from the Micro Focus Supportline website.

# Conventions

Terminology
: This document adopts the following terminological conventions:

- **StarTool IOO** and **IOO** refer to StarTool I/O Optimizer.

- *somnode* refers to your library High Level Qualifier (HLQ).

Textual Conventions
: The following textual conventions identify clickable hyperlinks throughout this document:

| Convention | Meaning |
|---|---|
| Blue | Dynamic cross-reference within this document. |
| `Blue monospaced` | Hypertext link to the World Wide Web. |

Additional textual conventions are used in technical contexts to highlight or delimit special information:

| Convention | Meaning |
|---|---|
| Greater-than symbol > | Separates items in a sequence of menu, submenu, or command selections on a GUI client. Example:<br>**Start > Programs > Micro Focus > *product_name*** |
| Vertical bar  \| | Inside braces, separates mutually exclusive parameter values. Example: `SETBLK={NO|YES}` |
| Curly braces  { } | Required parameter value is to be selected from a list. Example: `SETBLK={NO|YES}` |
| Square braces  [ ] | Optional parameter, usually selected from a list. Example: `[,BUFFER=(TRACKS,xx)]` |
| Ellipsis  ... | Optional unlimited repetitions in a list. |
| **Bold** | Panel title, data entry field name, or menu option. Example:<br>**DCF Master Control Panel** |
| UPPERCASE | Key name or keystroke combination in a data entry context. Examples: PF3, the ENTER key. |
| Monospaced | Source code, JCL, XML tags, or message text. Also marks keyboard data entries. Example: `//JOBNAME    JOB` |
| MONOSPACED UPPERCASE | Member name, library name, command name, or required value in source code, JCL, or parameter list. Example: `IOOINIT` |
| MONOSPACED UPPERCASE UNDERLINE | Default value in a parameter list. Example: `SETBLK={`<u>`NO`</u>`|YES}` |
| *Monospaced italics* | Variable element in data set names, member names, or parameter lists. Parameter example: `STATTHLD=`*`xxx,`* where *xxx* indicates a 3-byte value. Data set example: `STRIO.V`*`v`*`R`*`r`*`M`*`m`*`,` where *v* = version number, *r* = release, and *m* = modification level. (Non-italicized characters are literals.) |
| Monospaced mixed case<br><br>Examples:<br><br>■   `ABC*`<br><br>■   *`yyyy/mm/dd`*<br><br>■   `C'`*`aa`*`'`<br><br>■   `X'`*`nn`*`'` | Pattern for a field value or parameter you enter. Number of characters is significant. Punctuation such as slashes or single-quotes must be reproduced in the position shown.<br><br>Examples:<br><br>■   Search string containing the literal ABC and the asterisk (*) wildcard character.<br><br>■   Date with four-digit year, two-digit month, and two-digit day separated by required slashes. Leading zeroes required. Example: `2010/01/01`<br><br>■   Alphanumeric character string, two characters long, in default character encoding, delimited by required single quotes, and prefixed by literal C. Example: `C'AB'`<br><br>■   Hexadecimal number, two digits long, delimited by required single quotes, and prefixed by literal X, where *n* = 0 to F. Example: `X'FF'` |

# Documentation

IBM Reference

*VSAM Demystified* (SG24-6105) is an IBM Redbook that provides conceptual and reference information about IBM's Virtual Sequential Access Method (VSAM) for z/OS. It is useful for understanding many of the optimization principles employed by StarTool IOO.

Product Documentation

The StarTool IOO product documentation consists of the Readme file and the following manuals:

| Title | Description |
|---|---|
| *Serena® SER10TY™ User's Guide* | Licensing information and instructions on how to apply StarTool IOO license keys using SER10TY. |
| *StarTool® IOO Installation and Setup* | Installation instructions and migration tips for StarTool IOO. |
| *StarTool® IOO User's Guide* | Functions and operational information for StarTool IOO. |
| *StarTool® IOO Messages* | Lists and describes StarTool IOO messages. |

## Accessing the Documentation

All product documentation is available for download from the Micro Focus Supportline website at <u>https://supportline.microfocus.com/</u>.

### *Using the PDF Documentation*

Downloading Adobe Reader

All electronic manuals are delivered in Adobe Portable Document Format (PDF). To view PDF documents, you need Adobe® Reader® software. Version 7.0.5 or higher is recommended. Adobe Reader is freely available from the Adobe Web site at:

> <u>http://get.adobe.com/reader/</u>.

**TIP** Be sure to download the *full version* of Adobe Reader. The more basic version does not include the cross-document search feature.

PDF Document Features

The PDF manuals and the Adobe Reader include the following features to simply use of the documentation and to make information easy to find:

- **Bookmarks.** All of the online manuals contain predefined bookmarks that make it easy for you to quickly jump to a specific topic. By default, the bookmarks appear to the left when you open an online manual.

- **Links.** Cross-reference links within an online manual enable you to jump to other sections within the manual and to other manuals with a single mouse click. These links appear in blue.

- **Printing.** While viewing a manual, you can print the current page, a range of pages, or the entire manual.

- **Advanced search.** Starting with Version 6, Adobe Reader includes an advanced search feature that enables you to search across multiple PDF files in a specified directory. (This is in addition to using any search index created by Adobe Catalog — see step 3 below.)

Multiple Document Search

Adobe Reader Version 6 and higher supports multiple document search. To search across multiple PDF documents concurrently, perform the following steps:

**1** In Adobe Reader, select Edit > Advanced Search (or press SHIFT+CTRL+F).

**2** In the text box, enter the keywords for which you want to search.

**3** Select the **All PDF Documents in** option, then browse to select the folder in which you want to search. (If you have a document open that has an Adobe Catalog index attached, you can leave the **In the index named...** option selected to search across all the manuals in the index.)

**4** Optionally, select one or more of the additional search options, such as **Whole words only** or **Case-Sensitive**.

**5** Click the **Search** button.

For more information, see Adobe Reader's online help.

# Customer Support

- **Online Support** — The Micro Focus SupportLine website at http://supportline.microfocus.com provides an easy way for customers to manage their support cases, download software patches, or search the Knowledgebase for the latest technical information about our products at any time.

- A user ID and password are required to access the Micro Focus SupportLine website. To obtain a user ID and password, sign up online or contact Customer Support by email or telephone.

- **Telephone Support** — Telephone contact numbers for Customer Support are posted by region at the following URL:

  https://www.microfocus.com/support-and-services/serena/

  No password is needed to access Customer Support contact information.

- **Email Support** — Customers can send support requests by email to:

  customercare@microfocus.com.

**Authorized Support Representative** — Some support contracts specify that support calls will be centrally managed within the customer organization. This practice enables related issues to be aggregated across multiple customer sites or functional groups, and it ensures accountability for the satisfactory resolution of open issues. You should determine whether you have such a support contract before contacting Customer Support. If you do have such a contract, submit your support request to the designated support manager within your organization and arrange for that person to contact Customer Support.

# Chapter 1
# Controlling IOO

This chapter describes how to start and stop IOO, change the IOO environment, and control IOO system intercepts.

# IOO Startup and Shutdown

You can start IOO in two different modes:

- Using a persistent address space.

  Once IOO startup is complete, the IOO start-up task will go into a wait state and wait for operator commands. The address space created to start IOO will remain active.

- Using a transient address space.

  Once IOO startup is complete, the IOO start-up task will terminate and the address space will be deleted. IOO itself will remain active.

Follow your installation standards about using persistent or transient address spaces for active products.

## Controlling IOO Using a Persistent Address Space

Use the following operator command to start IOO with a persistent address space:

`S IOO,OPTIONS={K0|K1|K2|K3|K4|K5}`

The OPTIONS parameter identifies to IOO the global options that are to be used. "IOO Start-up Options" on page 10 describes global options K0 through K5.

Use the following operator command to stop IOO:

`P IOO`

## Controlling IOO Using a Transient Address Space

Use the following operator command to start IOO with a transient address space:

`S IOOINIT,OPTIONS={K0|K1|K2|K3|K4|K5}`

The OPTIONS parameter identifies to IOO the global options that are to be used. "IOO Start-up Options" on page 10 describes global options K0 through K5.

Use the following operator command to stop IOO. This command starts the shutdown task:

`S IOOSTOP`

# IOO Start-up Options

The IOO start-up options consist of a 2-character value that you enter with the OPTIONS operand of the IOO JCL procedures as follows:

- JCL procedure IOO (starts IOO and requests that IOO use a persistent address space).

- JCL procedure IOOINIT (starts IOO and requests that IOO use a transient address space).

- JCL procedure IOOMODS (requests that a new start-up option be implemented without having to stop the product).

The following table shows the supported start-up options and the resultant processing:

| Option | Reblock Non-VSAM | VSAM Optimization Strategies | | |
|--------|------------------|------------------------------|---|---|
| | | COBOL RMODE-31 Buffers | Non-COBOL RMODE-31 Buffers | Allow LSR Bufferspaces > 16 MB |
| K0 | No | X | | |
| K1 | No | X | X | |
| K2 | No | X | X | X |
| K3 | Yes | X | | |
| K4 | Yes | X | X | |
| K5 | Yes | X | X | X |

Examples:

**1** Starting IOO with an OPTIONS value of K4 will start IOO and authorize it to do the following:

- Perform non-VSAM reblocking.

- Use RMODE-31 buffers where possible for both COBOL and non-COBOL programs.

**2** Starting IOO with an OPTIONS value of K2 will start IOO and authorize it to do the following:

- Do not perform non-VSAM re-blocking.

- Use RMODE-31 buffers where possible for both COBOL and non-COBOL programs, and allow the bufferspace for LSR component optimizations to exceed 16 MB.

IOO uses the start-up option that you specify to identify:

- The global options table that IOO is to use. The start-up option is appended to the string ULTIGB to form the 8-character module name of the global options table that IOO will load from the load library that you specify in the STEPLIB DD statement of the IOO, IOOINIT, or IOOMOD procedures. The global options table may be one of the ones distributed with IOO (as identified in the table below) or it may identify a user-customized table.

**NOTE** Appendix A, "Rules Tables" described how to create a customized global options table if the options table that is shipped with StarTool IOO does not satisfy your needs.

- The DCF rules table that IOO is to use. The start-up option is appended to the string DCF to form the 5-character name of the member containing the DCF rules that IOO is to use. This member must reside in the JCL library. The member may be one of the ones distributed with IOO or it may identify a user-customized set of DCF rules.

For example, if startup option K3 is specified, IOO will attempt to use global options table ULTIGBK3 and DCF rules member DCFK3.

## Global Options Tables Distributed with IOO

The following members of the STRIOO.V*v*R*r*M*m*.LOAD library identify the global options tables that are associated with the IOO start-up options:

| IOO Start-Up Option | STRIOO.V*v*R*r*M*m*.LOAD Library Member |
|---|---|
| K0 | ULTIGBK0 |
| K1 | ULTIGBK1 |
| K2 | ULTIGBK2 |
| K3 | ULTIGBK3 |
| K4 | ULTIGBK4 |
| K5 | ULTIGBK5 |

## DCF Rules Tables Distributed with IOO

The following members of the STRIOO.V*v*R*r*M*m*.JCL library identify the DCF rules tables that are associated with the IOO start-up options:

| IOO Start-Up Option | STRIOO.V*v*R*r*M*m*.JCL Library Member |
|---|---|
| K0 | DCFK0 |
| K1 | DCFK1 |
| K2 | DCFK2 |
| K3 | DCFK3 |
| K4 | DCFK4 |
| K5 | DCFK5 |

# Changing the IOO Environment

Once IOO has been successfully started, IOO will maintain the requested environment until it is modified. IOO supports two types of environment modification:

- Changing the global options (see "Changing the Global Options" on page 12).

- Changing the DCF rules (see "Changing the DCF Rules" on page 13).

## Changing the Global Options

You can modify the global options used by IOO by doing one of the following:

- By stopping IOO (with the appropriate shutdown command as described in "IOO Startup and Shutdown" on page 10) and then restarting IOO with the desired global option.

- By using the IOOMODS JCL procedure to load and activate the new global option. You can start the IOOMODS JCL procedure with the following operator command:

```
S IOOMODS,OPTIONS={K0|K1|K2|K3|K4|K5}
```

The OPTIONS parameter identifies to IOO the global option that is to be used.

The IOOMODS JCL procedure will cause the DCF rules that correspond to the selected global option to be activated.

## Changing the DCF Rules

The Dynamic Control Facility (DCF) rules are loaded each time that IOO is started and each time that the IOOMODS JCL procedure is used to load and activate new global options.

You can also take one of the following actions to load DCF rules:

- Issue the following operator command:

  ```
  S IOODCF,DCFIN='dsn'
  ```

  where *dsn* is the name of the data set with the DCF input. The IOODCF JCL procedure will replace the existing DCF rules with those loaded from the specified data set. For example:

  ```
  S IOODCF,DCFIN='STRIOO.VvRrMm.JCL(DCFK3)'
  ```

  specifies that DCF input is to be taken from the STRIOO.V*v*R*r*M*m*.JCL(DCFK3) data set.

- From within the DCF ISPF interface, select option 3 on the main panel (panel DCF001). On the resulting panel (panel DCF009), enter the name of the data set containing the desired DCF rules. Press Enter and IOO will submit the command to activate the specified rules.

The DCF is described in detail in Chapter 3, "Dynamic Control Facility".

# Controlling the IOO System Intercepts

To successfully process each access to a data set, IOO intercepts selected SVCs. Enter the following command to deactivate the IOO intercepts:

```
F IOO,OFF
```

Note that this command will disable the IOO intercepts; it will neither remove nor delete them.

Enter the following operator command to reactivate the IOO intercepts:

```
F IOO,ON
```

# Displaying IOO Status

You can display the processing status of IOO at any time by typing the following operator command on the system console:

```
S IOOINQY{,OCLS=A}
```

The optional OCLS parameter identifies the output class.

# Dynamic Memory Management (DMM) Considerations

At data set OPEN time StarTool IOO analyzes the potential region size for both 24- and 31-bit storage. Taking into account Local System Queue Area (LSQA) requirements, IOO dynamically maximizes region limits to give executing programs access to as much storage as possible.

In maximizing region limits, IOO uses the values of the following keyword operands from the IOO global table:

| Global Table Keyword Operand | Description |
|---|---|
| ELIMIT= | DMM: Limit value > 16MB |
| IEALIMIT= | DMM: Limit value < 16MB |
| EREGION= | DMM: Region size > 16MB |
| REGNHWM= | DMM: Region size < 16MB |
| LSQA= | DMM's LSQA reserved amount |
| LSQALEXT= | DMM's LSQA minimum contiguous space |

If SETRGN=YES is specified in the global table source or GBT SETRGN=YES is specified in the DCF input (indicating that the DMM feature is to be activated), two actions are required:

1. Define a facility to RACF with a discrete profile name of your own choosing as follows:

```
RDEFINE FACILITY profilename ACC(NONE)
```

2. Update the global table source to set the FACILITY keyword to the name of the profile specified in the REDEFINE above:

```
FACILITY=profilename
```

The default value for *profilename* is $SERENA.IOO.DMM.

When you start StarTool IOO, and SETRGN=YES is specified in the global table source or GBT SETRGN=YES is specified in the DCF input, IOO will ask RACF if the facility profile name specified by the FACILITY keyword exists. If the profile does exist, DMM will be activated. If the profile does not exist DMM will not be activated.

To see the effect of DMM, you can run a job that qualifies for IOO optimization with the following JCL statement:

```
//SZITRDMM DD DUMMY
```

Refer to Appendix A, "Rules Tables" on page 79 for more information about global table operands.

# Chapter 2
# Controlling a Job within IOO

This chapter describes how to bypass optimization, request optimization, and force the use of a specific optimization rule.

# Bypassing Optimization

IOO provides two ways to bypass optimization:

- With Dynamic Control Facility (DCF) rules (see "Bypassing Optimization with DCF" on page 16).

- With the execution JCL (see "Bypassing Optimization with JCL" on page 18).

## Bypassing Optimization with DCF

IOO will bypass the optimization of a data set when the selected DCF rule requests the bypassing of the desired IOO component. The selected DCF rule will be the first DCF rule that matches all coded criteria.

For example, given the following DCF rules:

```
BYPASS=ALL          Rule #1
   PGM=TEST7

BYPASS=ALL          Rule #2
   DD=SYSUT2

SELECT=ALL          Rule #3
   JOB=MYJOB1
```

Note the following results:

- Any job executing program TEST7 will be bypassed from optimization by rule #1. No additional DCF rules will be searched. Thus, a job named MYJOB1 would be bypassed from optimization when it executes program TEST7 even though there is a select rule for the job in rule #3. IOO will not process rule #3 because it has already matched rule #1.

- Any data set accessed with DDname SYSUT2 will be bypassed from optimization by rule #2. No additional DCF rules will be searched. Thus, a job named MYJOB1 would be bypassed from optimization when it accesses DDname SYSUT2 even though there is a select rule for the job in rule #3. IOO will not process rule #3 because it has already matched rule #2.

- Job MYJOB1 will be selected for optimization by rule #3 except when:

  • It executes program TEST7 (in which case it will be bypassed when IOO matches rule #1).

  • It accesses DDname SYSUT2 (in which case it will be bypassed when IOO matches rule #2).

To request the bypassing of optimization with the DCF rules, either:

- A new rule containing sufficient criteria to identify the optimization to be bypassed must be added to the DCF rules (prior to any select rule whose selection criteria would match).

  or:

- An existing DCF bypass rule must be updated such that it contains the needed criteria.

In both cases, you should use the DCF ISPF interface to update the DCF rules.

### Scope of the Bypass

The scope of the bypass will depend in part upon the selection criteria that you identify. All specified selection criteria must be matched for the DCF rule to be selected.

#### Example #1

If you specify both a job name and a DDname (as in the following rule):

```
BYPASS=ALL
    JOB=MYJOB1
    DD=SYSUT2
```

both criteria must be matched for the DCF rule to be selected. In this case, DDname SYSUT2 will be bypassed only when it is accessed by job MYJOB1. DDname SYSUT2 will be optimized if any other job accesses it. Any other DDname will be optimized if job MYJOB1 accesses it.

#### Example #2

If you want to bypass all optimizations for any job named TEST37 (no matter what data set is accessed or what step is involved), code only the job name as the selection criteria:

```
BYPASS=ALL
    JOB=TEST37
```

This is an example of a job-level bypass.

#### Example #3

If you want to bypass all optimizations for step STEP6 within job TEST37 (no matter what data set is accessed), code the job name and the step name as the selection criteria:

```
BYPASS=ALL
    JOB=TEST37
    STEP=STEP6
```

This is an example of a step-level bypass. If the job name were not specified in the DCF rule, IOO would select the rule for any step name STEP6, no matter what job name was used.

#### Example #4

If you want to bypass all optimizations for data set PAYROLL.BACKUP (no matter where it is accessed):

```
BYPASS=ALL
    DSN=PAYROLL.BACKUP
```

This is an example of a system-level bypass (for the indicated data set).

### Updating the DCF Input Data

Refer to for specific instructions on starting the DCF ISPF interface.

Refer to "Command/Option 1: Insert a Rule into the Start of an IOO-DCF Table" on page 29 for specific instructions on defining a single new rule and inserting it in front of any existing rules.

Refer to "Command/Option 2: Maintain an IOO-DCF Table" on page 39 for specific instructions on updating an existing rule within the DCF rules.

# Bypassing Optimization with JCL

IOO provides several ways to bypass optimization with JCL. Optimization can be bypassed for an entire job, an entire step, or selected DDnames.

### Requesting a Job-Level Bypass

Change the job name to one having one of the following prefixes:

- NOIOO

- NOSIO

- NOULT

This will bypass optimization for the entire job.

### Requesting a Step-Level Bypass

There are two ways to accomplish this:

**1**   Change the step name to one having one of the following prefixes

- NOIOO

- NOSIO

- NOULT

This will bypass optimization for the entire step.

**2**   Add the DD_SOFF DDname to the step JCL. For example, if the default value of ULT@SOFF was taken for the DD_SOFF global table value, adding the following DD statement to the step JCL will cause the bypassing of optimization for the entire step:

```
//ULT@SOFF DD DUMMY
```

> **NOTE**  ULT@SOFF is the default DDname that is shipped for the global table DD_SOFF operand. See "DCF IOO Global Table Override Records" on page 48 for more information about the DD_SOFF operand.

### Requesting a Selective Bypass

To bypass IOO optimization for only a portion of a step, insert the DD_OFF DDname into the step JCL at the point you wish to turn off optimization. In the following example, assume that the default value of ULT@OFF was taken for the DD_OFF global table value:

```
//S010      EXEC PGM=MYPROG
//IN001     DD   DISP=SHR,DSN=MY.INPUT.FILE1
//ULT@OFF   DD   DUMMY
//OUT01     DD   DISP=SHR,DSN=MY.OUTPUT.FILE1
```

```
//OUT02    DD   DISP=SHR,DSN=MY.OUTPUT.FILE2
```

> **NOTE** ULT@OFF is the default DDname that is shipped for the global table DD_OFF operand. See "DCF IOO Global Table Override Records" on page 48 for more information about the DD_OFF operand.

Because DD statement IN001 falls outside the domain of any IOO DDname (it precedes the ULT@OFF DDname in the JCL), optimization of it depends upon the presence of a matching DCF SELECT rule. If no matching DCF rule is found, optimization is not performed.

Because DD statements OUT01 and OUT02 both fall within the domain of an IOO DD_OFF DDname, IOO would bypass optimization for both.

# Requesting Optimization

IOO provides two ways to request optimization:

- With Dynamic Control Facility (DCF) rules (see "Requesting Optimization with DCF" on page 19).

- With the execution JCL (see "Requesting Optimization with JCL" on page 21).

## Requesting Optimization with DCF

IOO will select the optimization of a data set when the selected DCF rule requests the selection of the desired IOO component. The selected DCF rule will be the first DCF rule matching all coded criteria.

For example, given the following DCF rules:

```
SELECT=ALL        Rule #1
   PGM=TEST7

SELECT=ALL        Rule #2
   DD=SYSUT2

BYPASS=ALL        Rule #3
   JOB=MYJOB1
```

Note the following results:

- Rule #1 will select any job that executes program TEST7 for optimization. No additional DCF rules will be searched. For example, a job named MYJOB1 would be selected for optimization when it executes program TEST7 even though there is a bypass rule for the job in rule #3. IOO will not process rule #3 because it has already matched rule #1.

- Rule #2 will select any job that accesses any data set with DDname SYSUT2 for optimization. No additional DCF rules will be searched. For example, a job named MYJOB1 would be selected for optimization when it accesses DDname SYSUT2 even though there is a bypass rule for the job in rule #3. IOO will not process rule #3 because it has already matched rule #2.

- Rule #3 will bypass MYJOB1 from optimization except when:

- It executes program TEST7 (in which case it will be selected when IOO matches rule #1).

- It accesses DDname SYSUT2 (in which case it will be selected when IOO matches rule #2).

To request the selection of optimization with the DCF rules, either:

■ A new rule containing sufficient criteria to identify the optimization to be selected must be added to the DCF rules (prior to any bypass rule whose selection criteria would match).

Or:

■ An existing DCF SELECT rule must be updated such that it contains the needed criteria.

In both cases, you should use the ISPF interface to DCF to update the DCF rules.

### *Scope of the Select*

The scope of the select will depend in part upon the selection criteria that you identify. All specified selection criteria must be matched for the DCF rule to be selected.

### **Example #1**

If you specify both a job name and a DDname (as in the following rule):

```
SELECT=ALL
    JOB=MYJOB1
    DD=SYSUT2
```

both criteria must be matched for the DCF rule to be selected. In this case, only DDname SYSUT2 will be selected when it is accessed by job MYJOB1. When any other job accesses DDname SYSUT2, it will not match this rule. When MYJOB1 accesses any other DDname, it will not match this rule.

### **Example #2**

If you want to optimize everything for a job named TEST37 (no matter what data set is accessed or what step is involved), code only the job name as the selection criteria:

```
SELECT=ALL
    JOB=TEST37
```

This is an example of job-level optimization.

### **Example #3**

If you want optimize everything for step STEP6 within job TEST37 job (no matter what data set is accessed) code the job name and the step name as the selection criteria:

```
SELECT=ALL
    JOB=TEST37
    STEP=STEP6
```

This is an example of step-level optimization.

Note that if the job name were not specified in the DCF rule, IOO would select the rule for any step name STEP6 no matter what job name was used.

**Example #4**

If you want to optimize all accesses of data set PAYROLL.BACKUP (no matter where it is accessed), code the following as selection criteria:

```
SELECT=ALL
    DSN=PAYROLL.BACKUP
```

This is an example of system-level optimization (for the indicated data set).

### *Updating the DCF Input Data*

Refer to "Starting the ISPF Interface to DCF" on page 27 for specific instructions on starting the ISPF interface to DCF.

Refer to "Command/Option 1: Insert a Rule into the Start of an IOO-DCF Table" on page 29 for specific instructions on defining a single new rule and inserting it in front of any existing rules.

Refer to "Command/Option 2: Maintain an IOO-DCF Table" on page 39 for specific instructions on updating an existing rule within the DCF rules.

# Requesting Optimization with JCL

IOO provides several ways to request optimization with JCL. You can request optimization for:

- An entire job (see "Requesting Job-Level Optimization" on page 21).

- An entire step (see "Requesting Step-Level Optimization" on page 21).

- Selected DDnames within a step (see "Requesting a Selective Optimization" on page 22).

### *Requesting Job-Level Optimization*

Changing the job name to one having one of the following prefixes will select optimization for the entire job:

- IOO

- ULT

### *Requesting Step-Level Optimization*

There are two ways to accomplish step-level optimization:

**1** Change the step name to one having one of the following prefixes:

- • IOO

- • ULT

**2** Add the DD_SON DDname to the step JCL. For example, if the default value of ULT@SON was taken for the DD_SON global table value, addition of the following DD statement to the step JCL will optimize the entire step:

```
//ULT@SON  DD DUMMY
```

> **NOTE** ULT@SON is the default DDname that is shipped for the global table DD_SON operand. See "DCF IOO Global Table Override Records" on page 48 for more information about the DD_SON operand.

### Requesting a Selective Optimization

To select IOO optimization for only a portion of a step, insert the DD_ON DDname into the step JCL at the point you wish to turn on optimization. In the following example, assume that the default value of ULT@ON was taken for the DD_ON global table value:

```
//S010     EXEC PGM=MYPROG
//IN001    DD   DISP=SHR,DSN=MY.INPUT.FILE1
//ULT@ON   DD   DUMMY
//OUT01    DD   DISP=SHR,DSN=MY.OUTPUT.FILE1
//OUT02    DD   DISP=SHR,DSN=MY.OUTPUT.FILE2
```

> **NOTE** ULT@ON is the default DDname that is shipped for the global table DD_ON operand. See "DCF IOO Global Table Override Records" on page 48 for more information about the DD_ON operand.

Because DD statement IN001 falls outside the domain of any IOO DDname (it precedes the ULT@OFF DDname in the JCL), optimization of it depends upon the presence of a matching DCF SELECT rule. If no matching DCF rule is found, optimization is not performed.

Because DD statements OUT01 and OUT02 both fall within the domain of an IOO DD_ON DDname, IOO would select both for optimization.

# Forcing the Use of a Specific Optimization Rule

Sometimes it is desirable to force IOO to use specific optimization parameters. For example, for a VSAM data set you may wish to either force the use of Local Shared Resources (LSR) or force the use of Non-Shared Resources (NSR).

You can force the use of a specific optimization rule with DCF or JCL.

## Forcing a Specific Optimization Rule with DCF

To force the use of a specific optimization rule, a RULE override must be added to the matching DCF rule. Because the optimization rule specified by the override will be used by all files matching the selected DCF rule, the DCF rule should be as granular as possible.

For example, given the following DCF rule:

```
SELECT=ALL
   JOB=MYJOB1
   RULE=VSMLSR31
```

IOO will attempt to use optimization rule VSMLSR31 for all files accessed by job MYJOB1 (whether the file is VSAM or not).

Use the ISPF interface to DCF to either define or update a SELECT rule that uniquely identifies the file in question. Include in the DCF rule a RULE override specifying the desired optimization rule.

The following field in panel DCF003 is where the name of the desired optimization rule should be entered:

```
Name of rule table entry to select ==> _
```

Position the cursor to the input area and press <PF1> to see a list of the provided rules.

contains instructions on starting the ISPF DCF interface.

contains instructions on inserting a single new rule in front of any existing DCF rules.

contains instructions on updating existing DCF rules.

# Forcing a Specific Optimization Rule with JCL

IOO uses two special DDnames to identify situations where a specific IOO rule has been requested for a file.

A DD statement having a DDname in the format of SZIRULE*x* (where *x* is any valid EBCDIC character) instructs IOO that a request is being made for a specific optimization rule. The DDname on the next DD statement will be used as the name of the optimization rule. The DD statement after that is the target DD statement.

For example:

```
//S010     EXEC PGM=MYPROG
//SZIRULE1 DD   DUMMY                <-- See note #1
//VSMLSR31 DD   DUMMY                <-- See note #1
//INFILE   DD   DISP=SHR,VSAM.FILE1  <-- See note #1
//SZIRULE$ DD   DUMMY                <-- See note #2
//VSMNSR24 DD   DUMMY                <-- See note #2
//OUTFILE  DD   DISP=SHR,VSAM.FILE2  <-- See note #2
```

- Note 1: The SZIRULE1 DDname instructs IOO that the DDname from the next DD statement (VSMLSR31) is to be used as the name of the optimization rule when the next DD statement (DDname INFILE) is optimized. When DDname INFILE is opened, IOO will use optimization rule VSMLSR31 and DDname INFILE will be optimized using VSAM Local Shared Resources with RMODE 31 buffers.

**NOTE** See "Criteria Records" on page 50 for more information about the VSMLSR31 operand of the RULE= criteria record.

- Note 2: The SZIRULE$ DDname instructs IOO that the DDname from the next DD statement (VSMNSR24) is to be used as the name of the optimization rule when the next DD statement (DDname OUTFILE) is optimized. When DDname OUTFILE is

opened, IOO will use optimization rule VSMNSR24 and DDname OUTFILE will be optimized using VSAM Non-Shared Resources with RMODE 24 buffers.

**NOTE** See "Criteria Records" on page 50 for more information about the VSMLSR24 operand of the RULE= criteria record.

# Chapter 3

# Dynamic Control Facility

The Dynamic Control Facility (DCF) is an ISPF interface to IOO that allows you to easily control how and when IOO optimizes access to files.

# Overview

During IOO startup, IOO reads the specified DCF input stream and creates a table of rules that will identify file accesses that are to be either selected for IOO optimization or bypassed from IOO optimization (based upon the selection criteria included in the DCF rules).

Once IOO optimization has been selected by a DCF rule, IOO searches the optimization rules table to find a rule containing suitable optimization parameters for that access. Individual optimization parameters within the selected optimization rule will be overridden by any corresponding override value provided in the DCF rule.

DCF rules are processed on a first-match basis. That is, DCF will accept the first matching rule that is found. For example, given the following DCF definitions:

```
SELECT=ALL          <--- Rule #1
    JOB=MYJOB1
BYPASS=ALL          <--- Rule #2
    JOB=MYJOB1
    STEP=STEP1
```

If job MYJOB1 were to access a data set in step STEP1, the data set would be selected for optimization by IOO (even though a bypass rule for that step exists in rule #2) because IOO would match the criteria in rule #1 and would not process rule #2.

In order to bypass processing for step STEP1 for job MYJOB1, the DCF rules should be coded as follows:

```
BYPASS=ALL          <--- Rule #1
    JOB=MYJOB1
    STEP=STEP1
SELECT=ALL          <--- Rule #2
    JOB=MYJOB1
```

# The ISPF Interface to DCF

The ISPF interface to DCF allows a user to perform the following activities:

- Customize the appearance of the DCF panels.

- Insert an IOO-DCF bypass/select rule into the start of a DCF definition file. The operations staff can use this feature to quickly select or bypass the use of IOO.

- Maintain an IOO-DCF table. The interface will allow a user to interactively update an existing DCF file. Existing DCF rules can be edited, deleted, or replicated. New rules can be added. The resultant DCF rules can be written to either back to the original input file or to a different output file.

- Activate a set of DCF definitions.

## Using Online Help

You can request panel-level help for any DCF panel by placing the cursor anywhere on the panel and pressing PF1.

You can request field-level help for any field on any DCF panel by placing the cursor in the data entry area for the field and pressing PF1.

# Starting the ISPF Interface to DCF

Take the follow actions to start the ISPF interface to DCF:

**1**   On ISPF screen 6, issue the command **IOODCF**.

The IOO-DCF Master Control Panel (DCF001) is displayed:

```
DCF001 V315  ----------  IOO-DCF Master Control Panel  ------------------------
Command/Option ===>                                         Use PF3/END to exit

   DCF input DSN   ===>
   DCF output DSN  ===>

               -- Select one of the following options --    Userid   - USER1
                                                             Time     - 11:42
                                                             Terminal - 3278
   0. Set Panel Attributes                                   Applid   - IOO

   1. Dynamically insert an IOO-DCF Bypass/Selection rule
      into the start of an IOO-DCF table.

   2. Maintain an IOO-DCF table.

   3. Activate an IOO-DCF table.
```

**2**   On the main panel (DCF001) enter:

| Field | Description |
|---|---|
| Command/Option | Enter one of the following values depending on what you want to do: 0, 1, 2, or 3 |
| DCF Input DSN | Enter the name of the rules table that you want to process. The target data set name is one of the following members of the STRIOO.V$v$R$r$M$m$.JCL distribution library:<br><br>DCFK0<br>DCFK1<br>DCFK2<br>DCFK3<br>DCFK4<br>DCFK5<br><br>where the specified member name corresponds to the start-up option K0 through K5. An example DCF Input DSN name is that identifies the rules table corresponding to start-up option K3 is:<br><br>*hlq*.STRIOO.V$v$R$r$M$m$.JCL(DCFK3) |
| DCF Output DSN | Enter the name of the output data set to which you want to write the modifications. If you leave this field blank, the contents of the DCF Input DSN are overwritten with your changes. |

Command/Options 0 through 3 are described in the following sections:

- Command/Option 0, see "Command/Option 0: Set Panel Attributes" on page 28.

- Command/Option 1, see "Command/Option 1: Insert a Rule into the Start of an IOO-DCF Table" on page 29.

- Command/Option 2, see "Command/Option 2: Maintain an IOO-DCF Table" on page 39.

- Command/Option 3, see "Command/Option 3: Activate an IOO-DCF Table" on page 46

## Command/Option 0: Set Panel Attributes

Take the following actions to change DCF panel display attributes if you want to change them:

**1** On panel DCF001, select option 0.

**2** Type in the DCF Input DSN.

**3** Press Enter.

The Attribute Definition Panel (IOOATTR1) is displayed:

```
IOOATTR1 V315  ---------  Attribute Definition Panel  -----------------------
Command ===>                                          Use PF3/END to exit
Userid   - USER1                                      Terminal - 3278
Time     - 12:41     -- Select Your Default Attributes --  Applid   - IOO


Intensity H = High   Color W = White  B =     Blue      Hilite U = Underscore

         L = Low           G = Green  Y = Yellow        R = Reverse video
                           R = Red    P = Pink          B = Blinking
                           T = Turquoise                N = Normal

Text fields     Intensity ===> L  Color ===> G  Hilite ===> N
Text titles     Intensity ===> L  Color ===> T  Hilite ===> N
Text pointers   Intensity ===> L  Color ===> T  Hilite ===> N

Panel titles    Intensity ===> L  Color ===> T  Hilite ===> R
Panel names     Intensity ===> H  Color ===> T  Hilite ===> N

Input fields    Intensity ===> H  Color ===> Y  Hilite ===> N
Error messages  Intensity ===> L  Color ===> R  Hilite ===> R
Output fields   Intensity ===> L  Color ===> Y  Hilite ===> N

Special fields  Intensity ===> H  Color ===> P  Hilite ===> N
```

Fill in the panel with the desired attribute definition codes and press PF3 to make the desired changes to the display attributes of DCF panels.

## Command/Option 1: Insert a Rule into the Start of an IOO-DCF Table

**NOTE** See the following sections for more information about coding DCF rules:

- "DCF Input" on page 47

- "General Rule Coding Guidelines" on page 48

- "Comment Records" on page 48

- "Scope Records" on page 49

- "Criteria Records" on page 50

- "Optimization Rule Override Records" on page 53

- "Global Table Override Records" on page 62

Take the following actions to define a new rule and insert it in front of the specified rule definition table:

**1** On panel DCF001, select option 1.

**2** Type in the name of the desired DCF Input and Output DSNs, as desired.

**3** Press Enter. The IOO-DCF Rule Definition panel (DCF002) is displayed:

```
DCF002 V315  ------------- IOO-DCF Rule Definition  -------------------------
Command/Option ===>                                      Use PF3/END to exit

Select a rule type ===>   . S = Select
                            B = Bypass

Select IOO components that will apply to this rule :

                     IOO-Block  ===>  .    Use Y to select the
                     IOO-VSAM   ===>  .    desired IOO component(s).
                     IOO-LSR    ===>  .

Define the selection criteria :
    Job name      ===>          . Step name    ===>          .
    JobStep name  ===>          . Program name ===>          .
    DD name       ===>          .
    Data set name ===>                                          .

Miscellaneous definitions :
    Allow started task selection by this rule ? ===>   ( Y or N )
```

**4** Fill in this panel as follows:

| Field | Description |
|---|---|
| Select a rule type | Specify S to select the target data sets for optimization by IOO.<br><br>Specify B to exclude the target data sets from IOO optimization. |
| IOO-Block | Specify Y to apply block processing to the target rule.<br><br>IOO optimizes the blocksizes for QSAM, BSAM, and sequential EXCP I/O. It improves QSAM, BSAM, and SORT processing.<br><br>If you leave this field blank, IOO-Block processing for any data set access matching the indicated selection criteria will not be impacted by this rule. |
| IOO-VSAM | Specify Y to apply VSAM processing to the target rule.<br><br>IOO optimizes all VSAM, regardless of access strategy. It builds VSAM resource pools for shared resources, defers direct PUTs until buffer is needed, allows Sequential Insert Strategy and reduces Start I/Os.<br><br>If you leave this field blank, IOO-VSAM processing for any data set access matching the indicated selection criteria will not be impacted by this rule. |
| IOO-LSR | Specify Y to apply VSAM LSR processing to the target rule.<br><br>This provides augmented local shared resource pools for batch LSR VSAM. |
| Job Name | Specify the job name criteria. ALL matches all jobs in the system and is identical to JOB=*. You can specify the variable-length * wildcard character and fixed-length ? wildcard as a prefix or a suffix.<br><br>For example, TESTJOB* selects (or bypasses) all jobs whose names begin with the characters TESTJOB. |
| Step Name | Specify the step name criteria. ALL matches all steps in the system regardless of Job Name and is identical to STEP=*. You can specify the variable-length * wildcard character and fixed-length ? wildcard as a prefix or a suffix.<br><br>For example, S01* selects (or bypasses) all job steps whose names begin with the characters S01. |
| JobStep Name | Specify the jobstep criteria. ALL matches all steps in the system and is identical to JOBSTEP=*. You can specify the variable-length * wildcard character and fixed-length ? wildcard as a prefix or a suffix.<br><br>For example, S01? selects (or bypasses) all job steps whose names begin with the characters S01*x*, where *x* is any single character. |

| Field | Description |
|-------|-------------|
| Program Name | Specify the program name criteria. ALL matches all program names in the system and is identical to PGMNAME=*. You can specify the variable-length * wildcard character and fixed-length ? wildcard as a prefix or a suffix.<br><br>For example, TEST59B selects (or bypasses) all steps where program TEST59B is executed. |
| DD Name | Specify the DDname criteria. ALL matches all DDnames in the system, and is identical to DD=*. You can specify the variable-length * wildcard character and fixed-length ? wildcard as a prefix or a suffix.<br><br>For example, SYSUT? selects (or bypasses) all DDnames whose names begin with SYSUT*x*, where *x* is any single character. |
| Data Set Name | Specify the data set name criteria. ALL matches all DSNs in the system, and is identical to DSN=*. You can specify the variable-length * wildcard character and fixed-length ? wildcard as a prefix or a suffix.<br><br>For example, MYDSN* selects (or bypasses) all data sets whose names begin with MYDSN. |

**5** Respond as follows to the prompt:

> **Allow started task selection by this rule?**

- **Y** means that the rule will be applied to started tasks.

- **N** means that the rule will not be applied to started tasks.

The value you specify overrides the started task rule in the target global table. If you leave this field blank, the started task rule in the target global table is not changed.

**6** Press Enter to validate the input. The IOO-DCF Rule Table Overrides (DCF003) panel is displayed:

```
DCF003 V315  ----------  IOO-DCF Rule Table Overrides  -----------------------
Command/Option ===>                                         Use PF3/END to exit

Name of rule table entry to select ==>           . (Omit to let IOO decide)

VSAM overrides
 - NSR  data buffer count ==>          .         - VSAM Dynamic Mode Selection
 - NSR index buffer count ==>          .            High-Water-Mark
 - NSR buffer space       ==>          .         - Count ==>     Percent ==>
 - NSR buffering bias     ==>       .         Non-VSAM overrides
      (Enter DIR, DYN, or SEQ)                 - Maximum blocksize ==>       .
 - LSR maximum buffer space :                   - Max LBI blocksize ==>
    - Index ==>         Data ==>        . - Buffer number     ==>        .
 - VSAM record level interface settings :      Trace options (Use S to select)
    ==>   : Force interface activation         OPN ==>   .   OPND ==>   .
    ==>   : Force interface de-activation      DCF ==>   .   DCFD ==>   .
    ==>   : No DMS (Dynamic Mode Selection)    JFC ==>   .   JFCD ==>   .
    ==>   : No DBM (Dynamic Buffer Management) VSM ==>   .   VSMD ==>   .
    ==>   : No DLP (Dynamic LSR Positioning)   VSI ==>   .    DEX ==>   .
    ==>   : No DBT (Dynamic Buffer Translation) DMM ==>  .    CSR ==>   .

Do you want to request advanced overrides ? ===>   ( Y or N )
```

**7** Fill in the fields on this panel as follows:

| Field | Description |
|---|---|
| Name of rule table entry to select | You can specify one of the following optimization rule names:<br><br>■ IOOQSAMN: Selects rebuffering only.<br>■ IOOQSAMY: Selects rebuffering and reblocking. You must specify the blocksize.<br>■ VSMLSR24: (VSAM) Use LSR rmode24 buffers.<br>■ VSMLSR31: (VSAM) Use LSR rmode31 buffers.<br>■ VSMNSR24: (VSAM) Use NSR rmode24 buffers.<br>■ VSMNSR31: (VSAM) Use NSR rmode31 buffers. For tables K2 and K5, IOO will use LSR bufferspaces exceeding 16MB.<br><br>Leave this field blank to have IOO select the rule. |
| NSR data buffer count | Specify the number (in kilobytes) of VSAM NSR data buffers. You can specify a maximum value of 32767. You can optionally specify the suffix K or KB. For example, 256 or 256KB.<br>If you leave this field blank, no override will be performed. |

| Field | Description |
|---|---|
| NSR index buffer count | Specify the number (in kilobytes) of VSAM NSR index buffers. You can specify a maximum value of 32767. You can specify the suffix K or KB. For example, 32 means 32KB.<br><br>If you leave this field blank, no override is performed. |
| NSR buffer space | Specify the total amount of VSAM NSR buffer space to a maximum of 128MB. You can specify the suffix K, KB, M, or MB.<br><br>If you leave this field blank, no override is performed.<br><br>If you omit the suffix, the actual value in bytes is used. For example:<br><br>■  124KB means 124 kilobytes.<br><br>■  124MB means 124 megabytes.<br><br>■  124 means 124 bytes. |
| VSAM Dynamic Mode Selection High-Water-Mark Count | The count of records that will be consecutively processed sequentially using LSR before IOO dynamically changes modes. You can enter a value from 0 to 255. For example, 12 means 12 records.<br><br>If you leave this field blank, no override is performed. |
| VSAM Dynamic Mode Selection High-Water-Mark Percent | The percentage of allocated buffers to be processed sequentially using LSR before IOO dynamically changes modes. You can specify a value from 0 to 99. For example, 85 means 85 percent.<br><br>If you leave this field blank, no override is performed. |
| NSR buffering bias | NSR buffering bias can have one of the following values:<br><br>■  DIR: IOO uses a biasing technique to allocate buffers on direct opens.<br><br>■  DYN: IOO uses a biasing technique to allocate buffers on dynamic opens.<br><br>■  SEQ: IOO uses a biasing technique to allocate buffers on sequential opens. |
| Maximum blocksize | You can specify a non-VSAM maximum blocksize value (in kilobytes) up to 32767 (kilobytes). You can optionally specify the suffix K or KB. If you omit the suffix, kilobytes is assumed. For example:<br><br>■  32720K means 32,720 kilobytes.<br><br>■  32720 means 32,720 kilobytes.<br>If you leave this field blank, no override is performed. |

| Field | Description |
|---|---|
| Max LBI blocksize | You can specify the LBI maximum blocksize can be a value up to 256 (kilobytes). You can optionally specify the suffix K or KB. For example:<br><br>■  128KB means 128 kilobytes.<br><br>■  128 means 128 kilobytes. |
| LSR maximum buffer space Index | You can specify the LSR maximum index buffer space up to 128MB. You can optionally specify the suffix K, KB, M, or MB. If you omit the suffix, the actual value in bytes is assumed. For example:<br><br>■  128KB means 128 kilobytes.<br><br>■  128M means 128 megabytes.<br><br>■  128 means 128 bytes.<br>If you leave this field blank, no override is performed. |
| LSR maximum buffer space Data | You can specify the LSR maximum data buffer space up to 128MB. You can optionally specify the suffix K, KB, M, or MB. If you omit the suffix, the actual value in bytes is assumed. For example:<br><br>■  128K means 128 kilobytes.<br><br>■  128MB means 128 megabytes.<br><br>■  128 means 128 bytes.<br>If you leave this field blank, no override is performed. |
| Buffer number | You can specify a buffer number value for non-VSAM overrides up to 255.<br>If you leave this field blank, no override is performed. |
| **VSAM Record Level Interface Settings**: | |
| Force interface activation | Specify S to force activation of the VSAM record level interface.<br>If you leave this field blank, no override is performed. |
| Force interface deactivation | Specify S to force deactivation of all functions of the VSAM record level interface.<br><br>Specifying S for this field is the same as specifying S for No DMS, No DBM, No DLP, and No DBT.<br>If you leave this field blank, no override is performed. |
| No DMS (Dynamic Mode Selection) | Specify S to force DMS deactivation for the VSAM record level interface.<br><br>DMS changes from LSR to NSR after a certain amount of sequential I/O.<br><br>If you leave this field blank, no override is performed. |

| Field | Description |
|-------|-------------|
| No DBM (Dynamic Buffer Management) | Specify S to force DBM deactivation for the VSAM record level interface.<br><br>DBM detects if BAL programs issuing I/O or other factors exist and forces the program to use below the 16 megabyte line (24-bit) buffers.<br><br>If you leave this field blank, no override is performed. |
| No DLP (Dynamic LSR Positioning) | Specify S to force DLP deactivation in the VSAM record level interface.<br><br>DLP performs any initial positioning needed by NSR coded programs that have been optimized using LSR.<br><br>If you leave this field blank, no override is performed. |
| No DBT (Dynamic Buffer Translation) | Specify S to force DBT deactivation in the VSAM record level interface.<br><br>DBT performs buffer translation for above-the-line buffers so programs coded in 24-bit mode can use above-the-line (31-bit) buffers.<br><br>If you leave this field blank, no override is performed. |
| **Trace Options:** | |
| OPN Trace Option | Specify S to trace OPN related information. If you leave this field blank, OPN tracing is not performed.<br><br>Tracing should only be done if a problem occurs and Customer Support asks you to use this trace facility. |
| OPND Trace Option | Specify S to trace OPND related information. If you leave this field blank, OPND tracing is not performed.<br><br>Tracing should only be done if a problem occurs and Customer Support asks you to use this trace facility. |
| DCF Trace Option | Specify S to trace DCF related information. If you leave this field blank, DCF tracing is not performed.<br>DCF trace option is the same as specifying the following DD statement in any job to be optimized by StarTool IOO:<br><br>`//SZITRDCF DD DUMMY`<br><br>Tracing should only be done if a problem occurs and Customer Support asks you to use this trace facility. |
| DCFD Trace Option | Specify S to trace DCFD related information. If you leave this field blank, DCFD tracing is not performed.<br>DCFD trace option is the same as specifying the following DD statement in any job to be optimized by StarTool IOO:<br><br>`//SZITDCFD DD DUMMY`<br><br>Tracing should only be done if a problem occurs and Customer Support asks you to use this trace facility. |

| Field | Description |
|---|---|
| JFC Trace Option | Specify S to trace JFC related information. If you leave this field blank, JFC tracing is not performed.<br><br>Tracing should only be done if a problem occurs and Customer Support asks you to use this trace facility. |
| JFCD Trace Option | Specify S to trace JFCD related information. If you leave this field blank, JFCD tracing is not performed.<br><br>Tracing should only be done if a problem occurs and Customer Support asks you to use this trace facility. |
| VSM Trace Option | Specify S to trace VSM related information. If you leave this field blank, VSM tracing is not performed.<br><br>Tracing should only be done if a problem occurs and Customer Support asks you to use this trace facility. |
| VSMD Trace Option | Specify S to trace VSMD related information. If you leave this field blank, VSMD tracing is not performed.<br><br>Tracing should only be done if a problem occurs and Customer Support asks you to use this trace facility. |
| VSI Trace Option | Specify S to trace VSI related information. If you leave this field blank, VSI tracing is not performed.<br><br>Tracing should only be done if a problem occurs and Customer Support asks you to use this trace facility. |
| DEX Trace Option | Specify S to trace DEX related information. If you leave this field blank, DEX tracing is not performed.<br><br>Tracing should only be done if a problem occurs and Customer Support asks you to use this trace facility. |
| DMM Trace Option | Specify S to trace DMM related information. If you leave this field blank, DMM tracing is not performed.<br><br>Tracing should only be done if a problem occurs and Customer Support asks you to use this trace facility. |
| CSR Trace Option | Specify S to trace CSR related information. If you leave this field blank, CSR tracing is not performed.<br><br>Tracing should only be done if a problem occurs and Customer Support asks you to use this trace facility. |

**8**   Respond to the following prompt at the bottom of the panel:

**Do you want to request advanced overrides ? ===>**

- If you respond **N** or leave the field blank and press Enter, the IOO-DCF Rules Comments (DCF006) panel is displayed. (This panel is described on the next page.)

- If you respond **Y** and press Enter, the IOO-DCF Advanced Rule Overrides (DCF004) panel is displayed:

```
DCF004 V315  ---------  IOO-DCF Advanced Rule overrides  ----------------------
Command/Option ===>                                    Use PF3/END to exit

Override the advanced options for this rule:

   Action    Enter Y to force the use of the option, or N to bupass it's use.
             Omit the option to use the value coded in the selected IOO
             optimization rule.

 ===>    . ALCBUF31 (For VSAM) Allow assembler programs to use RMODE31 buffers.
                    (Non-VSAM) Request RMODE31 buffers.
 ===>    . ACCNVRPL Allow selection of RPLs not marked as VSAM in the RPL type.
 ===>    . EXCEEDCA Allow bufferspace to exceed the size of a control-area.
 ===>    . IGNHURBA Ignore the highest-used-RBA (hurba) value extracted from the
                    catalog when computing buffer counts.
 ===>    . REUSESHR For an LSR application, reuse the original LSR shrpool
                    number if all LSR shrpools have been used.
 ===>    . USEVRPLM For an LSR application, use the buffer counts from the
                    application BLDVRP as the minimum buffer count values.
```

Instruction for filling in this panel are provided on the panel, so they are not repeated here.

**9** Fill in the panel as appropriate and press Enter. The IOO-DCF Rules Comments (DCF006) panel is displayed:

```
DCF006 V315  -------------  IOO-DCF Rule Comments  ---------------------------
Command/Option ===>                             Use PF3/END to exit


Add up to 10 lines of comments to the rule :

===>
===>
===>
===>
===>
===>
===>
===>
===>
===>
```

You can enter comment text or leave this panel blank. Each line of comments must begin with the * character. The comments section is associated with the target rule, and is written to the DCF file ahead of the rule components.

**10** Press PF3 when you complete this panel. Keep pressing PF3 until you reach the Replace File (DCFREP) pop-up panel (where IOO will prompt you to confirm that you want to update the indicated output data set):

```
DCFREP          Replace file          Use PF3/END
Command ===>


 Press Enter to replace the member or data set specified.


 Press End or Cancel to discard changes.


 USER1.STRIOO.V3R1M5.CUSTOM.JCL(DCFK3)

```

**11** Press Enter if the data set name is correct. If you enter **EXIT** or **CANCEL**, your new rule will be discarded.

Take the following actions to activate the newly updated rules:

**1** Upon return to panel DCF001, select option 3 (Activate an IOO-DCF table) and press Enter. The IOO-DCF Restart Input DSN Definition Panel (DCF009) is displayed:

```
DCF009 V315  ----  IOO-DCF Restart Input DSN Definition Panel  ---------------
Command/Option ===>                                          Use PF3/END to exit

Press Enter to acivate IOO-DCF input data set

 DCF DSN ===> 'USER1.STRIOO.V3R1M5.CUSTOM.JCL(DCFK3)'          .
 IOO SSN ===> U315 .
```

**2** Fill in this panel as follows:

| Field | Description |
|-------|-------------|
| DCF DSN | Enter (or verify) the name of the data set containing the DCF input that you want to activate. |
|  | Press enter to request that IOO replace its active DCF control table with one populated from the DCF rules contained in the indicated data set. |
|  | Press PF3 to cancel the request. |
| IOO SSN | Enter the name of the target IOO subsystem. |
|  | The default value is U315. This value is set in the IOO and IOOSTART JCL procedures that are distributed with StarTool IOO 3.1.4. Users can change this value; however, changing the value is not recommended. |

**3** Press Enter to request that the modified DCF rules be reloaded in the current IOO environment. The following message is displayed:

```
SZIB004I IOO DCF activation command submitted.
***
```

**4**   Press Enter to continue. The IOO-DCF Master Control Panel (DCF001) is displayed.

# Command/Option 2: Maintain an IOO-DCF Table

**NOTE**  See the following sections for more information about coding DCF rules:

- "DCF Input" on page 47

- "General Rule Coding Guidelines" on page 48

- "Comment Records" on page 48

- "Scope Records" on page 49

- "Criteria Records" on page 50

- "Optimization Rule Override Records" on page 53

- "Global Table Override Records" on page 62

Take the following actions to update an existing rule within the DCF rules:

**1**   Bring up the ISPF interface to DCF. (See "Starting the ISPF Interface to DCF" on page 27 for details.)

**1**   Select option 2 on the IOO-DCF Master Control Panel (DCF001) and press Enter. The IOO-DCF Rule Selection (DCF005) panel is displayed. This panel displays a list of the DCF rules found in the indicated DCF input data set:

```
DCF005 V315 -------------- IOO-DCF Rule Selection -------------- Row 1 of 7
Command/Option ===>                                          SCROLL==> CSR

DSN of DCF definitions ===> 'USER1.STRIOO.V3R1M5.JCL(DCFK3)'

  S = Select and update a rule.        Use GLOBAL command to edit
  D = Delete a rule.                    global table overrides.
  I = Insert a new rule.
  R = Repeat a rule.

           -IOO Components-
 Sel   Type    Block VSAM LSR  Job Name  StepName  Job Step  Program   DDName
 ---   ------  ----- ---- ---  --------  --------  --------  --------  --------
       BYPASS    Y    Y    Y                                 IEBCOPY
       BYPASS    Y    Y    Y                                 DFH*
       SELECT    Y    Y    Y   IOO*
       SELECT    Y    Y    Y             IOO*
       SELECT    Y    Y    Y   ULT*
       SELECT    Y    Y    Y             ULT*
       BYPASS    Y    Y    Y   ALL
****************************** Bottom of data ******************************
```

**2**    Fill in the fields on the panel as follows:

| Field | Description |
|---|---|
| Command/Option | Enter one of the following commands:<br><br>■  **Global** - Allows editing of the global DCF overrides.<br><br>■  blank - Continue to the Sel input field.<br><br>■  **Cancel** - Discards changes to the DCF file. |
| Sel | Enter one of the following selection codes in the Sel input field that precedes a line item:<br><br>■  **S**: Select a rule and be guided through a series of ISPF panels for setting DCF criteria, overrides, and trace options.<br><br>■  **D**: Delete a rule from the DCF.<br><br>■  **I**: Insert a new rule to the DCF.<br><br>■  **R**: Repeat a rule as a model for a new rule. The new rule can then be selected for editing.<br><br>Select only one line item at a time and press Enter. DCF will take you through a series of panels so you can fully define each selected DCF rule (described below). |

**3**    Press Enter to process your selections.

**4**    If you specify the **Global** command/option on the IOO-DCF Rule Selection (DCF005) panel, the IOO-DCF Global Table Overrides (DCF007) panel (part 1 of 2) is displayed:

```
DCF007 V315  ---------  IOO-DCF Global Table Overrides  ----------------------
Command/Option ===>                                       Use PF3/END to exit


Enter your IOO global table overrides (part 1 of 2) :

   (Leave a field blank to not override the associated data element.)

 - IOO product options
    - Perform reblocking  ?        ===>     Enter Y to activate the option.
    - Perform rebuffering ?        ===>         N to deactivate the option.
    - Perform region optimization ? ===>    Omit to use global table value.

 - IOO message options         Enter W to use WTO.
    - Re-blocking  ===>    .        - L to use WTL.
    - Re-buffering ===>    .        - J to direct message to JESYSMSG DD.
                                    - N to not produce a message.
    - TSO messages ===>    .    Enter Y to produce messages in TSO.
                                    N to not produce messages in TSO.


 - VSAM bufferspace : Min ===>            . Max ===>           .
 - VSAM hiper-space : Min ===>            . Max ===>           .
```

**5**    Fill in the fields on the panel as follows. If you leave a field blank, the corresponding global table value will not be overridden:

| Field | Description |
|---|---|
| **IOO Product Options** | |
| Perform reblocking? | Enter one of the following values:<br><br>■   **Y**: Indicates that the reblocking option will be activated.<br><br>■   **N**: Indicates that the reblocking option will be deactivated. |
| Perform rebuffering? | Enter one of the following values: values:<br><br>■   **Y**: Indicates that the rebuffering option will be activated.<br><br>■   **N**: Indicates that the rebuffering option will be deactivated. |
| Perform region optimization? | Enter one of the following values: values:<br><br>■   **Y**: Indicates that region optimization will be activated.<br><br>■   **N**: Indicates that the region optimization will be deactivated. |
| **IOO Message Options** | |
| Reblocking Messages | Directs messages about reblocking to a specific output destination, as follows:<br><br>■   **W**: Use Write To Operator.<br><br>■   **L**: Use Write To Log.<br><br>■   **J**: Write to JESYSMSG DDname.<br><br>■   **N**: Do not write the message. |
| Rebuffering Messages | Directs messages about rebuffering to a specific output destination, as follows:<br><br>■   **W**: Use Write To Operator.<br><br>■   **L**: Use Write To Log.<br><br>■   **J**: Write to JESYSMSG DDname.<br><br>■   **N**: Do not write the message. |

| Field | Description |
|-------|-------------|
| TSO Messages | Specifies how to control the production of IOO messages within a TSO address space:<br><br>■ **Y**: Allow message to be produced. The message destination will be determined by the options selected for reblocking and rebuffering messages.<br><br>■ **N**: Do not allow messages to be produced. |
| VSAM Bufferspace: Min | Specify a value between 0 and 128MB or leave blank. You can optionally specify the suffix K, KB, M, or MB. If you omit the suffix, the specified value in bytes is used. For example:<br><br>■ 128K means 128 kilobytes.<br><br>■ 128MB means 128 megabytes.<br><br>■ 128 means 128 bytes.<br><br>The VSAM minimum bufferspace sets the minimum amount of VSAM bufferspace that IOO allocates per file. |
| VSAM Bufferspace: Max | Specify a value between 0 and 128MB or leave blank. You can optionally specify the suffix K, KB, M, or MB. If you omit the suffix, the specified value in bytes is used. For example:<br><br>■ 128K means 128 kilobytes.<br><br>■ 128MB means 128 megabytes.<br><br>■ 128 means 128 bytes.<br><br>The VSAM maximum bufferspace sets the maximum amount of VSAM bufferspace that IOO allocates per file. |

| Field | Description |
|---|---|
| VSAM Hiperspace: Min | Specify a value between 0 and 128MB or leave blank. You can optionally specify the suffix K, KB, M, or MB. If you omit the suffix, the specified value in bytes is used. For example:<br><br>■ 128K means 128 kilobytes.<br><br>■ 128MB means 128 megabytes.<br><br>■ 128 means 128 bytes.<br><br>The VSAM minimum hiperspace sets the minimum amount of hiperspace storage that IOO allocates per file. |
| VSAM Hiperspace: Max | Specify a value between 0 and 128MB or leave blank. You can optionally specify the suffix K, KB, M, or MB. If you omit the suffix, the specified value in bytes is used. For example:<br><br>■ 128K means 128 kilobytes.<br><br>■ 128MB means 128 megabytes.<br><br>■ 128 means 128 bytes.<br><br>The VSAM maximum hiperspace sets the maximum amount of hiperspace storage that IOO allocates per file. |

**6** After you have completed the panel, press Enter. The IOO-DCF Global Table Overrides (DCF008) panel (part 2 of 2) is displayed:

```
DCF008 V315  ---------  IOO-DCF Global Table Overrides  -----------------------
Command/Option ===>                                          Use PF3/END to exit


Enter your IOO global table overrides (part 2 of 2) :

   (Leave a field blank to not override the associated data element.)

 - Extended region size   ===>             .
 - Trace message limit    ===>         .

 - Allow started task selection ? ===>    ( Y or N )

 - SMF record number ===>      . (128-255)
 - IOO events to be recorded :
     - Non-VSAM re-blocking  ===>   . (Enter S to select)
     - Non-VSAM re-buffering ===>   . (Enter S to select)
     -     VSAM re-buffering ===>   . (Enter S to select)

 - IOO DDnames :          For entire step      At DDname Level
     - IOO deactivation : ===>         .        ===>         .
     - IOO   activation : ===>         .        ===>         .
```

**7** Fill in the fields on the panel as follows:

| Field | Description |
|---|---|
| Extended region size | The extended region size is the region for VSAM above-the-line storage. Specify a value between 0 and 512MB or leave blank. You can specify the K, KB, M, or MB suffix. If you omit the suffix, the specified value in bytes is used. For example:<br><br>■ 128K means 128 kilobytes.<br><br>■ 128MB means 128 megabytes.<br><br>■ 128 means 128 bytes. |
| Trace message limit | The trace message limit specifies the maximum number of trace entries produced for an optimization. Once reached, no more trace entries are printed or collected anywhere. You can specify a value between 0 and 9999 or leave blank. |
| Allow started task selection? | This option sets the global default used by IOO:<br><br>■ Y: Indicates that, by default, IOO is allowed to optimize started tasks.<br><br>■ N: Indicates that, by default, IOO is not allowed to optimize started tasks.<br><br>■ Leave blank to use the value in the active IOO global table.<br><br>Note that an STC option coded as an optimization rule override will override this default value. |
| SMF record number | Specify a value of 128-255. Specify 0 to disable IOO SMF user records.<br><br>The number specified will be used as the SMF record number for IOO SMF user records. IOO writes an SMF user record for VSAM rebuffering, QSAM rebuffering, and/or reblocking for selected files.<br><br>Leave blank to use the value in the active IOO global table. |
| IOO Events To Be Recorded | |
| Non-VSAM Reblocking | ■ Specify **S** to indicate that IOO will record non-VSAM reblocking events.<br><br>■ Leave blank to indicate that IOO will not record non-VSAM reblocking events.<br><br>If this option is selected, a valid SMF record number must also be selected. |

| Field | Description |
|---|---|
| Non-VSAM Rebuffering | ■ Specify **S** to indicate that IOO will record non-VSAM rebuffering events.<br><br>■ Leave blank to indicate that IOO will not record non-VSAM rebuffering events.<br><br>If this option is selected, a valid SMF record number must also be selected. |
| VSAM Rebuffering | ■ Specify **S** to indicate that IOO will record VSAM rebuffering events.<br><br>■ Leave blank to indicate that IOO will not record VSAM rebuffering events.<br><br>If this option is selected, a valid SMF record number must also be selected. |
| **IOO DDnames** | |
| IOO Deactivation: For Entire Step | Enter the DDname whose presence will cause IOO to be deactivated for the entire step. |
| IOO Deactivation: At DDname Level | Enter the DDname whose presence will cause IOO to be deactivated for any subsequent DD statements. |
| IOO Activation: For Entire Step | Enter the DDname whose presence will cause IOO to be activated for the entire step. |
| IOO Activation: At DDname Level | Enter the DDname whose presence will cause IOO to be activated for any subsequent DD statements. |

**8**    If you leave the Command/Option field blank on the IOO-DCF Rule Selection (DCF005) panel, you can process the line items on the panel one at a time by entering one of the following values in the Sel input field:

- **S**: Select the rule.

- **D**: Delete an existing rule.

- **I**: Insert (define) a new rule.

- **R**: Replicate an existing rule.

**9**    If you enter the **S** or **I** selection code, the following panels are display in sequence so that you can fully define the DCF rule:

- IOO-DCF Rule Definition panel (DCF002).

- IOO-DCF Rule Table Overrides (DCF003).

- IOO-DCF Advanced Rule Overrides (DCF004).

- IOO-DCF Rules Comments (DCF006).

See "Command/Option 1: Insert a Rule into the Start of an IOO-DCF Table" on page 29 for a description of the fields on these panels.

Fill in the fields in each panel as appropriate and press Enter to continue with each panel in succession. When you press Enter on panel DCF006, IOO will bring you back to the display of panel DCF005 (the list of rules found in the selected DCF input data set).

When all needed updates (either to global values or to individual rules) are complete, press PF3. IOO will present pop-up panel DCFREP to prompt you for confirmation that you want to update the indicated output data set.

Press Enter if the data set name is correct. If you enter **EXIT** or **CANCEL**, any updates will be discarded.

Take the following actions to activate the newly updated rules:

**1**    Upon return to the IOO-DCF Master Control Panel (panel DCF001), select option 3, Activate an IOO-DCF table and press Enter.

**2**    On the next panel (panel DCF009):

- Enter (or verify) the name of the data set that contains the updated DCF rules.

- Press Enter to request that the DCF rules be reloaded in the current IOO environment.

See "Command/Option 3: Activate an IOO-DCF Table" on page 46 for a description of this command/option.

# Command/Option 3: Activate an IOO-DCF Table

Take the following actions to activate the newly updated rules:

**1** Upon return to panel DCF001, select option 3 (Activate an IOO-DCF table) and press Enter. The IOO-DCF Restart Input DSN Definition Panel (DCF009) is displayed:

```
DCF009 V315  ----  IOO-DCF Restart Input DSN Definition Panel  --------------
Command/Option ===>                                        Use PF3/END to exit

Press Enter to acivate IOO-DCF input data set

 DCF DSN ===> 'USER1.STRIOO.V3R1M5.CUSTOM.JCL(DCFK3)'        .
 IOO SSN ===> U315 .
```

**2** Fill in this panel as follows:

| Field | Description |
|-------|-------------|
| DCF DSN | Enter (or verify) the name of the data set containing the DCF input that you want to activate. |
| IOO SSN | Enter the name of the target IOO subsystem. The default value is U315. This value is set in the IOO and IOOSTART JCL procedures that are distributed with StarTool IOO 3.1.4. Users can change this value; however, changing the value is not recommended. |

**3** Press Enter to request that the modified DCF rules be reloaded in the current IOO environment. The following message is displayed:

```
SZIB004I IOO DCF activation command submitted.
***
```

**4** Press Enter to continue. The IOO-DCF Master Control Panel (DCF001) is displayed.

# DCF Input

DCF input is supplied to IOO at two different times:

- Through DDname ULTIDCF during IOO start-up time. IOO will parse the DCF input stream and load its contents into memory.

- During DCF maintenance processing when an ISPF user updates a DCF input stream. The DCF input stream is identified to IOO through the data set name that the ISPF user enters on panel DCF001.

The DCF input stream consists of:

- DCF rule definition records (see "DCF Rule Definition Records" on page 47).

- IOO global table override records (see "DCF IOO Global Table Override Records" on page 48).

## DCF Rule Definition Records

You use the ISPF interface to DCF to build and maintain DCF rule definitions. DCF rule definitions consist of one or more sets of records. Each set of records defines a DCF rule by means of the following record types:

- Up to 10 lines of comments that describe the rule being defined (optional).

- One or more Scope records that define the IOO optimization components to be selected or rejected by the rule (required).

- One or more Criteria records that define elements that must be matched by an application's access to a data set for the DCF rule to apply (required).

- One or more override records that identify data elements to be overridden in the IOO optimization rule that IOO selects for this data set (required).

The records are maintained by the ISPF interface to DCF in the same order listed above (for example, the Scope records precede the Criteria records and so on).

The DCF input stream will consist of multiple sets of these records, each defining a different DCF rule.

## DCF IOO Global Table Override Records

The ISPF interface to DCF can also, optionally, maintain overrides for IOO global options. While these override records can be placed anywhere within the DCF input stream (even within the middle of a DCF rule definition), they will be moved to the end of the DCF input stream when the ISPF interface to DCF encounters them.

# General Rule Coding Guidelines

**NOTE**  We recommend that you use the ISPF interface to DCF to maintain DCF data. You should use the native ISPF edit function only in special circumstances. The DCF rule definition coding rules and formats provided in this manual describe DCF output, which you can use as a guide if you need to maintain DCF data with the ISPF edit function.

General DCF coding rules are as follows:

- DCF rule definition records can start anywhere within record positions 1-66.

- Embedded blanks within coded values are not supported. IOO ignores any record that contains a blank in record positions 1-66 and the ISPF interface to DCF deletes the record.

- Both uppercase and lowercase values are permitted.

- Only one operand can be coded per input record.

# Comment Records

A Comment record begins with an asterisk character (*) in the first byte of the record. Up to 10 comment records can be included for a DCF rule definition.

# Scope Records

Scope records identify the components that are to be either selected or bypassed by the DCF rule. Scope records must be coded as follows:

| Scope Record | Description |
|---|---|
| SELECT=(*IOO_component_list*) | Identifies the IOO components that will be selected if the DCF rule is matched. |
| BYPASS=(*IOO_component_list*) | Identifies the IOO components that will be bypassed if the DCF rule is matched. |

See "Rules for Coding the Component List" on page 49 for rules on coding the component list.

## Rules for Coding the Component List

The IOO component list identifies the IOO components whose selection or rejection is being defined by the DCF rule. Rules for coding the component list are:

- If more that one IOO component is to be identified, enclose the list of components in parentheses and separate the components by commas.

- If a single IOO component is to be identified, the use of parentheses is optional.

- Multiple Scope records can be coded for the same rule.

- The coding of ALL for a SELECT statement will remove any previously identified BYPASS requests. Conversely, the coding of ALL for a BYPASS statement will remove any previously identified SELECT requests.

The following component list values are supported:

| Value | Description |
|---|---|
| ALL | The rule applies to all IOO components. |
| IOO-Block | The rule applies to the IOO non-VSAM reblocking and rebuffering component. |
| IOO-LSR | The rule applies to the IOO LSR component. |
| IOO-VSAM | The rule applies to the IOO VSAM component. |

The following values are considered obsolete but will be supported as indicated during IOO startup. If they are encountered by the ISPF interface to DCF, they will be upgraded to the newer values.

| Value | Description |
|---|---|
| IOO-SAM | IOO startup treats this value as IOO-Block. |
| Ulti-Block | IOO startup treats this value as IOO-Block. |
| Ulti-SAM | IOO startup treats this value as IOO-Block. |
| Ulti-LSR | IOO startup treats this value as IOO-LSR. |
| Ulti-VSAM | IOO startup treats this value as IOO-VSAM. |

The following values are no longer supported. If IOO startup encounters them, they will be ignored. If the ISPF interface to DCF encounters them, they will be deleted from the DCF input stream.

| Value | Description |
|---|---|
| IOO-IMS | Not supported. |

## Examples

| Example | Meaning |
|---|---|
| SELECT=(ALL) | All IOO components are to be selected by the rule. |
| BYPASS=ALL | All IOO components are to be bypassed by the rule. |
| SELECT=ALL<br>BYPASS=IOO-VSAM | Select all IOO components except IOO-VSAM. |
| SELECT=IOO-VSAM<br>SELECT=(IOO-LSR) | Select IOO-VSAM and IOO-LSR components. |
| SELECT=ALL<br>BYPASS=(IOO-LSR) | Select all IOO components except IOO-LSR. |
| SELECT=ALL | Select all IOO components (including LSR).<br><br>**NOTE** Coding SELECT=ALL resets any pending BYPASS requests. |

# Criteria Records

Criteria records identify the elements that must be matched for IOO to apply the rule to the data set being accessed. The following Criteria records are supported. Code the desired rule name and include optimization rule overrides for any values you wish to change:

| Criteria Record | Description |
|---|---|
| JOB= | Identifies a job name to be used as match criteria for the DCF rule. |
| STEP= | Identifies a step name to be used as match criteria for the DCF rule. |
| JSTEP= | Identifies a job-step name that invokes a JCL procedure to be used as match criteria for the DCF rule. |
| PGM= | Identifies a program name to be used as match criteria for the DCF rule. |
| DD= | Identifies a DDname to be used as match criteria for the DCF rule. |
| DSN= | Identifies a DSN (data set name) to be used as match criteria for the DCF rule. |

| Criteria Record | Description | | |
|---|---|---|---|
| RULE= | Identifies the name of the optimization rule that is to be used (should the remaining Criteria records cause selection of the DCF rule). If no RULE= Criteria record is provided, IOO will search for the appropriate optimization rule. IOO provides the following optimization rule names for use with this operand: | | |
| | IOOQSAMN | Forces the use of a default of 128 buffers for QSAM files. Reblocking is not allowed. The default buffer count can be modified with a BUFNO DCF override record. | |
| | IOOQSAMY | Forces the use of a default of 128 buffers for QSAM files. Reblocking is allowed. The default buffer count can be modified with a BUFNO DCF override record. A specific blocksize can be requested with a BLKSIZE DCF override record. | |
| | VSMLSR24 | Forces the use of RMODE24 LSR buffers for VSAM files. The default maximum bufferspace is 152KB, but can be modified with the following DCF override records: | |
| | | DSIZEMX | Specifies the maximum bufferspace allocated to the data component. |
| | | ISIZEMX | Specifies the maximum bufferspace allocated to the index component. |
| | VSMLSR31 | Forces the use of RMODE31 LSR buffers for VSAM files. The default maximum bufferspace varies based upon the selected IOO start-up options: | |
| | | K0 or K3 | Maximum of 2MB. |
| | | K1 or K4 | Maximum of 2MB. |
| | | K2 or K5 | Maximum of 32MB. |
| | | The following DCF override records can modify the default maximum bufferspace: | |
| | | DSIZEMX | Specifies the maximum bufferspace allocated to the data component. |
| | | ISIZEMX | Specifies the maximum bufferspace allocated to the index component. |
| | | This rule allows non-COBOL programs to use RMODE31 buffers. A FLAGS DCF override record that specifies NO_ALCBUF31 can change this rule. | |
| | VSMNSR24 | Forces the use of RMODE24 NSR buffers for VSAM files. The default maximum bufferspace is 256KB, but can be modified with a BUFSP DCF override record. | |
| | VSMNSR31 | Forces the use of RMODE31 NSR buffers for VSAM files. The default maximum bufferspace is 3MB, but can be modified with a BUFSP DCF override record. | |

Coding notes:

■ All Criteria records except RULE support the use of the asterisk (*) as a variable-length wildcard character.

- All Criteria records except RULE support the use of the questions mark (?) as a fixed-length wildcard character.

- All coded Criteria records must be matched for the DCF rule to be selected.

- Only one of each type of Criteria record should be coded for each DCF rule. While this rule is now enforced by the ISPF interface to DCF, the similar-criteria feature of DCF still allows the coding of multiple criteria of the same type within a single DCF rule group when the DCF input is processed during IOO startup.

- The similar-criteria feature of DCF states that if IOO encounters two identical criteria types within the domain of the same Scope record, IOO will replicate the current rule and apply the new criteria to the new rule.

For example, the following single rule:

```
SELECT=ALL
    JOB=JOBNAME1
    DD=DDNAME1
    JOB=JOBNAME2
    DD=DDNAME2
```

will be resolved into the following three rules:

```
SELECT=ALL
    JOB=JOBNAME1
    DD=DDNAME1

SELECT=ALL
    JOB=JOBNAME2
    DD=DDNAME1

SELECT=ALL
    JOB=JOBNAME2
    DD=DDNAME2
```

IOO startup will resolve and honor any DCF input that is coded to use the similar-criteria feature of DCF, and the ISPF interface to DCF will convert that input into discreet rules. This behavior allows for the easy conversion of existing DCF data.

## Examples of Wildcard Use

With the fixed-length ? wildcard character, one and only one character (of any value) must be present. In the following example:

```
JOB=MYJOB?
```

- MYJOB1 would match. The ? is replaced with 1.

- MYJOB2 would match. The ? is replaced with 2.

- MYJOB would fail due to no replacement character.

- MYJOB21 would fail due to the presence of too many replacement characters.

With the variable-length * wildcard character any number of characters (including no characters) can be present. In the following example:

```
DD=SYS*1
```

- SYSUT1 would match. The * is replaced with UT.

- SYSUT21 would match. The * is replaced with UT2.

- SYS1 would match. The * is replaced with a null.

- SYS11 would match. The * is replaced with 1.

- SYSUT21A would fail because of the A suffix.

- SYSUT2 would fail because of no suffix of 1.

## Multiple Rule Example

Given the following three sets of DCF input:

- Select all IOO components for DDname SYSUT1 when it is processed by job MYJOB1:

```
SELECT=(ALL)
   JOB=MYJOB1
   DD=SYSUT1
```

and

- Select IOO-VSAM optimization for step STEP1 of job TESTV1. Force the use of IOO optimization rule VSMLSR31:

```
SELECT=IOO-VSAM
   JOB=TESTV1
   STEP=STEP1
   RULE=VSMLSR31
```

and

- Bypass all other jobs:

```
BYPASS=ALL
   JOB=ALL
```

The following would occur:

- Any data set accessed via DDname SYSUT1 by a job named MYJOB1 would be selected for optimization by the first DCF rule and would search the optimization rules table for the needed optimization parameters.

- Any data set accessed via DDname SYSUT2 by the same job would match the third DCF rule and would not be selected for optimization.

- Any data set accessed within step STEP1 of a job named TESTV1 would match the second DCF rule and would be selected for optimization. Rule VSMLSR31 (within the optimization rules table) would be used for the optimization parameters.

- Any data set accessed by a job named PROD1 would match the third DCF rule and would not be selected for optimization by IOO.

# Optimization Rule Override Records

Override records identify parameters in the selected optimization rule that are to be overridden when the current DCF rule is matched. There are three categories of override records:

- Non-VSAM override records. (See "Non-VSAM Override Records" on page 54 for a description of each non-VSAM override record.)

- VSAM override records. (See "VSAM Override Records" on page 55 for a description of each VSAM override record.)

- Common override records (See "Common Override Records" on page 61 for a description of each common override record.). Common override records apply to both VSAM and non-VSAM processing.

## Non-VSAM Override Records

The non-VSAM override records are:

- BLKSIZE

- BUFNO

- FLAGS

- LBI_BLKSIZE

These override records are described in the following sections.

### BLKSIZE=value{K|KB}

The BLKSIZE override record identifies the target blocksize that IOO will use when optimizing a data set's blocksize. Coding rules are:

- The value cannot exceed 32767.

- A value of zero will bypass the use of the optimization rule table blocksize. The blocksize will instead be extracted from the IOO device table.

- Suffixes of K and KB are supported.

Examples:

| Example | Meaning |
|---|---|
| `BLKSIZE=8000` | Request a blocksize of 8000. |
| `BLKSIZE=8K` | Request a blocksize of 8192. |

### BUFNO=*value*

The BUFNO override record identifies the target value that IOO is to use when optimizing a data set's buffer count. Coding rules are:

- The value must be between 0 and 255.

- A value of 0 will cause the buffer count to not be optimized.

- If BUFNO and LBI_BLKSIZE are both specified:

  • The BUFNO value will be applied to Large Block Interface (LBI) optimizations only (blocksizes of 32KB and above).

  • Non-LBI optimizations (blocksizes less than 32KB) will use the buffer count extracted from the optimization rule.

- If BUFNO is specified without LBI_BLKSIZE:

- The BUFNO value will be applied to non-LBI optimizations only (blocksizes of 32KB and above).

- Since LBI_BLKSIZE was not coded, no LBI optimizations will be performed for the rule.

Examples:

| Example | Meaning |
| --- | --- |
| BUFNO=32 | Request 32 buffers (non-LBI). |
| BUFNO=24<br>BLKSIZE=4096 | Request 24 buffers (non-LBI). |
| BUFNO=16<br>LBI_BLKSIZE=46K | Request 16 buffers (LBI). |

### FLAGS=(*options*)

The FLAGS override record identifies additional IOO optimization options. The supported value for non-VSAM, LBI optimizations only is:

    FLAGS=(ALCBUF31)

This record directs IOO to request that the buffers can be allocated from RMODE31 storage.

### LBI_BLKSIZE=*value*{K|KB}

The LIB_BLKSIZE override record defines the target blocksize that IOO will use when reblocking a data set that can use the Large Block Interface (LBI). Coding rules are:

- The value must be between 0 and 256KB.

- A value of 0 will cause the LBI blocksize not to be optimized.

- Suffixes of K and KB are supported, but optional.

Examples:

| Example | Meaning |
| --- | --- |
| LBI_BLKSIZE=40000 | Request a blocksize of 40,000. |
| LBI_BLKSIZE=128KB | Request a blocksize of 131,072. |

## VSAM Override Records

The VSAM override records are:

- BIAS

- BUFND

- BIFNI

- BUFSP

- DMSHWM

- DSIZEMX

- FLAGS

- ISIZEMX

- VSI

These override records are described in the following sections.

### BIAS={DIR|DYN|SEQ}

The BIAS override record identifies the bias to be used when NSR buffers are allocated. The BIAS record can be used when an application defines a file's access one way, but actually accesses it using a different mode. The following three values are supported:

| Value | Meaning |
|---|---|
| BIAS=DIR | IOO will bias the buffer allocation in favor of direct (random) access by the application. IOO will attempt to allocate enough index buffers to keep as many non-sequence-set records as possible in storage. IOO will ensure there are enough data buffers to satisfy the declared string count. |
| BIAS=DYN | IOO will bias the buffer allocation in favor of dynamic (random and sequential) access by the application. IOO will attempt to allocate enough index buffers to keep as many non-sequence-set records as possible in storage. IOO will attempt to allocate up to one Control Area (CA) of data buffers. |
| BIAS=SEQ | IOO will bias the buffer allocation in favor of sequential access by the application. IOO will ensure there are enough index buffers to satisfy the declared string count. IOO will attempt to allocate up to one Control Area (CA) of data buffers. |

Example:

| Example | Meaning |
|---|---|
| BIAS=SEQ | Request buffers be allocated with a bias toward sequential processing. |

### BUFND=value{K|KB}

The BUFND override record identifies the target data component buffer count for NSR optimizations. Coding rules are:

- The value must be between 0 and 32767.

- A value of zero will cause the override to be ignored. The value in the selected optimization rule will be used.

- Suffixes of K and KB are supported, but optional.

Example:

| Example | Meaning |
|---|---|
| BUFND=512 | Request 512 data buffers. |

### BUFNI=*value***{K|KB}**

The BUFNI override record identifies the target index component buffer count for NSR optimizations. Coding rules are:

- The value must be between 0 and 32767.

- A value of zero will cause the override to be ignored. The value in the selected optimization rule will be used.

- Suffixes of K and KB are supported, but optional.

Example:

| Example | Meaning |
|---------|---------|
| BUFNI=128 | Request 128 data buffers. |

### BUFSP=*value***{K|KB|M|MB}**

The BUFSP override record identifies the target bufferspace to be used for NSR optimizations. Coding rules are:

- The value must be between 0 and 134,217,728 (128MB).

- A value of zero will cause the override to be ignored. The value in the selected optimization rule will be used.

- Suffixes of K, KB, M, and MB are supported, but optional. If you omit the suffix, the specified value in bytes is used. For example:

  - 128K means 128 kilobytes.

  - 128MB means 128 megabytes.

  - 128 means 128 bytes.

Example:

| Example | Meaning |
|---------|---------|
| BUFNI=2MB | Request 2MB of NSR buffers. |

### DMSHWM=(*count,percent*)

The DMSHWM override record identifies the threshold (high-water-mark) values for the Dynamic Mode Selection (DMS) feature. For an LSR file, if the number of consecutive sequential requests exceeds the threshold value determined by IOO, IOO will close the data set and reopen it using NSR optimization.

- The *count* value refers to an absolute number of consecutive sequential requests.

- The *percent* value refers to the percentage of allocated buffers.

- The threshold value used by IOO depends upon the global DMS option as follows:

  - If the global DMS options is set to MIN (the default value), the minimum of the two values will be used as the threshold.

  - If the global DMS options is set to MAX, the maximum of the two values will be used as the threshold.

For example, assume that DMSHWM=(12,10) is coded in the selected DCF rule and that a data set selected for optimization by that rule is given 1024 data buffers (10 percent of which is a value of 102).

If DMS=MIN was defaulted in the global table, IOO would use a value of 12 (the minimum of the absolute count of 12 and the computed percent of 102) as the DMS threshold. Thus, when the application program issued its 13th consecutive sequential request, DMS would attempt to close the data set and reopen it using NSR optimization.

If DMS=MAX was coded to in the global table, IOO would use a value of 102 (the maximum of the absolute count of 12 and the computed percent of 102) as the DMS threshold. Thus, when the application program issued its 103rd consecutive sequential request, DMS would attempt to close the data set and reopen it using NSR optimization.

Coding rules are:

■ The *count* value must be between 0 and 255 (inclusive).

■ The *percent* value must be between 0 and 99 (inclusive).

If DMS=(0,0) is coded, DMS will effectively be turned off for this access of the data set.

### DSIZEMX=*value*{K|KB|M|MB}

The DSIZEMX override record identifies the maximum bufferspace that IOO will allocate for the data component during an LSR optimization. Coding rules are:

■ The value must be between 0 and 134,217,728 (128MB).

■ A value of zero will cause the override to be ignored. The value in the selected optimization rule will be used.

■ Suffixes of K, KB, M, and MB are supported, but optional. If you omit the suffix, the specified value in bytes is used. For example:

  ■ 128K means 128 kilobytes.

  ■ 128MB means 128 megabytes.

  ■ 128 means 128 bytes.

Example:

| Example | Meaning |
| --- | --- |
| DSIZEMX=64MB | Use a maximum of 64MB of data buffers for an LSR optimization selected by this rule. |

### FLAGS=(*option1*,*option2*,...)

The FLAGS override record identifies additional IOO optimization options. Supported options for VSAM optimizations are:

| Option | Meaning |
|---|---|
| FLAGS=(ACCNVRPL) | Directs IOO to allow Request Parameter Lists (RPLs) not specifically marked as being VSAM RPLs to be processed by the VSAM Record Level Interface. By default, the VSAM Record Level Interface will not process an RPL that is not marked as being a VSAM RPL. |
| FLAGS=(ALCBUF31) | Directs IOO to allow non-COBOL programs to use RMODE31 buffers. |
| FLAGS=(EXCEEDCA) | Directs IOO to allow a component's bufferspace to exceed the size of a Control Area (CA). By default, IOO will use the CA size as one of the maximums for a component's bufferspace. |
| FLAGS=(IGNHURBA) | Directs IOO to ignore the highest-used-RBA value (extracted from the catalog) when attempting to determine the maximum size of a component. |
| FLAGS=(REUSESHR) | Directs IOO to reuse an application's original LSR shrpool number when all other shrpool numbers have been used. |
| FLAGS=(USEVRPLM) | Directs IOO to use the buffer counts in an application BLDVRP macro as the minimums for its optimization. |
| The following options direct IOO to negate the requested option (if the option is present within the selected optimization rule). If coded with the requested option, this value takes precedence: | |
| FLAGS=(NO_ACCNVRPL) | Negates the ACCNVRPL option. |
| FLAGS=(NO_ALCBUF31) | Negates the ALCBUF31 option. |
| FLAGS=(NO_EXCEEDCA) | Negates the EXCEEDCA option. |
| FLAGS=(NO_IGNHURBA) | Negates the IGNHURBA option. |
| FLAGS=(NO_REUSESHR) | Negates the REUSESHR option. |
| FLAGS=(NO_USEVRPLM) | Negates the USEVRPLM option. |

For example, if the following were coded, the NO_ACCNVRPL value would supercede the ACCNVRPL value:

```
FLAGS=(ACCNVRPL,ALCBUF31,NO_ACCNVRPL)
```

### ISIZEMX=*value*{K|KB|M|MB}

The ISIZEMX override record identifies the maximum bufferspace that IOO will allocate for the index component during an LSR optimization. Coding rules are:

- The value must be between 0 and 134,217,728 (128MB).

- A value of zero will cause the override to be ignored. The value in the selected optimization rule will be used.

- Suffixes of K, KB, M, and MB are supported, but optional. If you omit the suffix, the specified value in bytes is used. For example:

- 128K means 128 kilobytes.

- 128MB means 128 megabytes.

- 128 means 128 bytes.

Example:

| Example | Meaning |
|---------|---------|
| `ISIZEMX=16MB` | Use a maximum of 16MB of index buffers for an LSR optimization selected by this rule. |

### VSI=(*option1,option2,...*)

The VSI override record identifies the options to be set for the VSAM Record Level Interface (RLI). The following options are supported:

| Option | Meaning |
|--------|---------|
| `VSI=OFF` | Deactivate the interface for this optimization. |
| `VSI=ON` | Activate the interface for this optimization. |
| `VSI=NODBM` | Do not the activate the Dynamic Buffer Management (DBM) feature for this optimization. The DBM feature will determine whether or not the issuer of an I/O request is a COBOL program or not. If it is a COBOL program, DBM will allow the use of RMODE31 buffers. If it is not a COBOL program, DMS will not allow the use of RMODE31 buffers unless the optimization was performed using FLAGS=ALCBUF31. Coding NODBM will cause this check to be bypassed by the IOO VSAM Record Level Interface. |
| `VSI=NODBT` | Do not activate the Dynamic Buffer Translation (DBT) feature for this optimization. The DBT feature will cause IOO to provide buffer translation services for GET/GETX requests using locate mode. |
| `VSI=NODLP` | Do not activate the Dynamic LSR Positioning (DLP) feature for this optimization. The DLP feature will cause IOO to position to the first record of a data set when opened using LSR optimization. |
| `VSI=NODMS` | Do not activate the Dynamic Mode Selection (DMS) feature for this optimization. The DMS feature will cause IOO to switch from LSR optimization to NSR optimization when the number of consecutive sequential requests exceeds the defined threshold. |

Coding rules are:

- Any combination of the supported values is allowed.

- If both OFF and ON are coded, the last one coded is the one used.

- If OFF is coded, all VSI features are deactivated for the data set access.

Examples:

| Example | Meaning |
|---------|---------|
| `VSI=(ON,NO_DMS)` | Activate the VSAM Record Level Interface, but do not allow DMS support to be used for this access of the data set. |
| `VSI=(OFF,NO_DMS)` | Deactivate the VSAM Record Level Interface. Because OFF was coded, the NO_DMS coding is not needed, but is allowed. |
| `VSI=(ON,OFF)` | Deactivate the VSAM Record Level Interface. When ON and OFF are both coded, the last one coded is used. |

# Common Override Records

Common override records apply to both VSAM and non-VSAM processing. Common override records are:

- STC

- TRACE

These records are described in the following sections.

### STC={YES|NO}

The STC override record identifies whether or not started tasks matching the DCF rule can be selected for optimization. This value overrides any STC value coded either in the global table or in the DCF input stream as a global table override. The shipped global table default is STC=YES. The following values are supported:

| Value | Meaning |
|-------|---------|
| STC=NO | Started tasks are not allowed to be selected by this DCF rule. When a started task matches the criteria for this DCF rule, IOO will continue its processing with the next rule. All other tasks matching the criteria for this DCF rule will be allowed to match the rule. Coding STC=NO within a DCF rule differs from the coding of a global value of STC=NO as follows: <ul><li>A global value of STC=NO will cause IOO to bypass optimization of ALL started tasks.</li><li>A value of STC=NO on a DCF rule will cause IOO to bypass optimization of only those started tasks that match the criteria of the DCF rule.</li></ul> |
| STC=YES | Started tasks are allowed to be selected by this DCF rule. Coding STC=YES for a DCF rule is ignored if a global value of STC=YES is coded. |

Example:

| Example | Meaning |
|---------|---------|
| STC=YES | Allow started tasks to be selected by this DCF rule even though STC=NO may be set globally. |

**TRACE=(*value,...,value*)**

The TRACE override record identifies the trace options to be set when the DCF rule is matched. The following values are supported:

| Value | Meaning |
|-------|---------|
| CSR | Select normal trace for the IOO CSR component. |
| DCF | Select normal trace for the IOO DCF component. |
| DCFD | Select detail trace for the IOO DCF component. |
| DEX | Select normal trace for the IOO DEX component. |
| DMM | Select normal trace for the IOO DMM component. |
| JFC | Select normal trace for the IOO JFC component. |
| JFCD | Select detail trace for the IOO JFC component. |
| OPN | Select normal trace for the IOO OPN component. |
| OPND | Select detail trace for the IOO OPN component. |
| VSI | Select normal trace for the IOO VSI component. |
| VSM | Select normal trace for the IOO VSM component. |
| VSMD | Select detail trace for the IOO VSM component. |

You should request TRACE only when asked to do so by Customer Support. Requesting trace incorrectly can add significant overhead to IOO.

# Global Table Override Records

Global table override records are identified by the presence of the characters GBT (either in uppercase or lowercase) as the first value of the input DCF record. The characters GBT must be followed by at least one space. A supported global table override operand must then be present.

Supported global table override operands are:

- BLKMSG
- BUFMSG
- DD_OFF
- DD_ON
- DD_SOFF
- DD_SON
- EREGION
- SETBLK
- SETBUF
- SETRGN
- SMF

- STC
- TRCLIM
- TSOMSG
- VMAXHSPC
- VMAXSPC
- VMINHSPC
- VMINSPC

The following sections describe these records.

### GBT BLKMSG={WTO|WTL|JESYSMSG|NO}

The BLKMSG override record identifies the message processing option when either of the following is true:

- IOO reblocks a non-VSAM data set (reblocking is selected).
- IOO identifies a non-VSAM data set that could have been reblocked, but reblocking was not allowed.

The following values are supported:

| Value | Meaning |
|---|---|
| GBT BLKMSG=NO | IOO will not issue any reblocking messages. |
| GBT BLKMSG=WTL | IOO will use the IBM WTL macro to issue reblocking messages. |
| GBT BLKMSG=WTO | IOO will use the IBM WTO macro to issue reblocking messages. |
| GBT BLKMSG=JESYSMSG | IOO will direct all re-blocking messages to the JESYSMSG DD statement. This is the default value in the shipped global table. |

### GBT BUFMSG={WTO|WTL|JESYSMSG|NO}

The BUFMSG override record identifies the message processing option when IOO rebuffers a data set (whether VSAM or non-VSAM). The following values are supported:

| Value | Meaning |
|---|---|
| GBT BUFMSG=NO | IOO will not issue any rebuffering messages. |
| GBT BUFMSG=WTL | IOO will use the IBM WTL macro to issue rebuffering messages. |
| GBT BUFMSG=WTO | IOO will use the IBM WTO macro to issue rebuffering messages. |
| GBT BUFMSG=JESYSMSG | IOO will direct all rebuffering messages to the JESYSMSG DD statement. This is the default value in the shipped global table. |

### GBT {DD_OFF|DD_ON}=ddname

The GBT DD_OFF override record identifies the DDname whose presence in a step will cause IOO processing to be bypassed until either the end of the step or until a new IOO DDname domain is encountered.

The GBT DD_ON override record identifies the DDname whose presence in a step will cause IOO processing to be selected for any subsequent DD statements until either the end of the step or until a new IOO DDname domain is encountered.

Any valid DDname may be coded.

The shipped defaults in the IOO global table are:

- DD_OFF=ULT@OFF

- DD_ON =ULT@ON

If the current DDname falls within the domain of either of these IOO DDnames, the DCF rule table will NOT be searched.

Examples:

The following DCF input is used in the examples. (Note that the DD_ON value is allowed to default):

```
SELECT=ALL
   STEP=STEP1
BYPASS=ALL
   STEP=STEP2
GBT DD_OFF=IOO$OFF
```

| Example | Meaning |
|---|---|
| ```//STEP1    EXEC PGM=PROG1``` <br> ```//SYSIN   DD   DISP=SHR,DSN=MY.CONTROL.FILE``` <br> ```//SYSUT2  DD   DISP=SHR,DSN=MY.OUTPUT.FILE``` <br> ```//IOO$OFF DD   DUMMY``` <br> ```//SYSUT1  DD   DISP=SHR,DSN=MY.INPUT.FILE``` | IOO processing would be bypassed for DDname SYSUT1 because it falls within the domain of the DD_OFF DDname of IOO$OFF. <br><br> The domain of the IOO$OFF DDname would extend to the end of the step since there is no ULT@ON DDname subsequent to it in the JCL. IOO would search the DCF rules table when DDnames SYSIN and SYSUT2 are accessed since those DDnames fall outside the domain of any IOO DDname. |
| ```//STEP2    EXEC PGM=PROG1``` <br> ```//SYSIN   DD   DISP=SHR,DSN=MY.CONTROL.FILE``` <br> ```//ULT@ON  DD   DUMMY``` <br> ```//SYSUT2  DD   DISP=SHR,DSN=MY.OUTPUT.FILE``` <br> ```//IOO$OFF DD   DUMMY``` <br> ```//SYSUT1  DD   DISP=SHR,DSN=MY.INPUT.FILE``` | IOO processing would be selected for DDname SYSUT2 because if falls within the domain of the default DD_ON of ULT@ON. <br><br> The domain of the ULT@ON DDname is terminated after the SYSUT2 DDname due to the presence of the IOO$OFF DDname. IOO processing would be bypassed for DDname SYSUT1 because it falls within the domain of the DD_OFF DDname of IOO$OFF. <br><br> The domain of the IOO$OFF DDname would extend to the end of the step because there is no ULT@ON DDname subsequent to it in the JCL. <br><br> IOO would search the DCF rules table for DDname SYSIN because it is outside the domain of an IOO DDname. Due to the DCF rules used in the example, DDname SYSIN would be bypassed for processing by IOO. |

## GBT {DD_SOFF|DD_SON}=ddname

The GBT DD_SOFF override record identifies the DDname to be bypassed for IOO processing for the entire step.

The GBT DD_SON override record identifies the DDname to be selected for IOO processing for the entire step.

If either of these DDnames is present in the step's JCL, the DCF rules table will not be searched.

You can code any valid DDname. The shipped defaults in the IOO global table are:

- DD_SOFF=ULT@SOFF

- DD_SON=ULT@SON

Examples:

The following DCF input is used in the examples. (Note that the DD_SON value is allowed to default):

```
SELECT=ALL
    STEP=STEP1
BYPASS=ALL
    STEP=STEP2
GBT DD_SOFF=IOO$SOFF
```

| Example | Meaning |
|---|---|
| ```//STEP1    EXEC PGM=PROG1```<br>```//SYSIN    DD   DISP=SHR,DSN=MY.CONTROL.FILE```<br>```//SYSUT2   DD   DISP=SHR,DSN=MY.OUTPUT.FILE```<br>```//IOO$SOFF DD   DUMMY```<br>```//SYSUT1   DD   DISP=SHR,DSN=MY.INPUT.FILE``` | All three DD statements (SYSIN, SYSUT2, and SYSUT1) will be bypassed from IOO processing due to the presence of DDname IOO$OFF in the step JCL.<br><br>If DDname IOO$OFF had not been present in the step JCL, the files would have been selected for IOO processing by the first DCF rule (the SELECT=ALL for step STEP1).<br><br>Note that the position of the IOO$SOFF DD statement within the step JCL does not matter. |
| ```//STEP2    EXEC PGM=PROG1```<br>```//SYSIN    DD   DISP=SHR,DSN=MY.CONTROL.FILE```<br>```//SYSUT2   DD   DISP=SHR,DSN=MY.OUTPUT.FILE```<br>```//SYSUT1   DD   DISP=SHR,DSN=MY.INPUT.FILE```<br>```//IOO$SON  DD   DUMMY``` | All three DD statements (SYSIN, SYSUT2, and SYSUT1) will be selected for IOO processing due to the presence of DDname IOO$ON in the step JCL. If DDname IOO$ON had not been present in the step JCL, the files would have been bypassed from IOO processing by the second DCF rule (the BYPASS=ALL for step STEP2).<br><br>Note that the position of the IOO$SON DD statement within the step JCL does not matter. |

### GBT EREGION=*value*{K/KB/M/MB}

This override record identifies the extended region size (the region above the 16MB line) to be set by IOO's Dynamic Memory Management (DMM) feature. Coding rules are:

- The value must be between 0 and 536,870,912 (512MB).

- Suffixes of K, KB, M, and MB are supported.

- The default value in the shipped global table varies with the selected start-up option as follows:

  • Start-up options K0, K1, K3 and K4 use a default of 128MB.

  • Start-up options K2 and K5 use a default of 256MB.

Example:

| Example | Meaning |
|---|---|
| GBT EREGION=300M | Use an extended region size of 300MB. |

### GBT SETBLK={YES|NO}

This override record identifies the status of the IOO reblocking feature. The following values are supported:

| Value | Meaning |
|---|---|
| GBT SETBLK=NO | IOO reblocking is not to be allowed. |
| GBT SETBLK=YES | IOO reblocking is to be allowed. |

The default value in the shipped global table varies with the selected start-up option as follows:

- Start-up options K0-K2 use a default of GBT SETBLK=NO.

- Start-up options K3-K5 use a default of GBT SETBLK=YES.

Example:

| Example | Meaning |
|---|---|
| GBT SETBLK=YES | Allow IOO reblocking. |

### GBT SETBUF={YES|NO}

This override record identifies the status of the IOO rebuffering feature. This record applies to both VSAM and non-VSAM rebuffering. The following values are supported:

| Value | Meaning |
|---|---|
| GBT SETBUF=NO | IOO rebuffering is not to be allowed. |
| GBT SETBUF=YES | IOO rebuffering is to be allowed. This value is the default in the shipped global table. It applies to all start-up options. |

Example:

| Example | Meaning |
|---|---|
| GBT SETBUF=YES | Allow IOO rebuffering. |

### GBT SETRGN={YES|NO}

This override record identifies the status of the IOO Dynamic Memory Management (DMM) feature. The following values are supported:

| Value | Meaning |
|---|---|
| GBT SETRGN=NO | DMM is not to be activated. |
| GBT SETRGN=YES | DMM is to be activated. This value is the default in the shipped global table. It applies to all start-up options. |

Example:

| Example | Meaning |
|---|---|
| GBT SETRGN=YES | Activate DMM. |

> **NOTE**  See "Dynamic Memory Management (DMM) Considerations" on page 14 for more information about DMM.

### GBT SMF=(*number,options*)

This override record identifies the information that IOO needs for the recording of IOO's user SMF records. This record has the following values:

| Value | Meaning | |
|---|---|---|
| *number* | The SMF record number that IOO is to use. A value of 0 causes no IOO SMF records to be written. A value between 128 and 255 (inclusive) causes IOO SMF records to be written.<br>The default value in the shipped global table is SMF=(0). This default requests that no SMF records be written. It applies to all start-up options. | |
| *options* | Identify which IOO SMF records are to be written. This value need not be coded if an SMF record number of 0 is coded. The following option values are supported: | |
| | QSAMBUF | Write an IOO SMF record when IOO rebuffers a non-VSAM data set. |
| | REBLOCK | Write an IOO SMF record when IOO reblocks a data set. |
| | VSAMBUF | Write an IOO SMF record when IOO rebuffers a VSAM data set. |

Examples

### GBT STC={YES|NO}

This override record identifies whether or not IOO is to select started tasks for optimization. This value sets the default to be used during the processing of DCF rules. It can be overridden by an STC operand coded on a matched DCF rule. The following values are supported:

| Value | Meaning |
|---|---|
| GBT STC=NO | Started tasks are not allowed to be selected by default. |
| GBT STC=YES | Started task are allowed to be selected by default. The shipped global table default is GBT STC=YES. |

Example:

| Example | Meaning |
|---|---|
| GBT STC=YES | Allow started tasks to be selected (unless overridden by the matched DCF rule). |

### GBT TRCLIM=*value*

This override option identifies the maximum number of trace lines that can be issued by the IOO VSAM Record Level Interface for an optimization. Once the trace limit has been reached, no more trace messages will be produced for the optimization being traced. The coded value must be numeric and must be between 0 and 9999.

Example:

| Example | Meaning |
|---|---|
| GBT TRCLIM=100 | Allow no more than 100 trace messages from the IOO VSAM Record Level Interface for each optimized VSAM file. |

### GBT TSOMSG={YES|NO}

This override option identifies whether or not IOO is to produce messages for optimizations that take place within a TSO address space. The following values are supported:

| Value | Meaning |
|---|---|
| GBT TSOMSG=NO | IOO is not to produce messages for optimizations that take place within a TSO address space. |
| GBT TSOMSG=YES | IOO is to produce messages for optimizations that take place within a TSO address space. Any messages produced by IOO will adhere to the global message options:<br><br>■ Messages for reblocking will honor the value set with the GBT BLKMSG option.<br><br>■ Messages for rebuffering will honor the value set with the GBT BUFMSG option. |

### GBT VMAXHSPC=*value*{K|KB|M|MB}

This override option identifies the maximum hiperspace buffers that IOO will allow for a data or index component. Coding rules are:

■ The numeric value must be between 0 and 134,217,728 (128MB).

■ Suffixes of K, KB, M, and MB may be used.

■ A value of 0 for VMAXHSPC indicates that no hiperspace buffers are to be used.

The shipped global table default is GBT VMAXHSPC=32MB.

Example:

| Example | Meaning |
|---|---|
| GBT VMAXHSPC=16MB | Allow a maximum of 16MB hiperspace buffers. |

### GBT VMINHSPC=*value*{K|KB|M|MB}

This override option identifies the minimum hiperspace buffers that IOO will allow for a data or index component. Coding rules are:

■ The numeric value must be between 0 and 134,217,728 (128MB).

- Suffixes of K, KB, M, and MB may be used.

The shipped global table default is GBT VMINHSPC=0.

Example:

| Example | Meaning |
|---|---|
| GBT VMINHSPC=0 | Do not enforce a minimum value. |

### GBT VMAXSPC=*value*{K|KB|M|MB}

This override option identifies the maximum bufferspace that IOO will allow for both the data and index components. Coding rules are:

- The numeric value cannot exceed 134,217,728 (128MB).

- Suffixes of K, KB, M, and MB may be used.

The shipped global table default is 32MB.

Example:

| Example | Meaning |
|---|---|
| GBT VMAXSPC=16MB | Allow a maximum of 16MB bufferspace. |

### GBT VMINSPC=*value*{K|KB|M|MB}

This override option identifies the minimum bufferspace that IOO will allow for an NSR optimization when either of the following is true:

- The optimization rule contained a specific bufferspace.

- The use of NSR was being forced. The enforcing of a minimum value ensures the presence of sufficient buffers for instances where the IOO VSAM Record Level Interface is switching from LSR optimization to NSR optimization.

Coding rules are:

- The numeric value cannot exceed 134,217,728 (128MB).

- Suffixes of K, KB, M, and MB may be used.

The shipped global table default is 32KB.

Example:6

| Example | Meaning |
|---|---|
| GBT VMINSPC=64MB | Ensure a minimum of 64KB bufferspace. |

# Chapter 4
# Statistics

StarTool IOO lets you see the attributes and activity associated with each VSAM cluster through statistical reports. There are two types of statistic reports:

- **Basic**. The basic statistics contain VSAM cluster information such as Data Buffers, Index Buffers, CI Size, CI Splits, and CA Splits.

- **Extended**. Extended statistics contain all of the information contained in the basic statistics plus additional information including gets, puts, points, verifies, and deletes.

The statistics appear in your job output under the DD NAME JESYSMSG JES2.

# Activating Statistics

To globally turn on and off the statistics, you must alter the Global Rules Table (see Appendix A, "Rules Tables"). The two parameters that affect the statistics are STATS and STATTHLD.

STATS allows you to turn the statistics feature off and on. In addition, STATS lets you choose the type of statistics, basic or extended. The format of the STATS parameter is:

> STATS=(<u>YES</u>|NO,
>      <u>BASIC</u>|EXTENDED)

The default value for STATS is STATS=(YES, BASIC).

**NOTE** If extended statistics are turned on, more CPU cycles are needed for each I/O.

STATTHLD determines the threshold (that is, the EXCP count of the data component of the VSAM data set) before the statistics are produced. This parameter allows you to defer the statistics for those programs with little or no I/O. For example, if STATTHLD=0, then each time the program ends, statistics are produced. The format of the STATTHLD parameter is:

> STATTHLD=*xxx*

> where *xxx* is a number from 0 to 32K.

The default value for STATTHLD is 10. If you want to turn the statistics feature on or off for a particular job, refer to "System Rules" on page 91.

# Basic Statistics

This section describes the fields found in StarTool IOO statistics. The statistics that you see depend on your job, whether it is run in Nonshared Resources (NSR) or Local Shared Resources (LSR).

For NSR and LSR jobs, information includes data buffers, index buffers, CI size, CI splits, and CA splits. For LSR jobs, additional information such as buffer lookasides, user-initiated writes and VSAM forced writes are included in the statistics output.

## LSR Example with Shared Data/Index Components

Here is an example of statistics when a file is placed in LSR and the data/index
components share the same LSR pool:

```
                        StarTool IOO 315
  Copyright(c)2001-2003 SERENA Software, Inc. All rights reserved

  DDname:  DD1      Cluster: somnode.CLUSTER
  Program: READSEQ  LSR      Data/Index share LSR pool  POOLID: 15
                    Buffer Count: 90

  ACBMACRF: 8A0048 (KEY,DIR,OUT,LSR,DFR)

  Data:  CISIZE:  4,096        CI Splits: 4        CA Splits: 1
         Reads:   10,444       Deletes:   22       Extents:   1
         Records: 2,145,012    Inserts:   1,890    Updates:   543
         EXCPs:   1,945
         Buffer LOOKASIDES:      9,043
         User-Initiated Writes: 1,450
         VSAM Forced Writes:     669

  Index: CISIZE:  4,096        CI Splits: 0        CA Splits: 0
         Reads:   0            Deletes:   0        Extents:   1
         Records: 10,458       Inserts:   0        Updates:   0
         EXCPs:   1,854        Levels:    2
```

## LSR Example with Nonshared Data/Index Components

This is an example of a file in LSR where the Data/Index components do not share the
same LSR pool. Notice that the difference between this statistics block and the one above
is that the LSR information is detailed for both the Data and Index. Moreover, both the
Data and Index have Buffer LOOKASIDES, User-Initiated Writes, and VSAM Forces Writes.

Also, since the data and index do not share the same pool, you see exactly how many buffers for each. In this case, there are 25 Data Buffers and 5 Index Buffers.

```
                          StarTool IOO 315
            Copyright(c)2001-2003 SERENA Software, Inc. All rights reserved
DDname:    VSAM0001   Cluster: somnode.VSAM.NONIMBED.CL1
Program:  COBOL300   LSR       SEPARATE Data/Index POOLS   POOLID: 255
                      Data Buffers: 25            Index Buffers: 5


ACBMACRF: 9C0048 (KEY,SEQ,DIR,IN,LSR,DFR)


Data:  CISIZE:  4,096        CI Splits: 0              CA Splits: 0
       Reads:   0            Deletes:   0             Extents:  2
       Records: 1,000        Inserts:   0             Updates:  0
       EXCPs:   0
       Buffer LOOKASIDES:    0
       User-Initiated Writes: 0
       VSAM Forced Writes:    0


Index: CISIZE:  1,024        CI Splits: 0             CA Splits: 0
       Reads:   0            Deletes:   0             Extents:  1
       Records: 3            Inserts:   0             Updates:  0
       EXCPs:   0            Levels:    2

       Buffer LOOKASIDES:    0
       User-Initiated Writes: 0
       VSAM Forced Writes:    0
```

# Field Definitions

| Field | Definition |
|-------|-----------|
| DDname | The DDname allocated to the Cluster. |
| Cluster | The Data Set Name (DSN) of the Cluster. |
| Program | The program name that issued the OPEN for the listed DDname/DSN. |
| LSR | The file (DDname/DSN) was placed into LSR (Local Shared Resources). The Data and Index components share the LSR pool. |
| POOLID | The LSR pool number into which the file was placed. |
| Buffer Count | The number of buffers in the LSR pool which was built by StarTool IOO. |
| ACBMACRF | These are the flag bytes (3 of them) located within the ACB (Access Control Block) that specify processing options. This is a complete list of processing options:<br><br>`KEY    ADR    ADD    CNV`<br>`BLK    SEQ    DIR    IN`<br>`OUT    UBF    SKP    RST`<br>`DSN    AIX    LSR    ICI`<br>`DFR    SIS    NCFX   MODE` |
| Data | The following information is associated with the data component of the file: |
| | CI Size | Control Interval size of the data component. |
| | CI Splits | Number of Control Interval splits that occurred during this program execution. |
| | Reads | Number of records read during this program execution. |
| | Deletes | Number of records deleted during this program execution. |
| | Extents | Number of extents currently allocated to the file. |
| | Records | Number of records currently held within the file. |
| | Inserts | Number of records inserted (added) to the file during this program execution. |
| | Updates | Number of records updated during this program execution. |
| | EXCPs | The number of EXCPs charged to the data component during this program execution. |

If the file was placed into LSR, this information is also displayed.

| LSR specific information | |
|--------------------------|--|
| Buffer LOOKASIDES | Number of times a request for record retrieval could be satisfied without an I/O operation. |
| User-Initiated Writes | Number of times the user-initiated writes from the buffer pool. |
| VSAM Forced Writes | Number of writes VSAM was forced to do because no buffers were available. |

| LSR specific information | |
|---|---|
| Index | The same information that is being detailed for the data component is listed for the index. The only additional piece of information that is listed under the Index that is not listed under the Data is Levels. |
| Levels | Number of index levels currently allocated to the file. |

## NSR Example

```
                      StarTool IOO 315
          Copyright(c)2001-2003 SERENA Software, Inc. All rights reserved

 DDname:   VSAM0003   Cluster: WSER132.VSAM.NONIMBED.CL1
 Program:  COBOL300   NSR      Data Buffers: 13      Index Buffers: 5

 ACBMACRF: 940000 (KEY,SEQ,IN)

 Data:  CISIZE:  4,096      CI Splits: 0          CA Splits: 0
        Reads:   997        Deletes:   0          Extents:   2
        Records: 1,000      Inserts:   0          Updates:   0
        EXCPs:   3

        Gets:    996        Seq:     996          Direct:  0
        Reads:   996        Seq:     996          Direct:  0


 Index: CISIZE:  1,024      CI Splits: 0          CA Splits: 0
        Reads:   0          Deletes:   0          Extents:   1
        Records: 3          Inserts:   0          Updates:   0
        EXCPs:   3          Levels:    2
```

# Extended Statistics

Extended statistics includes all of the information found in the basic statistics and more. It shows the activity associated with the VSAM cluster including Gets, Puts, Checks, Points, Deletes and Verifies.

# Example of LSR Extended Statistics

```
                        StarTool IOO 315
         Copyright(c)2001-2003 SERENA Software, Inc. All rights reserved

DDname:   VSAM0002   Cluster: somnode.VSAM.NONIMBED.CL2
Program:  COBOL300   LSR      SEPARATE Data/Index POOLS   POOLID: 254
                     Data Buffers: 50           Index Buffers: 27
                     Hiperspace:   50           Hiperspace:    07


ACBMACRF: 920040 (KEY,SEQ,OUT,LSR)

Data:  CISIZE:  4,096       CI Splits: 0          CA Splits: 0
       Reads:   4           Deletes:   0          Extents:   5
       Records: 200,000     Inserts:   0          Updates:   4
       EXCPs:   1
       Buffer LOOKASIDES:     1
       User-Initiated Writes: 0
       VSAM Forced Writes:    0

       Gets:    4           Seq:       4          Direct:  0
         Update: 4          Seq:       4          Direct:  0

       Puts:    4           Seq:       4          Direct:  0
         Update: 4          Seq:       4          Direct:  0


Index: CISIZE:  2,048       CI Splits: 0          CA Splits: 0
       Reads:   0           Deletes:   0          Extents:   1
       Records: 26          Inserts:   0          Updates:   0
       EXCPs:   2           Levels:    2

       Buffer LOOKASIDES:     0
       User-Initiated Writes: 0
       VSAM Forced Writes:    0
```

## Example 2 of LSR Extended Statistics

```
                        StarTool IOO 315
          Copyright(c)2001-2003 SERENA Software, Inc. All rights reserved

DDname:   VSAM0003   Cluster: WSER132.VSAM.NONIMBED.CL1
Program:  COBOL300   LSR       SEPARATE Data/Index POOLS   POOLID: 253
                     Data Buffers: 25            Index Buffers: 5
DMSCLOSE
ACBMACRF: 940040 (KEY,SEQ,IN,LSR)

Data:  CISIZE:  4,096       CI Splits: 0            CA Splits: 0
       Reads:   5           Deletes:   0            Extents:   2
       Records: 1,000       Inserts:   0            Updates:   0
       EXCPs:   1
       Buffer LOOKASIDES:     1
       User-Initiated Writes: 0
       VSAM Forced Writes:    0

       Gets:     5          Seq:        5           Direct:  0
         Reads:  5          Seq:        5           Direct:  0

       Points:   1          Seq:        1           Direct:  0

Index: CISIZE:  1,024       CI Splits: 0            CA Splits: 0
       Reads:   0           Deletes:   0            Extents:   1
       Records: 3           Inserts:   0            Updates:   0
       EXCPs:   2           Levels:    2

       Buffer LOOKASIDES:     0
       User-Initiated Writes: 0
       VSAM Forced Writes:    0
```

# Appendix A
# Rules Tables

This appendix describes the advanced functionality of StarTool IOO that allows for unique tuning requirements. It also gives insight into the system facilities that StarTool IOO controls and manipulates to achieve its performance results.

**NOTE**  The default rules allow complete use of the product. It is not necessary to ever code rules. We strongly recommend that you use default rules that contain thoroughly tested rules with all known compatibility problems bypassed, because you could introduce errors by coding new rules.

# Controlling StarTool IOO using Rules

You can control StarTool IOO through customizable tables. If you want to make rule modifications, read this section to understand how to code and customize them. The three rules tables are as follows.

| Table | Description |
|---|---|
| Global Rules | Contains StarTool IOO global rules and startup options. |
| Device Rules | Allows you to tailor different blocksize targets/goals for different devices in order to capitalize on device performance differences. |
| System Rules | Allows you to signify certain files to SELECT, EXEMPT, or BYPASS for types of optimization. |

**NOTE**  We reserve the right to change any or all rules (except DCF) at any time. Therefore, it is best that selections/bypasses be accomplished through DCF.

# Global Rules

The purpose of the Global Rules Table is to define global rules and startup options.

## Assembly and Link-Edit

The load module generated by assembling and link-editing these definitions must have the name ULTIGB*xx* where *xx* is a 2-character alphanumeric suffix.This member must reside in the same load library as the other StarTool IOO load modules. In addition, the load module must be link-edited with an authorization code of 1.

## Sample Definitions

The source of the distributed ULTIGB*xx* table is included in the STRIOO.V*v*R*r*M*m.*SRC distribution library under member name ULTIGB*xx*. The JCL to assemble and link-edit this table is included in the STRIOO.V*v*R*r*M*m.*JCL library under member name IOOGASM.

## Table Format

The format and description of the ULTIGBL macro, used to generate the ULTIGB*xx* table, is as follows.

| ULTIGBL | MODE=MIXMOD | Mode Of Operation |
|---|---|---|
| | SYSUFX=*xx* | Defines the System Table suffix (ULTITB*xx*) to be used for StarTool IOO processing. |

| | | |
|---|---|---|
| | DVSUFX=*xx* | Defines the Device Table suffix (ULTIDV*xx*) to be used for StarTool IOO processing. |
| | BLKMSG={WTO\|WTL\|NO} | Specifies whether or not the reblocking message is issued as a WTO (routed to the SYSLOG, JOBLOG and CONSOLE) or WTL (routed to the SYSLOG only). Specifying NO suppresses all reblocking messages. |
| | BLSR=OVERRIDE\|IGNORE | **OVERRIDE** Batch LSR (prevent Batch LSR from getting control) and use StarTool IOO to optimize.<br>**IGNORE** skips StarTool IOO processing and does not optimize the file if Batch LSR is going to optimize it. |
| | BUFMSG={WTO\|WTL\|NO} | Specifies whether or not the rebuffering message is issued as a WTO (routed to the SYSLOG, JOBLOG and CONSOLE) or WTL (routed to the SYSLOG only). Specifying NO suppresses all rebuffering messages. |
| | CSRERR=IGNORE\|TRACE | Specifies whether or not to **IGNORE** or **TRACE** the Catalog Service Routine Error Messages. |
| | DBM={YES\|NO} | Used to turn on/off Dynamic Buffer Management. |
| | DBMISIZE={*#chars*} | Threshold size for index buffers with Dynamic Buffer Management. Can specify M, MB, (M+*x*), (M-*x*), K, KB, (K+*x*), (K-*x*) or *x* where *x* equals the number of bytes. For example, you can specify 5242879 or 5M-1 to obtain identical results. |
| | DBMDSIZE={*#chars*} | Threshold size for data buffers with Dynamic Buffer Management. Can specify M, MB, (M+*x*), (M-*x*), K, KB, (K+*x*), (K-*x*) or *x* where *x* equals the number of bytes. For example, you can specify 5242879 or 5M-1 to obtain identical results. |
| | DBT={YES\|NO} | Used to turn on/off Dynamic Buffer Translation. |
| | DCFCTMX=*xxx* | Maximum number of entries in DCFCT (Dynamic Control Facility Control Table). |
| | DCFPROC=IOODCF | Dynamic Control Facility startup procedure name. |

| | DMMCNTL=<u>DCF</u>\|GLOBAL | GLOBAL signifies Dynamic Memory Management (DMM) is on globally. All jobs/steps have their regions/address-spaces optimized unless you specify a BYPASS=ALL in DCF. All regions are optimized for memory management regardless of whether or not StarTool IOO optimized their I/O or not.<br><br>DCF (default) signifies Dynamic Control Facility (DCF) controls DMM whereby only those jobs/ steps specifically selected have their regions/address-spaces optimized. This memory management depends on whether or not the job was selected by DCF or DDname, and it is independent of the internal System Rules. Jobs that are *not* selected by DCF or //ULT@ DDname will *not* have their regions optimized. Jobs that *are* selected by DCF, but not selected in the internal System Rules still have their regions optimized. |
|---|---|---|
| | DMS=({YES\|NO},{MAX\|MIN} | Turn on/off Dynamic Mode Selection and determine how it functions (see DMSHWM). |
| | EXITNM=ULTIEXT | Identifies the name of Ulti-BLOCKs user exit load module. |
| | FEATURES=(OPM31,DCF) | Level dependant features (Micro Focus internal use only). |
| | HPERDLWM=(*#buffers,#chars*) | The data Hiperspace low-water-mark. After normal virtual storage buffers are filled with data (and Hiperspace is eligible for use), these parameters control the minimum additional number of buffers of data or minimum additional amount of characters of data that must exist before Hiperspace is created. If the amount of data that will be put into a Hiperspace is less than either amount (*#buffers* or *#chars*), additional normal above-the-16M-line virtual storage is used and Hiperspace is not used. |

| | | |
|---|---|---|
| | HPERILWM=(*#buffers*,*#chars*) | The index Hiperspace low-water-mark. After normal virtual storage buffers are filled with indexes (and Hiperspace is eligible to be used), these parameters control the minimum additional number of buffers of indexes or minimum additional amount of characters of indexes that must exist before Hiperspace is created. If the amount of index-data that will be put into a Hiperspace is less than either amount (*#buffers* or *#chars*), additional normal above-the-16M-line virtual storage is used and Hiperspace is not used. |
| | ICON=YES | Generate StarTool IOO icon. |
| | IEALIMIT=*xxx* | Specifies the amount of region per job step by which the region limit is increased. Can specify M, MB, (M+*x*), (M-*x*), K, KB, (K+*x*), (K-*x*) or *x* where *x* equals the number of bytes. |
| | LSQA=*xxx* | The cushion amount of region that dynamic memory management (DMM) leaves available for Local System Queue Area (LSQA). DMM subtracts this amount from the maximum region available to accommodate any requests for additional LSQA. Can specify M, MB, (M+*x*), (M-*x*), K, KB, (K+*x*), (K-*x*) or *x* where *x* equals the number of bytes. |
| | LSRMAX=255 | Specifies the maximum number of LSR pools that IOO builds for any jobstep. The maximum is 255 (default). We recommend that you run StarTool IOO using the default value of 255. |
| | MAXHSPC=*xxx* | Specifies the maximum amount of Hiperspace storage StarTool IOO allocates per file. Can specify M, MB, (M+*x*), (M-*x*), K, KB, (K+*x*), (K-*x*) or *x* where *x* equals the number of bytes. |
| | MAXSPC=*xxx* | Specifies the maximum amount of QSAM/VSAM buffer space StarTool IOO allocates per file. Can specify M, MB, (M+*x*), (M-*x*), K, KB, (K+*x*), (K-*x*) or *x* where *x* equals the number of bytes. |

| | | |
|---|---|---|
| | MINSPC=*xxx* | Specifies the minimum amount of QSAM/VSAM buffer space StarTool IOO allocates per file. Can specify M, MB, (M+*x*), (M-*x*), K, KB, (K+*x*), (K-*x*) or *x* where *x* equals the number of bytes. |
| | MINHSPC=*xxx* | Specifies the minimum amount of Hiperspace storage StarTool IOO allocates per file. Can specify M, MB, (M+*x*), (M-*x*), K, KB, (K+*x*), (K-*x*) or *x* where *x* equals the number of bytes. |
| | PGMDCB={YES|NO} | Specifies whether or not Ulti-BLOCK is to optimize block sizes that are hard coded in programs. We recommend that PGMDCB=YES be specified. |
| | REGNHWM={*#chars*} | Specifies the region High-Water Mark for dynamic memory management (DMM). Any region that is less than or equal to this value gets the maximum amount of region available when the first file in a step is opened. If the region is greater than this value, DMM processing is bypassed. Can specify M, MB, (M+*x*), (M-*x*), K, KB, (K+*x*), (K-*x*) or *x* where *x* equals the number of bytes. For example, specify 5242879 or 5M-1 to obtain identical results. |
| | RPG={YES|NO} | If your installation is using RPG or COBOL, specify RPG=YES to dynamically correct problems caused by coding inconsistencies that were not apparent until the block size was changed. |
| | SETBLK={YES|NO} | Specifies whether or not to optimize the block size of a file. This option can be overridden by the SETBLK option in StarTool IOO System Table. If SETBLK=YES is specified in the Global Table and SETBLK=NO is specified on an individual selection in the System Table, Ulti-BLOCK generates a reblocking message for informational purposes only (the physical block size is not changed). |
| | SETBUF=(YES/NO) | Specifies whether or not to optimize the number of buffers in the DCB or ACB. This option can be overridden by the SETBUF option in StarTool IOO System Table. |

| | | |
|---|---|---|
| | SETMAC={YES\|NO} | Specifies whether or not to override the MACRF in the ACB. If SETMAC=NO is specified, the OVRMAC option in the System Table does not take effect. |
| | SETRGN=YES | Turn On Dynamic Memory Management (DMM). See also the FACILITY parameter. |
| | SHROPT4={YES\|<u>NO</u>} | Process SHROPT4 clusters (YES/ NO). |
| | SMFREC=[0,REBLOCK] or<br><br>SMFREC=[(*rec#*,VSAMBUF, QSAMBUF,REBLOCK, {SELECT\|ALL})] | No SMF records built.<br><br>StarTool IOO writes an SMF user record for VSAM rebuffering, QSAM rebuffering, and/or QSAM reblocking for SELECTed files or ALL possible files. SMFREC=(0,REBLOCK) suppresses all of StarTool IOO SMF user records. The SMF produced has the standard SMF header. |
| | SORTIN=(SORTIN) | SORTIN must reference the input ddname (such as SORTIN, SORTIN01, SORTIN02) used in any of the common sort products. Since most sort products use their own unique EXCP processing, Ulti-BLOCK has to handle sort processing differently than QSAM/ VSAM/BSAM processing. This option builds an internal list that defines sort input ddnames requiring special EXCP processing. |
| | SSN= | StarTool IOO Internal Subsystem name. *Do not change.* |
| | STATS=(<u>YES</u>\|NO, <u>BASIC</u>\|EXTENDED) | STATS turns StarTool IOO statistic feature on and off. Specify whether or not you want BASIC statistics or EXTENDED statistics. If you specify STATS=(YES), then you get YES,BASIC. To turn the statistics feature off, specify STATS=(NO). The default is STATS=(YES,BASIC).<br>If the extended statistics are turned on, more CPU cycles are needed for each I/O. |

| | | |
|---|---|---|
| | STATTHLD=*xxx* | *xxx* defines the threshold that must be met or exceeded before statistics are printed. This keyword allows you to limit the amount of statistics from jobs that do little or no I/O.<br><br>*xxx* can be a number from 0 to 32K. If you specify STATTHLD=0, then each time a data set is CLOSED, you will see the statistics. The default value is 10. |
| | TRACE={YES\|NO} | Specify YES to activate the StarTool IOO log facility to see more detail about each optimization or if directed by the Customer Support Center. This is not a trace as far as the amount of overhead consumed, but a log. |
| | TSOMSG={YES\|NO} | Specifies whether or not to issue blocking and buffering messages to TSO users. |
| | TSOVSAM={YES\|NO} | Specifies whether or not to optimize VSAM in TSO. |
| | VSIERR=IGNORE\|TRACE | Specifies if the VSAM System Interface Routine Error Messages should be IGNOREd or TRACEd. |
| | DD_SOFF=ULT@SOFF | Is the DDname used in JCL for bypassing all IOO optimizations for the current jobstep. The specified DDname is coded as follows:<br>`  //ULT@SOFF DD DUMMY` |
| | DD_SON=ULT@SON | Is the DDname used in JCL for forcing IOO optimization of all DDnames in the current jobstep. The specified DDname is code as follows:<br>`  //ULT@SON DD DUMMY` |
| | DD_OFF=ULT@OFF | Is the DDname used in JCL for bypassing all IOO optimizations in the current jobstep until the DD_ON DDname is reached. |
| | DD_ON=ULT@ON | Is the DDname used in JCL for nullifying the effect of the DD_OFF DDname. For example, the following DDnames result in the optimization of DD1, DD2, and DD5: |

```
//DD1       DD DISP=SHR,DSN=dsname1
//DD2       DD DISP=SHR,DSN=dsname2
//ULT@OFF   DD DUMMY
//DD3       DD DISP=SHR,DSN=dsname3
//DD4       DD DISP=SHR,DSN=dsname4
//ULT@ON    DD DUMMY
//DD5       DD DISP=SHR,DSN=dsname5
```

| | |
|---|---|
| ELIMIT=4MB | The ELIMIT keyword does for 31 bit storage what the IEALIMIT keyword does for 24 bit storage. |
| EREGION=64MB | The EREGION keyword does for 31 bit storage what the REGNHWM keyword does for 24 bit storage. |
| FACILITY=$SERENA.IOO.DMM | The FACILITY keyword specifies the name of the RACF resource profile used to protect the DMM feature. See "Dynamic Memory Management (DMM) Considerations" on page 14 for more information. |
| LBI=YES | Determines whether IOO will use IBM's Large Block Interface (LBI) to help optimize data set accesses. |
| LBIMAX=256KB | Sets a ceiling on the blocksize IOO will set when using IBM's Large Block Interface (LBI) to help optimize data set accesses. |
| LSR_IMBED=YES | Determines whether IOO will adopt LSR for a VSAM data set defined with the IMBED attribute. |
| LSQALEXT=128KB | Defines the minimum size of the largest free LSQA extent IOO's DMM feature will strive to achieve. |
| MINHSPC=0 | If IOO determines, based on the VSAM cluster definitions, that the LSR buffers are to be placed in a hiperspace, this keyword sets a minimum size for that hiperspace. The value coded must be less than the MAXHSPC specification. |
| PRODUCTS=(IOO-VSAM,IOO-LSR, IOO-BLOCK,IOO-SAM, IOO-IMS) | Specifies the IOO features that are to be activated. |
| SMS=NO | Determines whether IOO will use the System-Determined Blocksize feature of DFSMS when an input data set has a blocksize of zero. |
| STC=YES | Determines whether IOO will perform optimizations for data sets being accessed by started tasks (STCs). |

| | | |
|---|---|---|
| | TSONVRGN=NO | Determines whether IOO's DMM feature will optimize TSO regions. |
| | TRCLIM=*9999* | Sets a ceiling on the number of messages IOO will issue when asked to trace VSAM optimizations. You can specify a number from 1 to 9999. |
| | VMAXSPC=32MB | Imposes a maximum buffer space that IOO will request when performing optimizations of a non-VSAM data set. The value coded must be greater than both zero and the VMINSPC specification. |
| | VMINSPC=32KB | Imposes a minimum buffer space that IOO will request when performing optimizations of a non-VSAM data set. The value coded must be less than the VMAXSPC specification. |

# Example of Global Rules Table

```
ULTIGBL MODE=MIXMOD,            MODE OF OPERATION
        SYSUFX=BO,              SYSTEM TABLE SUFFIX
        DVSUFX=SA,              DEVICE TABLE SUFFIX
        , _____MESSAGES_____
        BLKMSG=NO,              OPTIMIZATION MSG (WTO/WTL/NO)
        BUFMSG=WTO,             BUFF CHANGE MSG(WTO/WTL/NO)
        , _____ACTIONS_____
        SETBUF=YES,             CHANGE QSAM BUFFERS (YES/NO)
        SETMAC=YES,             CHANGE VSAM MACRF (YES/NO)
        SETBLK=NO,              REBLOCK DATASETS? (YES/NO)
        PGMDCB=YES,             DO HARD CODED DCB'S? (YES/NO)
        SORTIN=(SORTIN,SORTIN01,SORTIN02,SORTIN03,SORTIN04,
        SYSUT1),
        , _____TSO_____
        TSOMSG=NO,              MESSAGES TO TSO USERS? (YES/NO)
        TSOVSAM=NO,             OPTIMIZE VSAM FROM TSO?
        , _____TRACE ENTRIES_____
        TRACE=NO,               DETAIL INFO TRACE FACILITY
        TWTOLIM=50,             TRACE WTO MSG LIMIT          R0
        TWTLLIM=100,            TRACE WTL MSG LIMIT          R0
        , _____VSAM ESA OPTIMIZATION_____
        MINSPC=32K,             MINIMUM VSAM BUFF SPACE
        MAXSPC=4M,              MAXIMUM VSAM BUFF SPACE
        MINHSPC=0,              MINIMUM HIPERSPACE VALUE
        MAXHSPC=5M,             MAXIMUM HIPERSPACE VALUE
        BLSR=OVERRIDE,          OVERRIDE BATCH LSR W/ULT OPTIMI
        DBM=YES,                TURN ON DUAL BUFF MGMT
        DBT=YES,                TURN ON DYNAM BUFF TRANSLATION
        DBMISIZE=128K,          DBM THRESHOLD SIZE 4 INDEX BUFS
        DBMDSIZE=128K,          DBM THRESHOLD SIZE 4 DATA BUFS
        DMS=(YES,MIN),          TURN ON DYNAM MODE SELECTION
        HPERILWM=(10,96K),      HIPERSPCE INDEX LOW-WATER MARK
        HPERDLWM=(10,96K),      HIPERSPCE DATA  LOW-WATER MARK
        SHROPT4=NO,             PROCESS SHROPT(4) CLUSTERS?
        EREGION=128M,           EREGION FOR VSAM ABOVE-THE-LINE
        ELIMIT=4M,              ELSQA LIMIT FOR ABOVE-THE-LINE
        , _____COMMON FUNCTIONS_____
        DCFPROC=IOODCF,         DCF PROCNAME FOR ACTIVATION
        DCFCTMX=400,            DCF CNTL TABLE MAXIMUM
        SMFREC=(0,SELECT),      DO NOT WRITE ULTI-SMF RECORDS
        RPG=YES,                CODE YES IF USING RPG
        STATS=YES,              CODE YES FOR STATISTICS
        STATTHLD=10,            EXCP THRESHOLD FOR STATISTICS
        EXITNM=ULTIEXT,         USER EXIT DUMMY NAME(BR 14)
        SETRGN=YES,             DMM OPTIMIZATION (YES/NO)
        DMMCNTL=DCF,            CONTROL OPTIMIZING MEMORY W/DCF
        LSQA=700K,              AMNT OF REGION LEFT FOR LSQA
        LSQALEXT=128K,          MINIMUM CONTIGUOUS EXTENT 4 LSQA
        IEALIMIT=128K,          IEALIMIT GETMAIN LIMIT
        , _____ESA REGION_____
        REGNHWM=12M             REGION HIGH-WATER-MARK
    END
```

# Device Rules

The Device Rules Table is used for SAM reblocking and specifies optimal physical blocksizes based on the physical characteristics of a particular device-type (for example, 3380 = 1/2 track blocking); it is not used for VSAM. The Device Table allows you to tailor different blocksize targets/goals for different devices to capitalize on device performance differences. This is done by defining the target maximum block sizes to be used for a particular device. Performance and device utilization improvements are dependent on the target values defined.

**NOTE** Any data set created or residing on a device type not defined in this table will not be optimized by StarTool IOO.

Choose the Device Table that StarTool IOO uses by modifying the DVSUFX parameter in the Global Rules Table. See "Table Format" on page 80 for more information on the DVSUFX parameter.

A complete table of Device Rules is sent with StarTool IOO. Although it may be modified to your performance objectives, use the default table delivered with the product.

## Assembly and Link-Edit

The load module generated by assembling and link-editing these definitions must be named ULTIDVxx where xx is a two-character alphanumeric suffix. This module must reside in the same library where all other StarTool IOO modules were installed. In addition, the load module must be link-edited with an authorization code of 1.

To use the new device table, change DVSUFX=xx to the alphanumeric suffix used in the link-editing above.

## Sample Definitions

The source of the distributed ULTIDVT table is included in the STRIOO.V*v*R*r*M*m*.MAC library under member name ULTIDVT. The JCL to assemble and link-edit this table is included in the STRIOO.V*v*R*r*M*m*.JCL library under member name IOODASM. The JCL parameters, such as the unit names in this assembly JCL, must be changed to your installation standards.

## Table Format

The format of the ULTIDVT macro used to generate the ULTIDVxx table is as follows:

| ULTIDVT | DEVICE=*device type* | Defines the device type to be optimized by StarTool IOO. The valid values are 3330, 3340, 3350, 3375, 3380, 3390, 3420, 3480 or 3490. The allowable values for this parameter are increased as new devices become available. (Required entry) |
|---|---|---|
| | BLKSIZE=*maximum block size* | Defines the maximum block size to be used when optimizing a data set residing on the specified device type. You can specify half track, full track, or user defined block size targets for the device. The maximum value that can be coded is 32760. All the default values are in the `STRIOO.V`*v*`R`*r*`M`*m*`.MAC` distribution library under the member name ULTIDVSA. (Required entry) |
| | [{,BUFFER=(NUMBER,*xx*)}] or [{,BUFFER=(TRACKS,*xx*)}] or [{,BUFFER=(SPACE,*xxxxxx*)}] | *xx* is number of buffers. Not normally used. *xx* is number of tracks. StarTool IOO uses the BLOCKSIZE and TRACK SIZE to calculate the number of buffers for a fill. *xx* is the total number of bytes in storage to be used for buffers for a file. |

# System Rules

The System Rules Table defines the resources that must be optimized or not be optimized. StarTool IOO first examines the EXEMPT and BYPASS entries to determine if the data set should not be optimized. If no BYPASS entries exist for the data set name, then StarTool IOO searches for an exact match with any SELECT statements. If a matching SELECT statement is found for the data set, optimization occurs. If the data set does not match any SELECT statement, no optimization occurs. All precoded table definitions are shipped to you using both EXEMPTs, BYPASSes and SELECTs.

The System Rules Table used by StarTool IOO is the one with the two-character suffix that matches the option chosen when you started or modified StarTool IOO, such as S STARTIOO OPTION=*xx*. See for more information on selecting StarTool IOO options.

Two types of system rules exist in the System Rules Table. Both are used by Micro Focus and may change with different releases of StarTool IOO.

| System Rule | Description |
| --- | --- |
| Selection Rules | Define to StarTool IOO what Micro Focus does not allow to be optimized through a set of BYPASS and EXEMPT rules and SELECT rules that select everything and should *not* be used for user selections (DCF should be used for user selections). |
| Optimization Rules | Specify how and what to do when optimizing. |

# Coding

All the MODE=BYPASS entries must be coded first, followed by MODE=EXEMPT entries and then followed by the MODE=SELECT entries. If the entries are not coded in this sequence, StarTool IOO may ignore some of the MODE=EXEMPT or MODE=BYPASS entries during MIXMOD processing. Any or all of the keywords may be coded on a single entry. StarTool IOO looks for an exact match on all values coded before it decides what processing to perform.

# Assembly and Link-Edit

The load module generated by assembling and link-editing definitions must have a name that consists of the prefix ULTITB and a two-character suffix. Use the two-character suffix of the Systems Rules Table as the option xx when starting and modifying StarTool IOO (OPTIONS=xx). The suffix must be two alphanumeric characters. You can alter these characters to generate variations of the system table to be used by one or more systems. The system table must reside in the same library where the other StarTool IOO modules were installed. In addition, the load module must be link-edited with an authorization code of 1.

# Sample Definitions

The source of the distributed ULTITB*xx* tables are included in the STRIOO.V*v*R*r*M*m*.SRC distribution library under member name ULTITB*xx*. The JCL to assemble and link-edit this table is included in the STRIOO.V*v*R*r*M*m*.JCL library under member name IOOTASM. Some of the JCL parameters, such as the unit names, may have to be changed to conform to your installation standards.

# Table Format

All system table entries use the following format of the ULTITBL macro to create the ULTITBxx table of rules:

ULTITBL

| | | |
|---|---|---|
| MODE= | {BYPASS\|EXEMPT\|SELECT} | Internal search criteria. Not used except by Micro Focus for special circumstances. The user should use DCF. |
| ACCMETH= | {BSAM\|<u>QSAM</u>\|VSAM\|EXCP\|BLDVRP} | |
| JOB= | *jobname*, or<br>ALL | |
| JPREFIX= | *jobname prefix*, or<br>(*jobname prefix,nn*) | |
| STEP= | *stepname* | |
| SPREFIX= | *stepname prefix*, or<br>(*stepname prefix,nn*) | |
| DDNAME= | *ddname* | |
| DPREFIX= | *ddname prefix*, or<br>(*ddname prefix,nn*) | |
| DSNAME= | *data-set-name* | |
| PREFIX= | *data-set-name-prefix*, or<br>(*data-set-name-prefix,nn*) | |
| VOLSER= | *volume serial number* | |
| VPREFIX= | *defines volser # prefix 1-5 characters*, or<br>(*1-5 character vol-ser prefix,nn*) | |
| PROGRAM= | *program name* | |
| PPREFIX= | *program name prefix*, or<br>(*program name prefix,nn*) | |
| DEVADDR= | *3-character device address like 190* | |

| | | |
|---|---|---|
| ASTRNO= | *number* | VSAM NSR only |
| BLKSIZE= | *maximum block size* | QSAM only |
| BUFNI= | (*number index buffers*,CKSTRNO) | VSAM only |
| BUFND= | (*number data buffers*,CKSTRNO) | VSAM only |
| BUFFER= | {(CKSTRNO,*xxxxxx*)} | VSAM only |
| | {(SPACE,*xxxxxx*)} | VSAM/QSAM |
| | {(TRACKS,*xx*)} | QSAM only |
| | {(NUMBER,*xx*)} | QSAM only |

| | | |
|---|---|---|
| BLDVRP= | (AUTO{,SEP} | VSAM LSR only |
| | RMODE31={ALL\|BUFF\|CB\|NONE} | |
| | ,STRNO=*number* | |
| | ,ISIZEMAX={*#chars*}<br>{(#KB or K[,*Hiperspace*])} | |
| | ,DSIZEMAX={*#chars*}<br>{(#KB or K[,*Hiperspace*])} | |
| or | | |
| BUFFERS= | (*size*(*number*[*Hiperspace*]),,...) | VSAM LSR |
| | FIX={BFR\|IOB\|(BFR\|IOB)} | |
| | KEYLEN=*length* | |
| | RMODE31={ALL\|BUFF\|CB\|<u>NONE</u>} | |
| | STRNO=*number* | |
| | SHRPOOL=*number* | |
| CHKMAC= | ({DIR\|<u>SEQ</u>\|SKP\|DYNAM} | |
| | {NSR\|LSR} | |
| | {IN\|OUT\|INOUT} | |
| | {RST} | VSAM only |
| DMSHWM= | (*yyy*,*zz*) | All VSAM |
| FLAGS= | {ALCBUF31}{REUSESHR} | |
| | {BIASDYNM}{BIASDIR}{BIASSEQ} | All VSAM |
| | (IGNHURBA) | All VSAM |
| FORCENSR=OK | | LSR VSAM |
| IGNMAC= | ({<u>SEQ</u> | VSAM only |
| | ,DIR | |
| | ,SKP | |
| | ,DYNAM | |
| | ,RST | |
| | ,LSR}) | |
| NSW= | YES\|NO | VSAM Hiperspaces |
| OVRAMP= | {YES\|<u>NO</u>} | VSAM only |
| OVRMAC= | ([<u>NSR</u>\|LSR] | VSAM only |
| | [,<u>SEQ</u>\|DIR\|SKP\|DYNAM] | |
| | [,<u>NDF</u>\|DFR] | |
| | [,<u>NIS</u>\|SIS] | |
| | [,AMODE31] | |

| | [,NFX|CFX] | Four not normally used |
| | [,NCI|ICI] | |
| | [,NUB|UBF] | |
| | [,DDN|DSN] | |
| SMS= | {<u>NO</u>|YES} | QSAM only |
| SETBUF= | {<u>YES</u>}{NO} | VSAM/QSAM |
| SETBLK= | {<u>NO</u>|YES} | QSAM only |
| SETMAC= | {YES|<u>NO</u>} | VSAM only |
| TRACE= | {YES|NO} | VSAM/QSAM |
| STATS= | (YES|NO, BASIC|EXTENDED) | |
| STATTHLD= | *xxx* | |
| VSI= | {<u>ON</u>|OFF} | |
| | {NODLP,NODMS,NODBM,NODBT} | |

# Description of Parameters for the System Rules Table

### MODE=BYPASS|EXEMPT|SELECT

| Subparameter | Description |
|---|---|
| BYPASS | Defines resources *not* to be optimized for both input and output during BYPASS mode processing. For example, use this command to bypass a particular file that you do not want StarTool IOO to override LSR processing on. |
| EXEMPT | Defines resources *not* to be optimized for output during EXEMPT mode processing. |
| SELECT | Defines resources to be optimized during SELECT mode processing. |

### ACCMETH={QSAM}
####         {BSAM}
####         {VSAM}
####         {EXCP}
####         {BLDVRP}

Defines whether or not the SELECT entry is for QSAM, BSAM, VSAM, EXCP(SORTOUT only) or BLDVRP. QSAM is the default.

If a data set is accessed by both QSAM and BSAM, then code two SELECT entries, one for QSAM and one for BSAM. ACCMETH=EXCP must be coded with DDNAME=SORTOUT on the same select entry. (See for an explanation of BLDVRP).

### JOB=*jobname*

Defines a jobname to be BYPASS(ed), EXEMPT(ed) or SELECT(ed) for processing. Use the keyword ALL to match all jobs.

**STEP=***stepname*

Defines a stepname to be BYPASS(ed), EXEMPT(ed) or SELECT(ed) for processing. The stepname is equal to the lowest level stepname allowing you to bypass one step within a proc, not the entire proc.

> **NOTE**  If no proc exists: stepname = jobstepname.
> If proc exists: stepname = procstepname.

**DDNAME=***ddname*

Defines a DDname to be BYPASS(ed), EXEMPT(ed) or SELECT(ed) for processing.

**DSNAME=***data set name*

Defines a data set name to be BYPASS(ed), EXEMPT(ed) or SELECT(ed) for StarTool IOO processing.

**VOLSER=***volume serial number*

Defines a tape or disk volume serial number to be BYPASS(ed), EXEMPT(ed) or SELECTed for processing.

**PROGRAM=***program name*

Defines the program name coded in the EXEC PGM=*program* JCL statement. StarTool IOO recognizes the program defined in the EXEC JCL statement, and all other programs LINKed to and from this program are automatically included.

You can SELECT, BYPASS or EXEMPT any JOB, STEP, DDNAME, DSNAME, VOLSER, or PROGRAM by the appropriate prefix. To do so, code the following:

```
JPREFIX=jobname prefix

SPREFIX=stepname prefix

DPREFEIX=ddname prefix

PREFIX=data set name prefix

VPREFIX=volume serial number prefix

PPREFIX=program name prefix
```

When using the appropriate prefix, you can add an offset *nn* from the beginning of any JOB, STEP, DDNAME, DSNAME, VOLSER, or PROGRAM. To do so, code:

```
JPREFIX=(jobname prefix,nn)

SPREFIX=(stepname prefix,nn)

DPREFIX=(ddname prefix,nn)

PREFIX=(data set name prefix,nn)

VPREFIX=(volume serial number prefix,nn)

PPREFIX=(program name prefix,nn)
```

### DEVADDR=*xxx*

Defines a tape or disk device address to be BYPASS(ed), EXEMPT(ed) or SELECT(ed) for processing.

*xxx* represents a three-character device address defined to the operating system by your systems programmer during I/O generation, such as 190 or 39C.

### BLKSIZE=*maximum block size*                                    QSAM only

Defines an override block size to be used when optimizing this data set. Override blocksizes ignore the target blocksize on the device table and are not normally coded. Code a compatible blocksize for the data set residing on a specific device type. The maximum value that can be coded is 32760.

Code this option only for items like DDNAMES for SYSGO since the linkage editor has an artificially small blocksize limitation.

### BUFNI=(*number index buffers*,CKSTRNO)                          VSAM only

BUFNI allocates the number of I/O index buffers placed on the VSAM data set when it is opened and placed in NSR.

CKSTRNO is an optional parameter. If specified, StarTool IOO sets the number of index buffers to the value specified in STRNO + 1 when processing SEQUENTIALLY. When processing DIRECT, StarTool IOO sets the number of index buffers to the number index buffers.

If CKSTRNO is not specified, the number of index buffers is set to the number index buffers specified for both SEQUENTIAL and DIRECT.

### BUFND=(*number data buffers*,CKSTRNO)                           VSAM only

BUFND specifies the number of I/O data buffers VSAM uses for transmitting data between virtual and auxiliary storage for KSDSs, ESDSs, and RRDSs.

CKSTRNO is an optional parameter. If specified, StarTool IOO sets the number of index buffers to the value specified in number data buffers when processing SEQUENTIALLY. When processing DIRECT, StarTool IOO sets the number of index buffers to the STRNO + 1.

If CKSTRNO is not specified, the number of index buffers is set to the number index buffers specified for both SEQUENTIAL and DIRECT.

### BUFFER={(CKSTRNO,*xxxxx*)}                                      VSAM only
### or BUFFER={(SPACE,*xxxxx*)}

BUFFER allocates the total amount of buffer space to be used when calculating data and index buffers for VSAM. If you specify less space than specified in the BUFFERSPACE parameter of the DEFINE command when the data set was defined or what was specified in BUFSP within the Access Control Block (ACB) in the program, the larger value is used.

If *xxxxx* is greater than BUFNI plus BUFND, then remaining buffers are allocated differently depending on the keyword that you specify.

If you specify **CKSTRNO**:

- For MACRF=DIR processing, StarTool IOO makes the number of data buffers equal to STRNO + 1 and then allocates the exact *xxxxx* amount to use for index CI buffers.

- For MACRF=SEQ processing, StarTool IOO makes the number of index buffers equal to STRNO + 1 and then allocates the remaining *xxxxx* to use for data CI buffers.

If you specify **SPACE**:

- If the *xxxxxx* amount is greater than the amount called for by BUFNI and BUFND, the extra space is allocated as follows: For MACRF=DIR processing, only index buffers are allocated; for MACRF=SEQ processing, one additional index buffer and as many data buffers as possible are allocated.

- If the *xxxxxx* amount is less than the amount called for by BUFNI and BUFND, the number of data and index buffers are decreased as follows:

| | |
|---|---|
| MACRF=DIR | Number of data buffers are decreased to the amount specified but not fewer than the minimum number required. If more space remains to be decreased, the number of index buffers is decreased until the amount called for by the combined BUFNI and BUFND complies with the BUFSP amount. |
| MACRF=SEQ | Number of index buffers is decreased to not fewer than 1 more than the minimum required. If more space remains to be decreased, the number of data buffers is decreased but never decreased more than the minimum number required to process. |

### BUFFER={(NUMBER,*xx*)},{(TRACKS,*xx*)},            QSAM only
### or BUFFER={(SPACE,*xxxxxx*)}

In QSAM, xx defines the number of buffers, tracks, or the amount of space to be used when optimizing a data set that matches the search criteria. This value overrides the device table value if that was coded. Normally, the device table should not be used. Manipulate buffers using this parameter in this table and do not use the device table space values.

**BLDVRP=(AUTO[,SEP])**                                                **VSAM only**
        **,STRNO=***number*
        **,ISIZEMAX= {***#chars***}**
                      **{(***#***KB** or **K[,***Hiperspace***])}**
        **,DSIZEMAX={***#chars***}**
                      **{(***#***KB** or **K[,***Hiperspace***])}**

| Subparameter | Description |
|---|---|
| AUTO | Specify only for Local Shared Resources (LSR). It invokes the BLDVRP (BuiLD VSAM Resource Pool) macro based on the information in the catalog (Index CIsize, Data CIsize, key length) and the overridden ACB. StarTool IOO builds a buffer pool large enough to hold the entire number of index and data CIs that the file contains, up to the ISIZEMAX= and DSIZEMAX= maximum limit. SEP tells StarTool IOO to issue the BLDVRP with separate TYPE=DATA and TYPE=INDEX parameters which creates a separate resource pool for each component. If SEP is not specified, both data and index components use the same data pool. |
| ISIZEMAX= DSIZEMAX= | Provides an upper-limit for these index/data buffer pool size (in characters or KB). StarTool IOO finds the high-used data RBA, calculates the number of index CIs needed to hold the entire index in memory, and uses this number (if it does not exceed ISIZEMAX). StarTool IOO allocates data buffers to hold all the data in memory that it can fit, up to a maximum amount of the DSIZEMAX value or the entire file (whichever occurs first). The combined total of ISIZEMAX and DSIZEMAX cannot be greater than the MAXSPC value in the Global table. |
| *Hiperspace* | Provides an upper-limit for allocating the amount of Hiperspace buffers. StarTool IOO determines how many (if any) actual Hiperspace buffers you can realistically use and allocates them. Use KB, K, MB, M for kilobytes and megabytes, respectively. |

**BLDVRP=**
        **,BUFFERS=(***size***(***number***[***Hiperspace***]),...)**
        **,FIX={BFR|IOB|(BFR,IOB)}**
        **,KEYLEN=***length*
        **,RMODE31={ALL|BUFF|CB|NONE}**
        **,SHRPOOL={***number***|0}**
        **,STRNO=***number*
        **[BLDVRP=SETSHR]**

Use this command if you want to manually force the use of specific resource pools. This command manually issues BLDVRPs; the following parameters can only be specified for Local Shared Resources (LSR). StarTool IOO issues this manual BLDVRP (build VSAM resource pool) macro based on the information that you code in this macro. It is your responsibility to use this macro correctly. StarTool IOO does not figure out what is optimum; it uses whatever you code and simply issues the BLDVRP based on your parameters.

BLDVRP=AUTO is the default, so you must code BLDVRP=, to bypass automatic optimization.

If you want StarTool IOO to determine what is optimal, do not use this manual BLDVRP, instead use BLDVRP=AUTO.

| Subparameter | Description |
|---|---|
| BUFFERS= (*size*(*number*[*Hiperspace*]), | The size and number of buffers in each buffer pool in the resource pool. *Hiperspace* is the number of Hiperspace buffers in the pool. StarTool IOO assumes that you understand how to use a manual BLDVRP and does not perform validity checking. When processing a KSDS, the index and data components share the buffers of a buffer pool. When an alternate index is used to process against a base cluster, the alternate index and the base cluster share buffers. Each component uses the buffer pool whose buffers are exactly the right size or the next larger size. Additionally, the combined total of all sizes and numbers cannot be greater than the MAXSPC value in the Global table. These are possible values for *size, number,* and *Hiperspace:* |
| | *size*  512, 1024, 2048, 4096, and increments of 4096 to max of 32k. |
| | *number*  3 or more. |
| | *Hiperspace*  The default is zero and the maximum is 16,777,215. |
| FIX={BFR\|IOB\|BFR,IOB} | Either I/O buffers or I/O control blocks or both are fixed in real storage. |
| KEYLEN=*length* | The maximum key length of any of the data sets that are going to share this resource pool. Normal KSDSs and alternate indexes must be provided for. Longest prime key of KSDSs; or zero if no KSDSs. |
| LOC={BELOW\|ANY} | Either the LSR pool is below 16 megabytes or the LSR pool is above 16 megabytes. *any* implies the application program running in the 31-bit addressing mode. LOC and RMODE31 are mutually exclusive parameters. |

| Subparameter | Description |
|---|---|
| RMODE31={ALL|BUFF|CB|NONE} | SP2/3/4 only. LOC and RMODE31 are mutually exclusive parameters. |
| | **ALL**     Both buffers and control blocks associated with the LSR pool are above the 16 megabyte line. |
| | **BUFF**    Only buffers above the 16 megabyte line. |
| | **CB**      Only VSAM I/O control blocks associated with the LSR pool are above the 16 megabyte line. |
| | **NONE**  Both buffers and control blocks are below the 16 megabyte line. |
| SHRPOOL=({*number*|0},) | LSR only. |
| | Specify the ID *number* (0 to 255) of the LSR pool that you want to create or it defaults to zero. |
| | If multiple files are shared in one LSR pool for a particular jobstep, the system table must specify a BUFFERS= parameter for the first file opened. Within this BUFFERS statement, all needed buffers for all files to be shared must be specified. All other files that share the subpool should only specify the same SHRPOOL number, a BLDVRP=SETSHR statement, and not specify a BUFFERS parameter. Use only a fixed pool number because the file that was opened first specified the correct size and number of buffers for all of these files. |
| | If you have multiple VSAM files used by a program and you want to manually issue the BLDVRP but you want each file to reside in its own SHRPOOL (for example, using Deferred writes), you must increment the sharepool number for all VSAM files accessed. |
| | For example: you can place the first file opened in SHRPOOL 1, the next file in SHRPOOL 2, up to SHRPOOL 255. VSAM prohibits any more LSR sharing even if you mis-specify it. you can avoid this by issuing BLDVRP=AUTO. |

| Subparameter | Description |
|---|---|
| STRNO=*number* | LSR only.<br><br>A number from 1 to 255 that equals the maximum number of requests that can be issued concurrently for all data sets that share the resource pool. For single files within a sharepool, this number is normally 1 since concurrent processing is unusual in batch. We recommend a minimum of 3.<br><br>To use one sharepool number for multiple files in batch, do *not* use BLDVRP=AUTO. On the SELECT statement where you manually specify the *size* and number of buffers for the StarTool IOO generated BLDVRP, STRNO= should be the sum total of STRNOs in the ACBs of all VSAM files that share resources in one region for LSR.<br><br>**NOTES** Multiple simultaneous requests require special programming skills; for example, CICS uses multiple strings and prevents deadlocks through CICS enqueues. If sharing of resources occurs by using multiple string numbers for a single file, then an exposure exists with LSR unless the programmer compensates by enqueuing at the CI level or knowing that multiple positions are requested, and they do not overlap within any given CI. With TRACE=ON, you can spot these situations.<br><br>Another problem exists for programs that mis-specify the number of concurrent processing strings (even if they are programmed to handle deadlocks). NSR performs dynamic string addition to provide more buffers and control blocks to those programs that have an ACB with an insufficient STRNO value. LSR does *not* support dynamic string addition so StarTool IOO compensates by specifying more strings than the minimum you need. The system reports any batch ACB with multiple strings (STRNO>1) for a single file placed in LSR pools for you to check deadlock potential. Deadlocks occur when a CI is under exclusive control of a string #1, and string #2 requests a logical record in the same CI. String #2 enters a WAIT causing the step to wait on itself -- Deadlocked. |

| Subparameter | Description |
|---|---|
| [BLDVRP=SETSHR] | To share multiple files in one LSR pool for a particular Jobstep, the system table must specify a BUFFERS= parameter for the file that is opened first. Within this BUFFERS statement, all buffers that you need to specify all of the files to be shared. |
|  | All other files with which you want to share the subpool should specify only the same SHRPOOL number, a BLDVRP=SETSHR statement and not specify a BUFFERS parameter. Therefore, BLDVRP=SETSHR is a required parameter for the second through final file SELECTS that use a previously built sharepool. |
|  | **Example** |
|  | You want three files to share an LSR pool (assume that the first and second files have twice as many CIs as the third file). The first and second file has an Index CIsize of 512 and Data CIsize of 4096. The third file has an Index CIsize of 1024 and Data CIsize of 8192. For the first file opened, a manual BLDVRP is issued: |
|  | <pre>ULTITBL  MODE=SELECT,<br>    ACCMETH=VSAM,DSNAME=TST.PAY.A,<br>    CKMAC=(DIR,NSR),<br>    OVRMAC=(LSR,DFR),<br><br>      BUFFERS=(512(400),4096(400),1024(<br>      100),8192(100)),<br>    RMODE31=BUF,<br>    SHRPOOL=2,<br>    STRNO=5<br>ULTITBL  MODE=SELECT,<br>    ACCMETH=VSAM,DSNAME=TST.PAY.B,<br>    CKMAC=(DIR,NSR),<br>    OVRMAC=(LSR,DFR),<br>    SHRPOOL=2,<br>    BLDVRP=SETSHR<br>ULTITBL  MODE=SELECT,<br>    ACCMETH=VSAM,DSNAME=TST.PAY.C,<br>    CKMAC=(DIR,NSR),<br>    OVRMAC=(LSR,DFR),<br>    SHRPOOL=2,<br>    BLDVRP=SETSHR</pre> |

**CHKMAC={SEQ,DIR,SKP,DYNAM},                                    VSAM only**
**     {IN,OUT,INOUT}**
**     ,{RST}**
**     ,{NSR,LSR}**

Defines whether or not this SELECT is to operate on VSAM data sets opened for Sequential, Direct, Skip Sequential, or Dynamic access; opened for Input, Output (COBOL uses RST for output) or I/O using Non Shared Resources or Local Shared Resources. Any combination of the above options can be coded.

**DMSHWM=(***yyy***,***zz***)**                                          **All VSAM**

Defines the threshold (High Water Mark) for DMS (Dynamic Mode Selection). If you define these values, StarTool IOO monitors the I/O into LSR and if the amount of SEQUENTIAL accesses exceeds the threshold values, StarTool IOO changes modes.

DMS uses the smaller matched value of *yyy* or *zz* based upon DMS=(YES,MIN) in Global Table. DMS uses the largest matched value of the two numbers, *yyy* and *zz*, if DMS=(YES,MAX) is specified in Global.

Also, if *yyy* and *zz* are both zero(0), it is the same as having DMS=NO in the Global Table except it applies only to this entry.

| Subparameter | Description |
|---|---|
| *yyy* | Count of CIs that you want to consecutively process sequentially using LSR before it dynamically changes modes; *yyy* may be from 0-255. |
| *zz* | Percentage of allocated buffers to process sequentially using LSR before it dynamically changes modes; *zz* may be any percentage between 0 to 99%. |

**IGNMAC= {SEQ,DIR,SKP,DYNAM,RST},**                         **VSAM only**

Defines whether or not this Select is to ignore operation on VSAM data sets opened for Sequential, Direct, Skip Sequential, or Dynamic access, opened with RST. Any combination of the above options can be coded.

**FLAGS={ALCBUF31}{BIASDYNM}**                                                          **All VSAM**
       **{BIASDIR}{BIASSEQ}{IGNURBA}**

| Subparameter | Description |
|---|---|
| ALCBUF31 | StarTool IOO can use dynamic buffer management (DBM) to detect when to put ALC programs' buffers below the 16M line. This operand allows you to override DBM for this entry. |
|  | Assembler programs can use VSAMs control block structure to access data buffers instead of going through the VSAM access method, and a user may need access to some of the accounting data saved in the placeholder(PLH). If StarTool IOO put buffers and control blocks above the line, this program could abend. Therefore, DBM dynamically places ALC program-buffers and control blocks below the line. |
| BIASDYNM | StarTool IOO uses a biasing technique to allocate buffers on dynamic opens. |
| BIASDIR | StarTool IOO uses a biasing technique to allocate buffers on direct opens. |
| BIASSEQ | StarTool IOO uses a biasing technique to allocate buffers on sequential opens. |
| IGNURBA | For NSR optimization, StarTool IOO ignores the High Used RBA for calculating buffer allocation. (Not normally used.) |

**NSW=YES|NO**                                                          **VSAM Hiperspaces**

Make the task that is accessing this file Non-Swappable since it is using Hiperspaces and Hiperspaces only exist while you are swapped in. Use this parameter if you want to use Hiperspaces on batch jobs.

**OVRMAC=([NSR|LSR]**                                                          **VSAM only**
       **[,DIR|SEQ|DYNAM|SKP]**
       **[,NIS}{SIS]**
       **[,NDF}{DFR]**
       **[,AMODE31]**
       **[,NFX|CFX]**
       **[,NCI|ICI]**
       **[,NUB|UBF]**
       **[,DDN|DSN])**

Use these parameters to override any MACRF parameters that are specified in the program. In COBOL, MACRF parameters are usually impossible to override.

Use caution because it is possible to specify illogical combinations across groups and some options require program modifications. All of these MACRF parameters work differently from the normal StarTool IOO parameters in that if you do not specify the parameter, StarTool IOO does not default to anything and does not override anything with this default. The items underlined are VSAM/AMS defaults, not defaults of StarTool IOO; therefore, they override values in the ACB only if they are specified in the StarTool IOO system table. Each item in a group that is specified in the StarTool IOO table turns off any other item in that group in the ACB.

The following items are grouped in the order that you would most likely override the ACB (1=most likely to use; 8=least likely to use):

**Group 1**

| | |
|---|---|
| NIS | Normal insert strategy for Direct processing. Divides the CI into two parts always at the midpoint for CI splits. |
| SIS | Sequential insert strategy efficiently inserts contiguous keys while processing directly. Divides the CI up at the record you are trying to insert (SIS is always used for sequential processing; therefore, specifying SIS allows direct inserts to be processed identically to sequential processing). |

**Group 2**

| | |
|---|---|
| NSR | Non-shared resources. |
| LSR | Local shared resources. Generally used in conjunction with Deferred Writes, Group 3. (Should also code BLDVRP=AUTO). |

**Group 3**

| | |
|---|---|
| NDF | Writes are not deferred for direct PUTs. |
| DFR | Direct PUT requests are deferred until the buffer is needed. Implies sharing resources (LSR). Should also code BLDVRP if the program was not coded to build its own VSAM resource pool; For example, COBOL programs). |

**Group 4**

| | |
|---|---|
| AMODE31 | Running with NSR and you want buffers allocated above the 16 MB line. |

**Groups 5 - 8**

NFX|CFX, NCI|ICI, NUB|UBF and DDN|DSN MACRFs are supported.

**OVRAMP={YES|NO}**                                                      **VSAM only**

This statement causes StarTool IOO to override any AMP parameters in the JCL as well as any program ACB or catalog entries.

**SETBLK={YES|<u>NO</u>}**                                               **QSAM ONLY**

SETBLK=YES defines that the SELECT entry invoke actual blocksize optimization action (blocksize changes). NO is the default if this parameter is omitted. NO says that the SELECT entry does not optimize but it will produce StarTool IOO Notification message to the job log, system log, and console (as if it did optimize).

**SETBUF={<u>YES</u>|NO}**                                               **VSAM / QSAM**

SETBUF=YES must be specified in the system table SELECT statements as well as the global table to enable StarTool IOO rebuffering and Start I/O optimization function. If SETBUF=NO is specified or the parameter is omitted, the number of buffers is determined by your installation's BUFNO default.

### SETMAC={YES|<u>NO</u>}                                        VSAM only

Determines whether or not StarTool IOO is to override any MACRF parameters in the ACB.

### STATS=(YES|NO,
####         BASIC|EXTENDED)

STATS turns StarTool IOO statistic feature on and off. STATS also specifies whether or not you collect BASIC statistics or EXTENDED statistics. If you specify STATS=(YES), then you will get YES,BASIC. To turn the statistics feature off, specify STATS=(NO). The STATS specification in the System Rules Table supersedes the STATS specification made in the Global Table.

> **NOTE**  If extended statistics are turned on, more CPU cycles are needed for each I/O.

### STATTHLD=*xxx*

*xxx* defines threshold that must be met or exceeded before statistics are printed. This keyword allows you to limit the amount of statistics from jobs that do little or no I/O. *xxx* can be a number from 0 to 32K. If you specify STATTHLD=0, then each time a data set is closed, you will see the statistics. Use in conjunction with the parameter STATS.

### VSI=(ON|OFF)                                              All VSAM
####     (NODLP,NODMS,NODBM,NODBT)

VSI is the I/O interface function that provides services at I/O time. This parameter allows you to specify exactly what portion of these functions you want.

| Function | Description |
|---|---|
| DLP (Dynamic LSR Positioning) | Performs positioning for NSR coded programs that use LSR. |
| DMS (Dynamic Mode Selection) | Changes from LSR to NSR if a lot of sequential processing is performed. |
| DBM (Dynamic Buffer Management) | Detects if BAL program issuing I/O and other factors and forces the program to use below the 16 Megabyte line buffers. |
| DBT (Dynamic Buffer Translation) | Performs buffer translation for above the line buffers so programs not coded in 31 bit mode can use above the 16 Megabyte line buffers. |

The functions that VSI contains are:

| Function | Description |
|---|---|
| **ON** | Turns on the capability for all of these to be active. |
| **OFF** | Turns off all of these functions simultaneously. |
| **NOD***xx* | Turns off each function individually. Multiple NOD*xx* statements can be specified on the same VSI statement. For example, VSI=(NODLP,NODMS,NODBM) leaves Dynamic Buffer Translation operating but shuts off all other functions of VSI. |

# Rules Example

The first entry selects the BLDVRP override for a job with the stepname of ABC by allowing StarTool IOO to get control when step ABC is going to issue its BLDVRP.

The second entry selects any files in step ABC that already use LSR (DIR or DYNAM access) and overrides them to put them in a separate sharepool that is optimized for each file component.

```
ULTITBL MODE=SELECT,ACCMETH=BLDVRP,SPREFIX=ABC

ULTITBL MODE=SELECT,                 SELECT THESE FILES:
   SPREFIX=ABC,                       STEPNAME PREFIX
   SETBUF=YES,                        CHANGE THE BUFFERS
   SETMAC=YES,                        CHANGE THE MACRF
   CHKMAC=(DIR,DYNAM,LSR),            VSAM MACRF SELECTED
   IGNMAC=(SEQ,SKP)                   MACRFs TO IGNORE
   OVRMAC=(DFR),                      CHANGE TO THIS MACRF
   OVRAMP=YES,                        OVERRIDE THE AMP JCL PARAMETER
   BLDVRP=(AUTO,SEP),                 FIGURE OUT # INDEXES/DATA BUFRS
   STRNO=8,                           MIN # OF STRINGS FOR BUFFER POOL
   RMODE31=ALL,                       BUF + CONTROL BLOCKS ABOVE 16M
   DSIZEMX=(256K,2M),                 MAX DATA CI AREA AUTO WILL USE
   ISIZEMX=(256K,500K),               MAX INDEX CI AREA AUTO WILL USE
   TYPE=LSR,                          USE LSR VERSUS NSR
   ACCMETH=VSAM                       SELECT ONLY VSAM FILES
```

# Copy-Code Members

The following members contain common copy code used to construct all System Rules. The copy-code members are in the STRIOO.V*v*R*r*M*m.*MAC distribution data set.

> **NOTE** Each Rules Table distributed with StarTool IOO has the #BYPASS copy-code member included. When updating a Rules Table, it is good practice to add any updates to the bottom of the current table. Thus, when the Rules Table is reassembled, the #BYPASS member is included again.

| Copy-Code Member | Description |
|---|---|
| #BYPASS = | Normal Bypasses |
| #EXEMPT = | All Exempts (Blocking Only) |
| #GNRMSG = | (02&03)GNR BLKNG MSGS (No Actual Reblocking) |
| #LSRESA = | Normal LSR-to-LSR Rules |
| #QSAMBLK = | QSAM Reblocking and Rebuffering Rules |
| #QSAMBUF = | QSAM Rebuffering Rules |
| #SRT#GNR = | Reblocking for Sort and IEBGENER Rules |

| Copy-Code Member | Description |
|---|---|
| #VSAME31 = | Normal NSR-to-NSR and NSR-to-LSR Rules |
| #VSAMESA = | Dual Buffering NSR-to-NSR and NSR-to-LSR Rules |

# Index

# Q

QSAM
    BUFFER 98

# R

README 5
REGNHWM 84
Replace File pop-up panel(DCFREP) 38
requesting optimization 19
RMODE31 99
RPG 84
RST 94
RULE criteria record 51
rule definition records 47
RULES
    LSR 108
    system 100, 106
rules
    assembly 80, 92
    coding guidelines 47
    Device 90
    device 90, 91
    forcing the use of 22
    Global 80, 81, 82, 83, 84
    optimization 22
    System 92, 95
    system 82, 91, 93, 95, 96, 100
rules table
    parameters 95
rules tables 80
    device 80
    global 80
    system 80

# S

S IOO operator command 10
S IOODCF operator command 13
S IOOINIT operator command 10
S IOOMODS operator command 12
scope records 49
select jobs
    by prefix 96
SELECT scope record 49
SEQ 94
sequential insert strategy 106
Set Panel Attributes 28
SETBLK 84, 106
SETBUF 84, 106
SETMAC 85
SHRPOOL 101
SIS 94

SKP 94
SMFREC 85
SORTIN 85
SPACE keyword 98
SPREFIX 96
starting IOO
    persistent address space 10
    transient address space 10
start-up options
    K0 through K5 10
statistic reports 71
statistics
    activating 72
    basic 72, 107
    control interval 75
    extended 76, 107
    LSR 73
    NSR 76
    output 71
STATS parameter 72, 107
STC override record 61
STEP criteria record 50
step name prefix 96
stepname 96
stopping IOO
    persistent address space 10
STRNO 99, 102
SYSGO 97
system rules table 80, 91
SYSUFX 80

# T

TRACE
    activate 86
    deactivate 86
TRACE override record 62
TSOMSG 86
tuning buffers
    automatic 99
    manual 99

# U

UBF 95
Ulti-Block rule 49
ULTIDVT macro
    format 91
Ulti-LSR rule 49
Ulti-SAM rule 49
Ulti-VSAM rule 49
Ulti-VSAM/SAM/BLOCK RULES 80

# V

VOLSER 96
volume serial number 96
VPREFIX 96
VSAM 7
    BUFFER 97
    BUFFER= 97
    BUFND 97
    BUFNI 97
    CHKMAC 103, 104
    Hiperspace 99
VSAM override records 55
VSI override record (VSAM) 60
VSMLSR24 optimization rule 51
VSMLSR31 optimization rule 51
VSMNSR24 optimization rule 51
VSMNSR31 optimization rule 51

# W

wildcard character 51, 52
writes not deferred 106