

- Allocate a buffer prefix extension for caching subpools. The extension resides as a non-contiguous area and is anchored from the prefix. It consists of two parts:
 - A fixed-length section for asynchronous processing.
 - A variable length section for data transfer buffer lists. The number of buffer lists is dependent upon the subpool buffer size.

Example: A subpool buffer size of 6 KB requires three buffer lists and a subpool size of 0.5 KB, or 1 KB requires one buffer list. A single buffer list accommodates 2 KB of subpool buffer.

- Establish miscellaneous values for coupling facility data transfer parameters including:
 - Element and directory ratios
 - Buffer increment number (number of 256 byte segments)
 - Number of buffers in the buffer list.

The write data option is supported at the subpool level and permits the definition of each subpool with a write cache option.

Restriction: When using sequential buffering and the coupling facility for OSAM data caching, the OSAM database block size must be defined in multiples of 256 bytes (decimal). Failure to define the block size accordingly can result in ABENDS0DB PIC15 from the coupling facility. This condition exists even if the IMS system using sequential buffering is accessing the database in read-only mode.

Enabling the long busy handling function

RAMAC disk arrays sometimes cause a problem to online programs when the DASD subsystem takes a very long time to do error recovery.

This condition (known as *long busy*) might range in time from a couple of seconds to many minutes. You can avoid these excessive long wait times by defining the Multiple Area Data Sets I/O Timing (MADSIOT) keyword on the DFSVSMxx member of the IMS PROCLIB data set, thus enabling IMS system's long busy handling function.

For more information about the long busy function, see Managing I/O errors and long wait times (Database Administration) in *IMS Version 14 Database Administration*.

►—MADSIOT=(*structurename*,*iotime*)—►

structurename

Specifies the name of the coupling facility list structure that IMS uses for the long busy handling function.

iotime

Specifies the maximum time (in seconds) before the long busy handling function is activated. The allowable values are 0 to 255. This value is passed to the I/O subsystem for all I/O requests to MADS. If the time interval specified expires before the request completes, the request is terminated. This explicit request timing value takes precedence over an I/O timing value specified at the volume level.

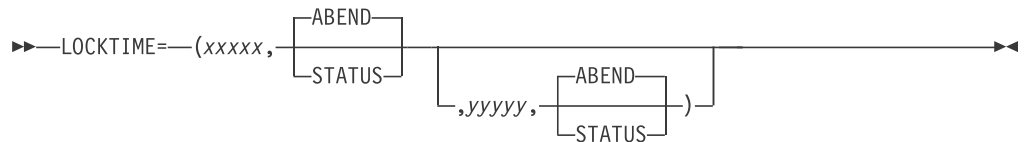
Enabling the IRLM lock timeout function

The IRLM Lock Timeout function enables you to interrupt processes that are waiting for locks for longer than a specified number of seconds.

To use this function, specify a positive decimal integer value from 1 to 32767 for the LOCKTIME parameter in the DFSVSMxx member of the IMS PROCLIB data set (or DFSVSAMP DD statement for IMS batch procedures). The LOCKTIME parameter specifies the number of seconds that IRLM waits before rejecting lock requests that have not been granted. For example, when LOCKTIME=10 is specified, IRLM waits ten seconds before rejecting lock request that have not been granted.

When no value is specified for the LOCKTIME parameter, IRLM issues message DXR162I to indicate that a task has held (or has been waiting for) a lock. The first message is issued after five minutes; thereafter, messages are issued at intervals of one minute. RMF records (type 79.15) can be formatted for more information about the task that is holding (or waiting for) the lock.

When a value is specified for the LOCKTIME parameter, IRLM interrupts all the dependent regions that have been waiting for a lock for longer than the number of seconds specified by the LOCKTIME parameter. An ABENDU3310 is issued for every dependent region that waits for longer than the number of seconds specified by the LOCKTIME parameter. The number of seconds can be changed after IMS initialization by issuing the command `MODIFY,irlmproc,SET,TIMEOUT=nn`.



ABEND

Specifies that an abend occurs when the value specified for LOCKTIME is exceeded. ABEND is the default.

STATUS

Specifies that a status code of BD be returned when the value specified for LOCKTIME is exceeded.

xxxxx When LOCKTIME=(xxxxx), xxxxx specifies the online timeout value in seconds. Valid values are from 1 to 32767. Optionally, you can specify ABEND or STATUS after xxxxx to determine whether an abend occurs or a status code is issued when the xxxxx value is reached. If a second timeout value is not specified, a value of xxxxx is also used for batch or a BMP. Unless xxxxx is specified as a multiple of the local IRLM deadlock timer value, the timeout occurs at the next deadlock timer interval.

yyyyyy If specified in the DFSVSMxx PROCLIB member after xxxxx, defines the timeout value, in seconds, for BMPs or JBPs. Valid values are from 1 to 32767. Optionally, you can specify ABEND or STATUS after yyyyyy to determine whether an abend occurs or a status code is issued when the yyyyyy value is reached.

If specified in the DFSVSAMP PROCLIB member after xxxxx, defines the timeout value, in seconds, for batch. Valid values are from 1 to 32767. Optionally, you can specify ABEND or STATUS after yyyyyy to determine whether an abend occurs or a status code is issued when the yyyyyy value is reached.

Unless xxxxx and yyyyyy are specified as a multiple of the local IRLM deadlock timer value, the timeout occurs at the next deadlock timer interval. For example, if

xxxxx is 5 seconds and yyyyyy is not specified, and the deadlock timer value is (3,1), the timeout occurs at 6 seconds because that is a multiple of 3.

Preventing transactions from being terminated when HALDB partitions are unavailable

Use the PPUR= control statement on the DFSVSMxx member to prevent transactions with a processing option of PROCOPT=GOx from being terminated if a HALDB partition is unavailable due to a database command in progress. This condition would cause either STATUSGG or STATUSBA to be issued.

►► PPUR=

N
Y

 ◀◀

- N Do not prevent STATUSGG or STATUSBA from being issued if a HALDB partition is unavailable because a database command is in progress.
- Y If a HALDB partition is being accessed by a transaction with a processing option of PROCOPT=GOx, and the partition is unavailable due to a database command in progress, terminate the transaction and reschedule it.

Preventing DBRC calls for HALDB version verification

Use the PSELNODBRC control statement in the DFSVSMxx member of the IMS PROCLIB data set to prevent the Database Recovery Control facility (DBRC) from being called when partition selection (either with key range selection or with a partition selection exit) finds that a key requested by an application call is not in the range of any partitions.

Calling DBRC in this case can cause performance degradation especially when many similar application calls are made or when many other applications or utilities are using the RECON data set.


►► PSELNODBRC ◀◀

PSELNODBRC

Specify this keyword if you want to prevent DBRC from being called when partition selection (either with key range selection or with a partition selection exit) finds that a key requested by an application call is not in the range of any partitions.

Otherwise, omit this keyword.

Related concepts:

 Overview of DBRC (System Administration)

Resuming an online reorganization for HALDBs during IMS warm or emergency restart

To resume the online reorganization of a high-availability large database (HALDB) during an IMS warm or emergency restart, use the FFROLR= control statement in the DFSVSMxx member of the IMS PROCLIB data set.

►► FFROLR=

E
D

 ◀◀