IMS
DB2 for z/OS

IRLM Messages and Codes for IMS and DB2 for z/OS

IBM
Contents

About this information ............................................................... v
Chapter 1. DXR messages ............................................................ 1
Chapter 2. IRLM return and reason codes. ....................................... 23
Chapter 3. IRLM abend codes ......................................................... 35
Notices ......................................................................................... 41

© Copyright IBM Corp. 1974, 2013
About this information

This publication describes messages and codes that are issued by the internal resource lock manager (IRLM) of IMS™ or DB2® for z/OS®. Unless otherwise specified, information applies equally to the following products:

- IMS Version 13
- IMS Version 12
- IMS Version 11
- DB2 11 for z/OS
- DB2 10 for z/OS
- DB2 Version 9.1 for z/OS

Where differences exist by product or version, the differences are identified by text, an image, or both.

In this information, the term “DB2 for z/OS” applies to DB2 11 for z/OS, DB2 10 for z/OS, and DB2 Version 9.1 for z/OS.

The information in this publication was previously published in the messages and codes publications for the individual products.
Chapter 1. DXR messages

This section describes the messages issued by the internal resource lock manager (IRLM). The database management system (DBMS) referred to throughout the messages is either DB2 or IMS. Some messages do not apply to DB2 operations.

The messages have the form of DXRxxxa and are listed in numeric order.

**DXR** The prefix for IRLM messages.

**xxx** The numeric part of the message.

**a** An alphabetic suffix, as follows:

- **I** The message is for information only.
- **E** The message reports an error condition which might require action.
- **A** Action is required of the user before processing can continue.

IRLM will now direct messages in two ways:

1. If a UN-solicited message is to be issued, the target "console" will be the one in which IRLM was started from.
2. If a solicited message is to be issued, i.e. via "F IRLM,STATUS" opercmd, it will automatically be directed back to the console from which the modify was accepted. Even if it was a TSO session running SDSF.

All messages contain the issuing IRLM subsystem name and ID irlmx, where irlmx is the concatenation of the IRLMNM and IRLMID specified as EXEC parameters when the IRLM was started, as shown below.

The following table shows the concatenation of IRLMNM and IRLMID into irlmx:

*Table 1. Format of irlmx*

<table>
<thead>
<tr>
<th>irlmx</th>
</tr>
</thead>
<tbody>
<tr>
<td>IRLMNM</td>
</tr>
<tr>
<td>IRLMID</td>
</tr>
<tr>
<td>4 bytes</td>
</tr>
<tr>
<td>1 to 3 bytes</td>
</tr>
</tbody>
</table>

- If IRLMNM='KRLM' and IRLMID='12' then irlmx='KRLM012'
- If IRLMNM='LRLM' and IRLMID='0' then irlmx='LRLM000'

The IRLMID is displayed as three characters in the messages.

<table>
<thead>
<tr>
<th>DXR100I irlmx</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>STOR STATS</strong></td>
</tr>
<tr>
<td><strong>PC:</strong> <strong>text</strong></td>
</tr>
<tr>
<td>The parameter as specified in the irlmproc. Its value is YES. Monitor the PVT and AHWM value when this is YES.</td>
</tr>
</tbody>
</table>

| **LTEW:** **text** |
| The lock table entry width, which is the number of bytes needed for each lock table entry in the lock structure. The more users in the group, the more bytes are needed to manage each lock table entry. Expected values are: |
| - 2 (for up to six users) |
DXR100I

- 4 (for up to 22 users)
- 8 (for up to 32 users)
- N/A (if the IRLM is not connected to a group)

LTE: num M
The number of lock table entries in units of 1,048,576 that were available in the coupling facility the last time this IRLM was connected to the group. If the value is less than one unit, the value is zero.

You can set this value initially with the LTE parameter in the IRLMPROC. Use the IRLM modify command `F irlmproc,SET,LTE=` to change this value.

This field is valid only for SCOPE=GLOBAL or NODISCON.

RLE: num
The number of record list entries available in the coupling facility the last time this IRLM was connected to the group.

This field is valid only for SCOPE=GLOBAL or NODISCON.

RLEUSE: num
The number of record list entries that were in use in the coupling facility at the time the MODIFY command is issued. If the IRLM is disconnected from the CF, this number represents those in use when last updated prior to DISCONNECT.

This field is valid only for SCOPE=GLOBAL or NODISCON.

BB PVT: num M
The extended private (below the bar) storage region limit threshold that is monitored for control blocks. When this limit is reached, new lock requests receive reason code 08 out-of-storage reason code unless they are must-complete requests. The default value is determined from the size of the extended private storage value minus 10% for a non-lock buffer to be used for IRLM and system required storage.

Use the IRLM MODIFY command `MODIFY irlmproc,SET,PVT=` to change this value.

AB PVT (MEMLIMIT): num P
Displays the current limit to private storage above the bar, also known as the MEMLIMIT, that is managed by MVS™. This storage is used for locks. IRLM monitors the use of above-the-bar storage, and if it exceeds ninety percent of MEMLIMIT, then non-must-complete requests fail with reason code 08, the out-of-storage reason code.

Use the IRLM MODIFY command `MODIFY irlmproc,SET,MLT=` to change this value.

CSA USE:
Displays information about the Common Service Area.

ACNT: num K
Is the current CSA + ECSA usage for lock control structures. This storage is called “accountable” because it is accountable against the MAXCSA value. Its value is specified as either k for kilobytes or M for megabytes. When PC=YES, this value will be 0K.

AHWM: num K
Is the high water mark or greatest amount of CSA + ECSA allocated by IRLM during this initialization period for lock control structures. Its value is specified as either k for kilobytes or M for megabytes.

CUR: num K
Is the current CSA + ECSA usage. Its value is specified as either k for kilobytes or M for megabytes. This value accounts for CSA + ECSA storage obtained in IRLM. IRLM often gets storage for locks under an application’s ASID and manages this storage regardless of the status of the owning ASID. The display of cur storage accounts for all of these under this IRLM’s CSA usage.

HWM: num K
Is the high water mark or greatest amount of CSA + ECSA allocated by IRLM during this initialization period. Its value is specified as either k for kilobytes or M for megabytes.

ABOVE 16M: num num K
The number of IRLM control block segments above 16M and the amount of storage allocated to those segments. The amount of storage is specified as either k for kilobytes or M for megabytes.

BELOW 16M: num num K
The number of IRLM control block segments below 16M and the amount of storage allocated to those segments. The amount of storage is specified as either k for kilobytes or M for megabytes.

AB CUR: num M
Applies to DB2 for z/OS Version 11 and later: The current 64-bit common storage usage. Its value is specified as either M for megabytes or G for gigabytes. This value accounts for all
64-bit common storage obtained in 
IRLM and it is managed by IRLM.

**AB HWM: num M**

Applies to DB2 for z/OS Version 11 
and later: The high water mark or 
greatest amount of 64-bit common 
storage allocated by IRLM during this 
initialization period. Its value is 
specified as either M for megabytes or 
G for gigabytes.

**PVT USE:**
Displays information about private storage.

**BB CUR: num K**
Is the current below the bar private 
usage. Its value is specified as either K 
for kilobytes or M for megabytes.

**BB CUR: num M**
Is the current above the bar private 
usage. Its value is specified as either M 
for megabytes or G for gigabytes.

**BB HWM: num M**
Applies to DB2 for z/OS Version 11 
and later: The high water mark or 
greatest amount of below the bar 
private storage allocated by IRLM 
during this initialization period. Its 
value is specified as either M for 
megabytes or G for gigabytes.

**AB HWM: num M**
Applies to DB2 for z/OS Version 11 
and later: The high water mark or 
greatest amount of 64-bit private 
storage (above the bar) allocated by 
IRLM during this initialization period 
for lock control structures. Its value is 
specified as either M for megabytes or 
G for gigabytes.

**CLASS text**
A tabular summary of storage allocation, in 
the format:

```
CLASS TYPE SEGS MEM
```

**CLASS text**
Specifies the category of storage. 
Expected values are:

**ACCNT**
The storage allocated to 
IRLM lock control blocks.

**PROC**
The storage allocated to 
IRLM structures used for 
IRLM processing (including 
requests). This storage 
includes CSA, ECSA, and 
IRLM private storage.

**MISC**
The storage allocated for the 
rest of IRLM’s needs. This 
includes trace buffers and 
other diagnostic structures. 
This storage includes CSA, 
ECSA, and IRLM private 
storage.

**TYPE text**
Specifies the subcategory for CLASS. 
For example, T-1 is storage for 
resource block structures, T-2 is 
storage for resource request 
structures, and T-3 is storage for 
requestor structures. The storage for 
types T-1, T-2, and T-3 is allocated 
from ECSA when PC=NO or IRLM 
private storage when PC=YES.

**SEGS num**
The number of storage segments 
allocated

**MEM num K**
The storage specified as . Its value is 
specified in one of the following 
units: K for kilobytes, 
M for megabytes, 
or G for gigabytes.

**System action:**
Processing continues normally.

**Sample output:**
See sample output in the information 
center: [DXR100I](#)

---

**DXR100I**

```
irlmx STATUS SCOPE=nnnnnn
DEADLOCK: dddd SUBSYSTEMS
IDENTIFIED
```

**NAME STATUS UNITS HELD WAITING**

```
ssname zzz wun reh rew
```

**RET_LKS**

```
  rtlks
```

**Explanation:**
This message is issued in response to the 
following command:

```
F irlmproc,STATUS
```

A list of DBMSs identified to the IRLM is displayed. 
Work unit and lock information is displayed for each 
DBMS.

The message variables are:

<table>
<thead>
<tr>
<th>nnnnn</th>
<th>One of the following:</th>
</tr>
</thead>
</table>
| LOCAL | SCOPE=LOCAL was specified as an 
| EXEC parameter when the IRLM was 
| started. |
| GLOBAL | SCOPE=GLOBAL was specified in the 
| IRLMPROC and the IRLM is 
| connected to a data sharing group. |
| DISCON | SCOPE=GLOBAL or 
| SCOPE=NODISCON was specified in |

Chapter 1. DXR messages 3
the IRLMPROC and the IRLM is not connected to a data sharing group.

NoDISC

SCOPE=NODISCON was specified in the IRLMPROC and the IRLM is connected to a data sharing group.

GINIT

SCOPE=GLOBAL was specified in the IRLMPROC and the IRLM is joining a data sharing group.

GTERM

SCOPE=GLOBAL was specified in the IRLMPROC and the IRLM is terminating from a data sharing group.

ddd The deadlock detection interval time for IRLM. The value is measured in milliseconds.

ssname An eight-character DBMS name.

tout DBMS specified timeout value for the resource timeout. The value is measured in seconds.

zzz One of the following:

UP The DBMS is active.

DOWN The DBMS failed.

UP-RO The DBMS subsystem is active and is currently identified to the IRLM as a 'read-only' subsystem.

UP-NS The DBMS subsystem is active and is currently identified to the IRLM as a 'no-share' subsystem.

CLEANUP The IRLM drove the DBMS STATUS exit, indicating a failure condition, and is waiting for that DBMS to respond with a PURGE, indicating cleanup is complete.

SFAIL The IRLM to which the DBMS is identified was disconnected from the data sharing group. The modify locks for all the subsystems on that IRLM were retained by IRLM. All DBMSs that are identified to IRLMs disconnected from a data sharing group are listed as SFAIL by the surviving IRLMs.

wun The number of work units under the DBMS that are holding or waiting for a lock. No work units exist for a failed DBMS. The field is set to ‘...’ for a failed DBMS.

rth The number of resources locked by the DBMS.

rew The number of waiting lock requests. No waiting requests are permitted for a failed DBMS. The field is set to ‘...’ for a failed DBMS.

rtlks The number of retained locks held by a subsystem that failed or was running on an IRLM that failed. In most cases, retained locks are purged when a DBMS reidentifies to IRLM; therefore the number is zero for active DBMSs. However, it is possible for a DBMS to hold both active and retained locks if it is recovering from a previous failure.

If no DBMSs are identified to this IRLM, the line beginning with ssname is replaced with NO INFORMATION AVAILABLE.

System action: Processing continues normally.

NAME ssname

STATUS zzz

RET-LKS rtlks

IRLMID id

IRLM_NAME iname

IRLM_LEVL ilevel

Explanation: This message is issued in response to the following commands:

F irlmproc,STATUS,ALLD
F irlmproc,STATUS,irlmx

where irlmx is the IRLMX + IRLMID fields as specified in the IRLMPROC whose status is requested.

A list of database subsystems identified to the IRLMs in the data sharing group is displayed.

The message variables are:

ssname The eight-character name of the database subsystem.

zzz One of the following:

UP The database subsystem is active.

DOWN The database subsystem failed.

UP-RO The database subsystem is active and is currently identified to the IRLM as a read-only subsystem.

UP-NS The database subsystem is active and is currently identified to the IRLM as a no-share subsystem.

CLEANUP The IRLM drove the database subsystem STATUS exit, indicating a failure condition, and is waiting for that database subsystem to respond with a PURGE, indicating cleanup is complete.

SFAIL The IRLM to which the database subsystem is identified was disconnected from the data sharing group.
group. The modify locks for all the database subsystems on that IRLM were retained by IRLM. All database subsystems that are identified to IRLMs disconnected from a data sharing group are listed as SFAIL by the surviving IRLMs.

\( rtlks \): The number of retained locks held by a subsystem that failed or was running on an IRLM that failed. In most cases, retained locks are purged when a database subsystem reidentifies to IRLM; therefore, the number is zero for active database subsystems. However, it is possible for a database subsystem to hold both active and retained locks if it is recovering from a previous failure.

\( id \): The ID of the IRLM to which the database subsystem is identified.

\( iname \): The name of the IRLM to which the database subsystem is identified.

If no database subsystems are identified to any IRLM or to the IRLM specified, or the IRLM specified is not known, the line beginning with \( ssname \) is replaced with NO INFORMATION AVAILABLE.

\( ilevel \): The IRLM function level that the database subsystem requests. This IRLM level is the level that the database subsystem requires to perform locking operations. This IRLM level might be less than the actual IRLM function level; however, IRLM still operates at the actual function level. If the database subsystem failed and its IRLM is the last IRLM to disconnect from the group, the value will be zero when displayed by any peer member that joins until the failed database subsystem is restarted.

**System action:** Processing continues normally.

---

**DXR103I**  
\( \text{irlmx} \) STATUS IRLMS PARTICIPATING IN DATA SHARING GROUP

**Function Level** \( glv \) \( m \)

<table>
<thead>
<tr>
<th>IRLM-NAME</th>
<th>( iname )</th>
</tr>
</thead>
<tbody>
<tr>
<td>IRLMID</td>
<td>( id )</td>
</tr>
<tr>
<td>STATUS</td>
<td>( zzz )</td>
</tr>
<tr>
<td>LEVEL</td>
<td>( lv )</td>
</tr>
<tr>
<td>SERVICE</td>
<td>( s )</td>
</tr>
<tr>
<td>MIN_LEVEL</td>
<td></td>
</tr>
</tbody>
</table>

**Explanation:** This message displays the active IRLMs in response to the following command:

\[ F \ \text{irlmproc,STATUS,ALL} \]

A list of IRLMs in the group that are actively data sharing is displayed. If not data sharing, the single IRLM is displayed.

\( glv \): The IRLM function level in use by all the IRLM(s) in the data sharing group.

\( m \): A decimal number such as 1, 2, and so on. When more than 10 lines are required in response to a status command, multiple messages are issued with \( m \) incremented by one in each successive message.

\( iname \): A 4-character IRLM name. If there is a '*' appended to the \( iname \), that \( iname \) is the GDM.

\( id \): The ID of the IRLM.

\( zzz \): One of the following:

- **UP**: The IRLM is active.
- **DOWN**: The IRLM failed.

**System action:** Processing continues normally.

---

**DXR104I**  
\( \text{irlmx} \) MAINTENANCE LEVELS

**Explanation:** This display is produced by the MODIFY \( \text{irlmproc,STATUS,MAINT} \) command. Maintenance levels of all IRLM load modules are displayed to the console in two column format except for DXRRRL183, DXRRRFTB, DXRRLM50, and DXRRRL186. Modules show the most recent APAR level and the compile date applied to each CSECT.

**System action:** IRLM processing continues normally.

**Operator response:** Review the maintenance level for any suspected module. If you are unable to correct the problem, contact your IBM Support Center for assistance.

**System programmer response:** No action is required.

---

**DXR105E**  
\( \text{irlmx} \) STOP COMMAND REJECTED. AN IDENTIFIED SUBSYSTEM IS STILL ACTIVE

**Explanation:** A STOP \( \text{irlmproc} \) command was entered, but the IRLM still has active DBMSs identified.

**System action:** The command is not processed, but
DXR106E  IRM processing continues normally.

**Operator response:** Enter the STOP command after all active DBMSs have terminated, or terminate the IRLM abnormally with the *F irlmproc,ABEND* command.

**DXR106E  irlmx INVALID MODIFY COMMAND PARAMETER SPECIFIED**

**Explanation:** A MODIFY command specified the procedure name of this IRLM, but the command either contained parameter values not supported by the IRLM or did not contain a required parameter.

**System action:** The command is not processed, but IRLM processing continues.

**Operator response:** Enter the MODIFY command with the correct parameters.

**DXR107E  irlmx STORAGE NOT AVAILABLE TO PROCESS THIS COMMAND**

**Explanation:** The IRLM cannot obtain sufficient internal working storage to process the command.

**System action:** The command is not processed, but IRLM processing continues.

**Operator response:** Inform the system programmer. Enter the command again after storage is available to process the command.

**DXR108I  irlmx PURGE COMMAND FAILED yyyyyyyyy**

**Explanation:** A PURGE command specified either a name of an active DBMS or a name that is not known to this IRLM.

The *dbmsname* specified on the PURGE,dbmsname command.

*yyyyyyyy* The reason the PURGE command failed:

**PURGE IN PROGRESS:**

Either another PURGE command is being processed, or the IRLM is in the process of cleaning up from the disconnection of another group member.

**NAME NOT FOUND OR ACTIVE:**

The DBMS name specified cannot be found, or that DBMS is identified as an active subsystem and cannot be purged by the PURGE command.

**System action:** The command failed, but IRLM processing continues normally.

**Operator response:** Enter the PURGE command with the correct parameters.

**DXR109I  irlmx PURGE COMMAND COMPLETED FOR yyyyyyyyy**

**Explanation:** A PURGE command completed successfully. *yyyyyyyy* The DBMS name specified on the PURGE,dbmsname command.

**System action:** Processing continues normally.

**DXR110I  irlmx STOP COMMAND ACCEPTED**

**Explanation:** The IRLM accepted the STOP command and is beginning the normal shutdown process.

**System action:** The IRLM is performing shutdown processing. The IRLM subtasks are being quiesced and detached. If the IRLM is actively part of a data sharing group, then the other IRLMs are informed when this IRLM leaves the data sharing group. If IRLM was autostarted by the DB2 IRLM will issue this message when the DB2 terminates.

**Operator response:** No action is required.

**DXR116E  irlmx INITIALIZATION FAILED, CODE=xx DETECTED BY zzzzzzzz**

**Explanation:** The IRLM was unable to complete initialization because of error code *xx*. The name of the module that detected the failure is *zzzzzzzz*.

The values of *xx* are:

- **01** Unable to allocate global storage.
- **02** Unable to allocate local storage.
- **03** The IRLMID was missing or invalid. IRLMID must be specified as a 1 to 3-digit number from 0 through 255. The IRLMID must be unique among all IRLMs belonging to the same data sharing group (having the same GROUP parameter). This code can also be returned if you are using a printable character (such as 'D') for the IRLM identifier. Because of the way z/OS interprets quotes, a printable character must be surrounded by enough single quotes for IRLM to determine that this is a printable character. For the IRLMID parameter in the start-up procedure, there must be seven quotes on either side of the character ('''''''D'''''''). If you are specifying the IRLMID on the EXEC statement, there must be three quotes on either side (''''D''').
- **05** The DEADLOCK parameter was missing or invalid. This parameter must be specified as *iiii,kkkk*, where *iiii* and *kkkk* are 1 to 4-digit numbers from 1 through 9999.
- **06** A z/OS subsystem SSCVT containing the name specified by the IRLMNM parameter does not exist. IRLM was not properly registered with z/OS to use the subsystem interface.
- **07** Error in module *zzzzzzzz* initialization.
Chapter 1. DXR messages

DXR117I irlmx INITIALIZATION COMPLETE

Explanation: The IRLM successfully completed initialization and is available for use. If SCOPE=GLOBAL or NODISCON is specified, message DXR132I is issued when the IRLM connects to the global environment and is capable of inter-system data sharing.

System action: The IRLM is processing normally.

DXR121I irlmx END-OF-TASK CLEANUP
SUCCESSFUL • HI-CSA xxxxxK or xxxxM
- HI-ACCT-CSA xxxxxK or xxxxM
- HI-PVT xxxxxK or xxxxM

Explanation: The IRLM end-of-task routine released all of the z/OS common storage that was in use by the terminating IRLM, except for storage that was required for EOT/EOM processing.

System action: Not notify the system programmer.

System programmer response: Take the following action based on the code:

01 Make global storage available to IRLM.
02 Increase the region size on the IRLM start-up procedure. Analyze the dump for more information.
03 Correct the IRLMID parameter on the IRLM start-up procedure.
04 Correct the DEADLOCK parameter on the IRLM start-up procedure.
05 Verify that the IRLM start-up procedure specifies the desired IRLM name. If IRLMNM specifies the desired name, verify that a z/OS subsystem with that name is defined.
06 Analyze the dump to determine the problem.
07 Correct the PGPROT parameter and reissue the START command.
08 Correct the value for the IRLMID parameter.
09 Verify that the IRLM start-up procedure specifies the desired IRLM name.
10 Correct the MAXCSA parameter on the IRLM start-up procedure.
11 Correct the MAXCSA parameter on the IRLM start-up procedure.
12 Correct the SCOPE parameter on the IRLM start-up procedure.
13 Correct the value for the TRACE parameter.
14 Correct the IRLM PC parameter.
15 IRLM must be run on an MVS/XA or higher system.
16 Correct the MAXUSERS parameter.
17 Correct the XCF group name parameter in the IRLM start-up procedure.
18 Correct the lock table name parameter in the IRLM start-up procedure.
19 Analyze the dump to see what is wrong with CTRACE.
20 Analyze the dump to see what is wrong with CTRACE.
21 Analyze the dump to see what is wrong with CTRACE.
22 Correct the value for the IRLMID parameter.
23 Correct the value for the TRACE parameter.
24 Correct the HASH= value and reissue the START command.
25 Correct the PGPROT parameter and reissue the START command.

For codes 1, 2, 6, 7, 8, 13, 18, 19, and 20, an SDUMP is also created. The IRLM discontinued its initialization processing.

System action: IRLM abend 2018 is issued.

Operator response: Notify the system programmer.

System programmer response: Take the following action based on the code:

01 Make global storage available to IRLM.
02 Increase the region size on the IRLM start-up procedure. Analyze the dump for more information.
03 Correct the IRLMID parameter on the IRLM start-up procedure.
04 Correct the DEADLOCK parameter on the IRLM start-up procedure.
05 Verify that the IRLM start-up procedure specifies the desired IRLM name. If IRLMNM specifies the desired name, verify that a z/OS subsystem with that name is defined.
06 Analyze the dump to determine the problem.
07 Correct the PGPROT parameter and reissue the START command.
08 Correct the value for the IRLMID parameter.
09 Verify that the IRLM start-up procedure specifies the desired IRLM name.
10 Correct the MAXCSA parameter on the IRLM start-up procedure.
11 Correct the MAXCSA parameter on the IRLM start-up procedure.
12 Correct the SCOPE parameter on the IRLM start-up procedure.
13 Correct the value for the TRACE parameter.
14 Correct the IRLM PC parameter.
15 IRLM must be run on an MVS/XA or higher system.
16 Correct the MAXUSERS parameter.
17 Correct the XCF group name parameter in the IRLM start-up procedure.
18 Correct the lock table name parameter in the IRLM start-up procedure.
19 Analyze the dump to see what is wrong with CTRACE.
20 Analyze the dump to see what is wrong with CTRACE.
21 Analyze the dump to see what is wrong with CTRACE.
22 Correct the value for the IRLMID parameter.
23 Correct the value for the TRACE parameter.
24 Correct the HASH= value and reissue the START command.
25 Correct the PGPROT parameter and reissue the START command.

For codes 1, 2, 6, 7, 8, 13, 18, 19, and 20, an SDUMP is also created. The IRLM discontinued its initialization processing.
**DXR122E • DXR131I**

**Multiple forms**: This message has two forms. The form
that is shown above applies to IRLM Version 2.3 or
later. The following form applies to versions before
IRLM Version 2.3:

```
END-OF-TASK CLEANUP
SUCCESSFUL - HI-CSA
 xxxxx
K or xxxxM -
HI-ACCT-CSA xxxxK or xxxxM
```

The HI-CSA is the largest amount of CSA + ECSA
allocated to the IRLM during this initialization. The
value for HI-CSA is displayed as xxxxK for kilobytes or
xxxxM for megabytes.

The HI-ACCT-CSA is the largest amount of CSA +
ECSA allocated to the IRLM for lock control blocks
(MAXCSA applies only to this value). The value for
HI-CSA is displayed as xxxxK for kilobytes or
xxxxM for megabytes.

The HI-PVT is the largest amount of below-the-bar
private storage that is allocated to the IRLM. The value
for HI-PVT is displayed as xxxxK for kilobytes or
xxxxM for megabytes.

**System action**: The IRLM terminated processing.

**Operator response**: None is required. Enter the $irlmproc command to restart the IRLM.

---

**DXR122E  irlmx ABEND UNDER IRLM TCB/SRB IN MODULE xxxxxxx ABEND CODE zzzz**

**Explanation**: The IRLM experienced an unrecoverable
programming error while processing under an IRLM
execution unit. Global sharing with the IRLM issuing
the message is inhibited.

```
xxxxxxx
The name of the module that detected the
failure.
zzzz The abend code associated with the failure.
```

**System action**: The IRLM is performing abnormal
termination processing.

**Operator response**: Notify the system programmer.
Ensure that the SYS1.DUMPxx data set containing the
SDUMP is not overwritten before the dump is printed.
Follow IRLM restart procedures.

**System programmer response**: Use the IPCS service
aid to analyze the dump to determine the cause of the failure.

---

**DXR131I  irlmx OUT OF xxx STORAGE DETECTED DURING DEADLOCK. DEADLOCK PROCESS DELAYED.**

**Explanation**: IRLM detected an out-of-storage
condition during deadlock detection resolution
processing. Rather than abending the IRLM, the
deadlock process for this deadlock detection interval
was ended before completely detecting or resolving
deadlocks. If this condition is permitted to persist,
undetected deadlocks might occur.

```
xxxx The type of storage IRLM was attempting to
get when the out-of-storage condition was
detected. It is either ECSA for extended CSA,
or PRIVATE for IRLM private storage.
```

**System action**: The IRLM cleans up deadlock data
structures before completing deadlock detection or
resolution. Deadlock detection or resolution is attempted in one local deadlock cycle as specified on the IRLM startup procedure parameter.

**Operator response:** Stop and restart the IRLM specifying a larger region size on the IRLM startup procedure. IRLM does not need to be stopped immediately and can remain up indefinitely if DBMS activity is limited to reduce the volume of requests to the IRLM.

---

**DXR132I**  
*irlmx SUCCESSFULLY JOINED THE DATA SHARING GROUP WITH yyyyM LOCK TABLE ENTRIES AND nnnnnnnn RECORD LIST ENTRIES*

**Explanation:** The IRLM connected to the data sharing group, joined the XCF group, and connected to the lock structure. (XCF is the cross-system coupling facility component of z/OS.) The IRLM can now participate in inter-system data sharing.

In the message text:
- yyyy The number of lock table entries in units of 1,048,576. If the number of LTEs is less than one unit, yyyy is zero. The first IRLM to connect to the group will allocate the structure and dictate the values for the group.
- nnnnnnnn The number of record list entries available in the coupling facility. The first IRLM to connect to the group causes structure allocation and dictates these values for the group.

**System action:** The IRLM continues operation.

**Operator response:** No action is required.

---

**DXR133I**  
*irlmx TIMEOUT DURING GLOBAL INITIALIZATION WAITING FOR irlmy*

**Explanation:** A time-out occurred while the IRLM attempted global initialization. This problem occurs because incorrect lock structure or XCF group names were specified on the LOCKTABL and GROUP parameters. (XCF is the cross-system coupling facility component of z/OS.) However, anything that delays global initialization for at least 30 seconds causes this message to be issued.

In the message text, irlmy can be the IRLMNM + IRLMID values for an existing peer member or NOEVENT.

**System action:** IRLM continues to issue this message until one of the following occurs:
- Global initialization completes.
- The system programmer abends the IRLM.
- IRLM terminates abnormally with abend 2025 after an internal limit of time-out messages is reached.

**Operator response:** Notify the system programmer.

**System programmer response:** Use the value of irlmy to determine what is delaying the initialization. If irlmy is the name of a peer IRLM, that member is not sending the required XCF messages to the new member that is joining. This situation might be due to XCF messaging problems, processing problems such as DXR167E, or lack of XCF EVENTs. If the new joining member had a prior failure, make sure that each existing member issued the DXR137I recovery message. If irlmy is NOEVENT, then the problem is probably restricted to the new joining member, and only a dump of this member and XCFAS is likely to be needed.

Check the lock structure and XCF group names to ensure that the names specified for this IRLM match the names of all other group members. You can use XCF commands to display the current members in the XCF group to see if this IRLM joined successfully. If necessary, create a memory dump to assist in failure analysis.

Because this message can be issued as the result of missing interaction with other members, you might need dumps of peer sysplex IRLM members, along with XCFAS. If irlmy is a peer IRLM, create storage dumps of that member and the joining member. If you want to get the new member to successfully join, terminate any peer member whose name appears for irlmy before attempting to start the new member.

---

**DXR134E**  
*irlmx JOIN XCF GROUP wwwwwwww FAILED FOR GLOBAL INITIALIZATION, RETURN=xx, REASON=yyyy*

**Explanation:** The IRLM cannot join the XCF group. (XCF is the cross-system coupling facility component of z/OS.)

wwwwwwww The IRLMGRP name specified on the IRLM startup procedure

**System action:** IRLM does not connect to the data sharing group. The identify request for the DBMS attempting to identify is rejected.

**Operator response:** Notify the system programmer.

**System programmer response:** Determine why the IRLM cannot join. See the IXJOIN section of z/OS MVS Programming: Sysplex Services Reference (SA22-7618) for an explanation of the return and reason codes. Once the reason for the connection failure is corrected, restart the IRLM and reidentify the DBMS to it.

---

**DXR135E**  
*irlmx CONNECT TO LOCK TABLE wwwwwwww FAILED FOR GLOBAL INITIALIZATION, RETURN=xx, REASON=yyyy*

**Explanation:** The IRLM cannot connect to the lock table.
DXR136I • DXR140E

wwwwwwwww

The lock table name passed at identify time or the LOCKTAB name specified on the IRLM startup procedure.

**System action:** IRLM does not connect to the data sharing group. The identify request for the DBMS attempting to identify is rejected.

**Operator response:** Notify the system programmer.

**System programmer response:** Try to determine why the IRLM cannot connect. See the IXLCONN section of z/OS MVS Programming: Sysplex Services Reference for an explanation of the return and reason codes. If the reason for the connection failure was corrected, restart the IRLM and reidentify the DBMS to it.

---

**DXR136I** irlmx HAS DISCONNECTED FROM THE DATA SHARING GROUP

**Explanation:** The IRLM left the data sharing group by disconnecting from both the XCF group and the lock structure. (XCF is the cross-system coupling facility component of z/OS.) IRLM takes this action when all identified DBMSs have issued a QUIT request to the IRLM to terminate their active data sharing activity. The message may also be issued by a new member attempting to join the sysplex while a lock structure rebuild is occurring. In this case, the disconnect is temporary and the IRLM should connect to the group when the rebuild is completed. If the IRLM does not connect to the group when the rebuild is completed, create a storage dump and terminate IRLM.

**System action:** This IRLM disconnected from the data sharing group.

---

**DXR137I** irlmx GROUP STATUS CHANGED. irlmx HAS BEEN DISCONNECTED FROM THE DATA SHARING GROUP

**Explanation:** IRLM irlmx lost connectivity to the data sharing group.

**irlmx** The ID of the disconnected IRLM as explained in IRLM messages and codes.

All surviving IRLMs issue this message. The disconnected IRLM might issue the DXR136I message.

**System action:** IRLM irlmx is disconnected from the data sharing group.

**Operator response:** No response is required on this z/OS system. See the DXR136I message for the proper operator response on the disconnected system.

---

**DXR138E** irlmx QUERY STRUCTURE FOR LOCK TABLE wwwwwwwww FAILED FOR GLOBAL INITIALIZATION, RETURN=xx, REASON=yyyy

**Explanation:** This message is issued when IRLM gets a nonzero return code from a QUERY to XCF for the lock structure size.

**wwwwwwwww**

The lock table name passed at identify time or the LOCKTAB name specified on the IRLM startup procedure.

IRLM is unable to connect to the data sharing group if it does not have a valid structure size.

**System action:** IRLM does not connect to the data sharing group. The identify request for the DBMS attempting to identify is rejected.

**Operator response:** Use the return and reason codes specified in the message to determine the problem with XCF. See the IXCQUERY section of z/OS MVS Programming: Sysplex Services Reference for an explanation of the return and reason codes. Restart the DBMS when the problem is fixed.

**System programmer response:** If the operator is unable to find the problem, determine why the XCF QUERY request failed.

---

**DXR139E** irlmxzzzzzzzz REQUEST FAILED, RETURN=xx, REASON=yyyy

**Explanation:** An IRLM request to z/OS failed with an unexpected return and reason code. The request type is zzzzzzz.

**System action:** IRLM terminates with abend 2025.

**Operator response:** Notify the system programmer.

**System programmer response:** Use the request type, reason and return codes to find out what caused the failure. See z/OS MVS Programming: Sysplex Services Reference for an explanation of the return and reason codes. Save the IRLM dump for analysis by IBM® service.

---

**DXR140E** irlmx IRLMID=zzz IS NOT UNIQUE FOR THE DATA SHARING GROUP IT ATTEMPTED TO JOIN

**Explanation:** An IRLM attempted to join a data sharing group, but IRLMID zzz that was specified in the IRLM startup procedure was already in use by another IRLM in the group.

**System action:** IRLM terminates with abend 2025.

**Operator response:** Notify the system programmer.

**System programmer response:** Change the IRLMID in the startup procedure to a number from 1 through 255 that is not already in use by another IRLM in the data sharing group.
DXR141I  
**irlmx THE LOCK TABLE was allocated in a nonvolatile facility**

**Explanation:** The z/OS coupling facility containing the lock table being used by IRLM is executing in nonvolatile state. The volatile state indicates that there is no battery backup for the facility if a power failure occurs.

**System action:** Data sharing continues normally.

**Operator response:** Notify the system programmer.

**System programmer response:** Take action to move to a nonvolatile structure if that is desired for better availability.

DXR142E  
**irlmx THE LOCK STRUCTURE is xx% in use**

**Explanation:** This message shows what percent of the available capacity of the lock structure is being used by the group and can indicate that some action is needed to relieve the storage. The percentage value that is indicated in the message is the percentage of the capacity in the Record List (RLE) portion of the lock structure.

**System action:** This message stays on the console until the storage falls below 70% or until the operator removes the message. Data sharing continues, possibly with restrictions, depending on how full the lock structure is.

- At 80% full, data sharing continues with no restrictions, but storage is approaching a critical threshold.
- At 90% full, data sharing continues with restrictions. Only 'must-complete' type of requests that require lock structure storage are processed. All others that require lock structure storage are denied with an 'out of lock structure storage' reason code.
- At 100% full, data sharing continues with additional restrictions. Any request that requires lock structure storage is denied with an 'out of lock structure storage' reason code.

**Operator response:** Notify the system programmer if there is a concern about running out of space in the lock structure.

**System programmer response:** Take some action to increase the size of the lock structure if more space is needed. Force update-type transactions to complete in order to free modify locks which will free some lock structure storage.

DXR143I  
**irlmx REBUILDING LOCK STRUCTURE BECAUSE IT HAS FAILED OR AN IRLM LOST CONNECTION TO IT**

**Explanation:** The lock structure failed or one or more IRLMs lost connection to it. The IRLM attempts to rebuild all of the current information into the new lock structure without modifying any of the attributes.

**System action:** Data sharing is stopped while rebuild is in progress. After the rebuild completes successfully, data sharing continues normally. If rebuild does not succeed, IRLM disconnects from the data sharing group.

**Operator response:** Notify the system programmer to repair the failed structure or connections to it.

**System programmer response:** Determine the reason for the structure or connection failure and recover the failed structure.

DXR144I  
**irlmx REBUILDING LOCK STRUCTURE BECAUSE ALL CONNECTIONS TO IT ARE IN USE**

**Explanation:** The last available connection to the lock structure was allocated. IRLM attempts to rebuild into another structure and modify the lock structure attributes to allow more connections.

**System action:** Data sharing is stopped while the rebuild is in progress. After the rebuild completes successfully, data sharing continues normally with more connections available for the group. If the rebuild does not succeed, data sharing continues but new members are not able to join the group.

**Operator response:** Determine whether the group needs to be able to handle more members. If the group needs to handle more members, change the MAXUSRS parameter on the IRLMPROC.

**System programmer response:** Determine why the automatic rebuild was triggered. Take actions to make a bigger structure available to handle the bigger group. Review their MAXUSR value if the number of permanent group members has increased.

DXR145I  
**irlmx REBUILDING LOCK STRUCTURE AT OPERATORS REQUEST**

**Explanation:** The operator started a rebuild with an MVS SETXCF command. IRLM attempts to rebuild into another structure without modifying any of the lock structure attributes.
**DXR146I** • **DXR151I**

**System action:** Data sharing is stopped while rebuild is in progress. After the rebuild completes, data sharing continues normally. If the rebuild failed, data sharing still continues normally.

**Operator response:** Ensure that the rebuild completes successfully. Notify the system programmer if any problem occurs during the rebuild. If the rebuild failed, contact the system programmer to determine why.

**System programmer response:** Determine why the automatic rebuild was triggered. Take actions to make the desired structure available.

**DXR146I**  
*irlmx REBUILD OF LOCK STRUCTURE COMPLETED SUCCESSFULLY WITH yyyyM LOCK TABLE ENTRIES AND nnnnnnnn RECORD LIST ENTRIES*

**Explanation:** The lock structure was successfully moved to another coupling facility structure and might be rebuilt into the same structure or a different structure. In the message text:

- **yyyy** The number of lock table entries (LTE) in units of 1,048,576. If the number of LTEs is less than one unit, yyyy is zero. The first IRLM to connect to the group, causing structure allocation, dictates these values for the group.
- **nnnnnnnn** The number of record list entries available in the coupling facility.

**System action:** Data sharing continues normally.

**DXR147I**  
*irlmx REBUILD OF THE LOCK STRUCTURE WAS STOPPED*

**Explanation:** The lock structure could not be moved to another coupling facility structure. The rebuild process terminated.

**System action:** Data sharing continues normally if the old lock structure is still available. If the lock structure failed, IRLM disconnects from the group.

**Operator response:** Contact the system programmer to determine why the rebuild was stopped.

**System programmer response:** Determine why the rebuild was stopped. Take actions to make a suitable coupling facility structure available for the rebuild.

**DXR148I**  
*irlmx REBUILD OF LOCK STRUCTURE WAS STOPPED DUE TO FAILURE OF THE NEW STRUCTURE*

**Explanation:** Rebuild was stopped because the NEW STRUCTURE failed.

**System action:** The REBUILD is stopped and IRLM will either DISCONNECT from the group or continue to process under the old structure, depending on the reason rebuild was initiated.

**Operator response:** If IRLM disconnects from the group, follow local recovery procedures.

**System programmer response:** Determine what failure occurred on the new structure and take appropriate action to restore rebuild capability.

**DXR151I**  
*irlmx REBUILD OF LOCK STRUCTURE WAS STOPPED DUE TO FAILURE ON QUERY OR CONNECT OF THE NEW STRUCTURE*

**Explanation:** Rebuild was stopped because IRLM received a bad return code on the IXQUERY or IXLCONN request to XCF for the new structure.
**System action:** The REBUILD is stopped and IRLM will either DISCONNECT from the group or continue to process under the old structure, depending on the reason rebuild was initiated.

**Operator response:** If IRLM disconnects from the group, follow local recovery procedures.

**System programmer response:** Determine why the bad return code was received and take appropriate action to restore rebuild capability.

### DXR152I

**Explanation:** Rebuild was stopped because the new structure contains insufficient space to hold all of the data from the old structure or from the restart.

**System action:** The rebuild is stopped and IRLM will either disconnect from the group or continue to process under the old structure, depending on the reason rebuild was initiated.

**Operator response:** If IRLM disconnects from the group, follow local recovery procedures.

**System programmer response:** Determine why the new structure was not large enough to contain all of the data from the old structure or restart. Take appropriate action to restore rebuild capability.

### DXR153I

**Explanation:** Rebuild was stopped because IRLM received a bad return code on a request to SLM.

**System action:** The rebuild is stopped and IRLM will either disconnect from the group or continue to process under the old structure, depending on the reason rebuild was initiated.

**Operator response:** If IRLM disconnects from the group, follow local recovery procedures.

**System programmer response:** Determine why the return code was received from SLM and take appropriate action to restore rebuild capability.

### DXR154I

**Explanation:** IRLM started a REBUILD to establish a new group function level because a new member with a different service level is joining the group. The IRLM rebuild process then checks to see if a complete rebuild is needed to complete the group function level change. This message is issued from the existing members when a joining member cannot coexist with one or more of the existing members of the group.

**System action:** The REBUILD is stopped and the new IRLM member does not join the group. The DBMS that was identifying itself to the new IRLM member is notified that the IDENTIFY call failed. Message DXR157I is issued by the member that was trying to join the group or message DXR158I is issued by other members in the joining member’s behalf.

**System programmer response:** Compare the service level of the IRLM member that failed to join the group with the service level of the other IRLM members in the group. The pertinent service levels can be found by issuing the IRLM modify command “firlmproc,status,alli” on all of the IRLMs involved.

### DXR155I

**Explanation:** IRLM started a REBUILD to establish a new group function level because a new member with a different service level is joining the group. The IRLM rebuild process then checks to see if a complete rebuild is needed to complete the group function level change. This message is issued when a complete rebuild is not needed for a successful group function level change.

**System action:** The REBUILD is stopped and the new IRLM member joins the group.

### DXR156I

**Explanation:** This message is issued when IRLM starts a REBUILD to establish a new group function level when a new member with a different service level is joining the group. Later messages describe further actions IRLM takes during the process of changing the group function level.

**System action:** The REBUILD is started. This allows IRLM to process a group function level change.

### DXR157I

**Explanation:** The IRLM member, irlmx, could not coexist with one or more existing members of the group because of differences in service levels.

**System action:** The REBUILD is stopped and the new IRLM member does not join the group. The DBMS that was identifying itself to the new IRLM member is notified that the IDENTIFY failed.

**System programmer response:** Compare the service
level of the IRLM member that failed to join the group with the service level of the other IRLM members in the group. The pertinent service levels can be found by issuing the MVS modify command `f irlmproc,status,alli` on all the IRLMs involved.

**DXR158I**  
IRLM **irlmx** CANNOT COEXIST WITH AT LEAST ONE EXISTING MEMBER. ALL DBMS IDENTIFY REQUESTS TO THAT IRLM WILL BE DENIED.

**Explanation:** The IRLM member, **irlmx**, could not coexist with one or more existing members of the group because it does not have the service level that includes coexistence support.

**System action:** The REBUILD is stopped and the new IRLM member is not allowed to join the group. The DBMS that identified itself to the new IRLM member is notified that the IDENTIFY failed.

**System programmer response:** Compare the service level of the IRLM member that failed to join the group with the service level of the other IRLM members in the group. The pertinent service levels can be found by issuing the MVS modify command `f irlmproc,status,alli` on all the IRLMs involved.

**DXR159I**  
IRLM **irlmx** RUNTIMEO:TIMEOUT DATA COLLECTION STARTED.

**Explanation:** The MODIFY irlmproc,RUNTIMEO has been successfully issued and time-out detection has been started. When two global deadlock cycles have finished, DXR162I is issued.

**System action:** IRLM processing continues normally with deadlock detection enabled for two global cycles.

**Operator response:** Watch for message DXR162I which indicates that data collection has completed. Then use RMF® to generate the time-out BLOCKER/WAITER report. If no candidate waiters were found, the report will be null and no further action is required unless the waiters persist. If the waiters persist and no 79.15 records are written, attempt to recover by issuing the SETXCF REBUILD command, if applicable. If a report is generated, follow local operation guidelines. If diagnostic analysis is necessary, take z/OS console dumps of the IMS and IRLM address spaces for each member before taking any recovery action. If this is in a sysplex environment, include XESDATA in the dump options. Contact the system programmer.

**System programmer response:** Determine why the IMS does not have time-out support code.

**DXR160I**  
IRLM **irlmx** RUNTIMEO:COMMAND IS NOT SUPPORTED BY AT LEAST ONE IRLM GROUP MEMBER, AND IS REJECTED.

**Explanation:** At least one of the IRLMs in the group does not have the code that is necessary to support TIMEOUT. The TIMEOUT code must be present on all members.

**System action:** IRLM processing continues normally with no time-out detection.

**Operator response:** Notify the system programmer.

**System programmer response:** At least one IRLM in the data sharing group does not have PN79682 applied. The APAR must be applied to all IRLM members of the data sharing group.

**DXR161I**  
IRLM **irlmx** RUNTIMEO:COMMAND IS NOT SUPPORTED BY ANY DBMS SUBSYSTEM ON THIS IRLM, AND IS REJECTED.

**Explanation:** The MODIFY irlmproc,RUNTIMEO has been issued on this IRLM but none of the IMS systems identified to this IRLM provided a time-out exit.

**System action:** IRLM processing continues normally.

**Operator response:** Notify the system programmer.

**System programmer response:** Determine why the IMS does not have time-out support code.

**DXR162I**  
IRLM **irlmx** CYCLE NUMBER nnnnnnnnn PROCESSED FOR TIMEOUT.

**Explanation:** A TIMEOUT candidate has been detected on at least one IRLM. If an incompatible blocker was found, the IMS TIMEOUT EXIT has been driven to present WAITER/BLOCKER information.

**System action:** IRLM processing continues normally. DXR162I is issued only by the Global Deadlock Manager (GDM). The message is issued during the next deadlock cycle after the candidate waiter is found, and then once per minute as long as there is at least one candidate waiter found during the global deadlock cycle. If a cycle occurs and no time-out waiter is found during that time, the flag is reset and DXR162I will be issued the next time a candidate waiter is present.

You can tell which IRLM is the GDM by issuing the F irlmproc,STATUS,ALLI command and finding the IRLM name which is followed by an asterisk '*'. The first one to join, is the GDM until any member DISCONNECTS, then the GDM is recalculated and the one with the lowest ID becomes the GDM.

**Operator response:** Based on local operational guidelines, use the Resource Management Facility to generate a report from the 79.15 (4F 0F) SMF records with cycle number nnnnnnnn. If the message is issued
and no 79.15 records are found, one of these situations is likely to have occurred:

- SMF has not been enabled for the 79.15 to be cut to the MAN* data set. Check the *.IEFU84 member for MODNAME(ERBDSSMF) on each system.
- SMF has not presented the 79.15 to the RMF data space due to a setup error in the data space. Make sure the data space has been defined. Enable the 79.15 to the RMF with: RMF,SMFBUF(RECTYPE(79))
- The 79.15 record with cycle number nnnnnnnn was delayed in writing. Request the same cycle number again.
- IRLM did not find an incompatible blocker
- The lock wait was resolved before the lock wait time expired, but after this message was issued. The lock wait time is specified by the LOCKTIME parameter in:
  - The DFSVSMxx member of the IMS-tm PROCLIB data set, for environments other than IMS batch
  - The data set that is specified by the DFSVSAMP DD statement, for IMS batch

If the waiter cannot be identified and you wish to consult Customer Service, start the IRLM *INTERNAL* CTRACE for subtype XCF on each member, then collect MVS CONSOLE dumps of the IRLM(s) and their IMS DLI and any suspicious dependent regions. If this is in a sysplex environment, you must create and save storage dumps of all IRLMs with XESDATA and associated DBMS regions. Once the dumps are captured, you may turn off XCF tracing. If it appears to be an invalid WAIT condition and this is a sysplex environment, issue a SETXCF REBUILD to attempt a recovery. If this fails to resume the waiter, follow local operation guidelines.

**System programmer response:** Determine if the wait is caused by application logic, operator error, or other factors. (The most common cause is incorrect dispatch priority). If you are unable to correct the problem, contact your IBM Support Center for assistance.

---

**DXR163I**  
_irlmx_ RUNTIMEO:COMMAND IS NOT SUPPORTED BY AT LEAST ONE OF THE OTHER DBMS GROUP MEMBERS, AND IS REJECTED.

**Explanation:** The MODIFY irlmproc,RUNTIMEO has been issued on this IRLM and at least one IMS did not specify a time-out exit on this IRLM or another IRLM in the group.

**System action:** IRLM processing continues normally.

**Operator response:** Notify the system programmer.

**System programmer response:** Determine why the IMS does not have time-out support code.

---

**DXR164E**  
_irlmx_ CANNOT RECOVER FOR A FAILED MEMBER DUE TO LOSS OF CONNECTIVITY TO THE LOCK STRUCTURE

**Explanation:** IRLM detects loss of connectivity to the lock structure during member recovery or global initialization. IRLM is no longer able to support data sharing without the retained lock information.

**System action:** Disconnect from the data sharing group.

**User response:** Notify the system programmer to correct the problem with the coupling facility.

**Problem determination:** See IXC messages in [z/OS V1R11.0 MVS System Messages, Vol 10 (IXC-IZP)](http://www.ibm.com) to determine the root cause of the failure and the appropriate corrective action.

---

**DXR165I**  
_irlmx_ TERMINATED VIA IRLM MODIFY COMMAND

**Explanation:** An operator issued a MODIFY IRLM,ABEND command.

**System action:** IRLM shuts down.

---

**DXR166E**  
_irlmx_ CONNECT TO LOCK STRUCTURE FAILED FOR GLOBAL INITIALIZATION

**Explanation:** IRLM detects loss of connectivity to the lock structure during member recovery or global initialization. IRLM is unable to support data sharing for a longer period of time without the retained lock information.

**System action:** Disconnect from the data sharing group.

**User response:** Notify the system programmer to correct the problem with the coupling facility.

**Problem determination:** See IXC messages in [z/OS MVS System Messages](http://www.ibm.com) to determine the root cause of the failure and the appropriate corrective action.

---

**DXR167E**  
_irlmx_ IRLM HAS DETECTED A DELAY IN COMPLETION OF vvvvvv PROCESS.

**Explanation:** IRLM has detected a delay in a process that might prevent other DBMS or IRLM processes from running. _vvvvvv_ is one of the following values:

*ASIDxxxx*

Issued when IRLM is unable to process main latch activity for some period and finds a DBMS agent that appears to be inactive. This is usually the result of low dispatch priority for some task on an over-committed...
CPU. Can cause all the symptoms listed under System Action.

**MAIN_SRB**

Issued in a sysplex when IRLM detects that a main latch process has been active for an extended period of time. This is usually caused when there are too many waiters in a sysplex such that the IRLM deadlock SRB must execute for long periods of time in order to resolve all of the waiting relationships. MAIN_SRB might cause all of the symptoms listed under System Action.

**DPRIORTY**

Issued when IRLM is unable to get its main latch for some period of time but the Health routine cannot identify any delayed request. This is usually the result of having an incorrect dispatch priority set for the IRLM when compared to the rest of the application tasks or DBMS address spaces. Might cause all of the symptoms listed under System Action.

**NOTIFY_G**

Issued in a sysplex when IRLM detects that this member sent a NOTIFY to peer members and at least one peer has not responded for some period of time. Might cause all of the S2 symptoms listed under System Action.

**NOTIFY_L**

Issued when IRLM finds a NOTIFY Exit drive to its DBMS subsystem has not returned for some period. If this is a sysplex, the peer member should issue NOTIFY_G. Might cause all of the S2 symptoms listed under System Action.

**MAXCYCLE**

Issued when IRLM is unable to process all of the waiters during a deadlock cycle because the number of waiting relationships created would require excessive SRB execution time. Might result in all S2 symptoms listed below. This is usually the result of an application problem or a temporary lock-up that allowed the number of waiters in the sysplex to become excessive, typically in the hundreds. This lock-up could be the result of an application deadlock occurring and not broken quickly because global deadlock is running too slowly. If the transaction rate is high and a deadlock occurs on key resources and is not detected and broken quickly, it can result in many tasks being suspended and involved in deadlocks.

**System action:** The IRLM continues operation. This message stays on the console until the delay is no longer detected. When all DXR167E delays are corrected, IRLM issues DXR168I. The delayed completion might prevent other IRLM or DBMS processes from running. These delayed processes might cause any of the following symptoms:

- A new DBMS cannot identify.
- New members might not be able to join a data sharing group. They will time out during global initialization and eventually terminate abnormally.
- A terminated DBMS is not resumed.
- Failed members might not get properly partitioned from the group because recovery actions for them cannot run.
- Transactions do not deadlock or time out.
- REBUILD of the IRLM will not run.
- S2 symptoms:
  - A transaction locks up.
  - Other processes contending with the locked-up process might also lock up, potentially causing a chain reaction.
  - The whole group might lock up.

**Operator response:** Notify the system programmer. Follow established operation procedures.

**System programmer response:** This delayed process might not cause a noticeable problem to the users. If no problems are observed or reported, monitor the system for the completion of the delayed process. If problems with the system are observed, take the following actions based on the variable term in the message.

**Note:** If you plan to pursue the problem with IBM Service, capture the appropriate documentation before taking recovery actions. If dumps are not captured prior to the DXR167E being removed, IBM Service might not have the documentation needed to resolve the problem. Respond according to the following variables found in *xxxxxxx*.

**ASIDxxxx**

Identify the task in question using *xxxx*, which is the hexadecimal address space identifier (ASID) of the detected task. IRLM will attempt to increase the task's priority in order to get the CPU cycles required to complete the task. The priority of the task will be returned to its previous value when IRLM processing finishes. If message DXR167E with this ASID continues to be displayed and message DXR168I is not displayed, terminate this task to return to normal processing.

**Important:** Take dumps of this DBMS, IRLM, and ASIDxxxx before taking any action. Also, include the text of the DXR167E message from...
the System Console when opening a problem report with IBM Service.

**MAIN_SRB**

You can continue to wait for it to complete, terminate the DBMS or IRLM with the most WAITers shown by the modify command `F irlmproc, STATUS`, or try to terminate WAITing tasks.

**Important:** Take dumps of all DBMSs and IRLM ASIDs before taking any action.

**DPRIORTY**

Correct the dispatching priority order.

**Important:** Take dumps of all DBMSs and IRLM ASIDs before taking any action.

**NOTIFY_G**

Terminate the IRLM or DBMS that issued the NOTIFY_L message.

**Important:** Take dumps of all DBMSs and IRLM ASIDs before taking any action.

**NOTIFY_L**

Terminate the DBMS or IRLM that issued the NOTIFY_L message.

**Important:** Take dumps of all DBMSs and IRLM where the `DXR167E` is issued. If only NOTIFY_G is seen, create storage dumps of all DBMS and IRLM ASIDs before taking any action.

**MAXCYCLE**

Consider lowering the first value of the DEADLOCK parameter in the irlmproc. Issue `MODIFY irlmproc, STATUS` on each member. If the message persists, terminate the IRLM with the largest number of waiters.

**Important:** Take dumps of all DBMSs and IRLM ASIDs before taking any action.

**DXR168I** `irlmx DELAYED PROCESSES NO LONGER DETECTED`

**Explanation:** This message appears if all situations reported in a previous DXR167E messages have been resolved.

**System action:** The IRLM continues normal operation.

**Operator response:** Notify the system programmer.

**System programmer response:** Try to determine what might have caused the temporary problem based on the variable insert in the DXR167E message.

**DXR169I** `irlmx THE STORAGE SHORTAGE FOR LOCK STRUCTURE wwwwwwwww HAS BEEN RELIEVED.`

**Explanation:** The storage shortage indicated by a previously reported DXR142E message is relieved. The amount of the lock structure in use has dropped below 70%.

Is the lock table name currently in use by the IRLM.

**System action:** IRLM resumes normal data sharing with no restrictions.

**DXR170I** `irlmx THE LOCK STRUCTURE wwwwwwwww IS zz% IN USE`

**Explanation:** This message shows what percent of the available capacity of the lock structure is being used by the group. The values are 50%, 60% and 70%. This message is issued only once at each of the three levels of usage and gives an indication of peak usage during the execution of an IRLM subsystem.

The lock table name currently in use by the IRLM.

zz One of the following values: 50, 60, 70.

**System action:** The IRLM continues operation.

**Operator response:** Notify the system programmer.

**System programmer response:** This message is a warning that there is a potential to overcommit the storage for the lock structure named in the message. Track the occurrences of this message to determine a growth trend. If a growth pattern is detected, increase the storage size specifications for the lock structure in the coupling facility policy and activate the revised policy.

**DXR171I** `irlmxxxxxxxx ARM REGISTRATION FAILED, MVS ARM RETURN CODE = yyyyyyyyy, MVS ARM REASON CODE = xxxxxxxx`

**Explanation:** IRLM encountered an error using the specified z/OS automatic restart manager (ARM) function. The `xxxxxxx` is the IRLM ARM element name. For data sharing environments, it is a concatenation of the IRLM group name, the IRLM subsystem name, and the IRLM ID. For non-data sharing environments, it is the IRLM subsystem name and the IRLM ID.

**System action:** IRLM continues. However, if IRLM abends, the z/OS ARM cannot automatically restart IRLM.

**System programmer response:** Refer to the [z/OS MVS Programming: Sysplex Services Reference](https://www.ibm.com/docs/en/zos/2.4.0?topic=smplex-service-reference) for the function, return code, and reason code from the `IXCARM` request.
**DXR172I**  
**irlmxxxxxxxx ARM READY**  
**COMPLETED. MVS ARM RETURN CODE = yyyyyyyyy, MVS ARM REASON CODE = xxxxxxxx**  

**Explanation:** IRLM initialization progressed and enabled the automatic restart manager (ARM) to restart dependent element types (SYSDB2, SYSIMS). xxxxxxxx is the IRLM ARM element name. In a data sharing environment, the element name is a concatenation of the IRLM group name, the IRLM subsystem name and the IRLM ID. In a non-data sharing environment, the element name is the IRLM subsystem name and the IRLM ID. If the return and reason codes are not zero, IRLM encountered an error using the specified z/OS automatic restart manager (ARM) function.

**System action:** IRLM continues. However, if the return and reason codes are not zero, and IRLM abnormally terminates, the z/OS ARM cannot automatically restart IRLM.

**System programmer response:** Refer to [z/OS MVS Programming: Sysplex Services Reference](#) for the function, return code, and reason code from the IXCARM request.

---

**DXR173I**  
**irlm xxxxxxxx ARM**  
**DEREGISTRATION FAILED. MVS ARM RETURN CODE = yyyyyyyyy, MVS ARM REASON CODE = xxxxxxxx**  

**Explanation:** IRLM encountered an error using the specified z/OS automatic restart manager (ARM) function. xxxxxxxx is the IRLM ARM element name. For data sharing environments, it is a concatenation of the IRLM group name, the IRLM subsystem name and the IRLM ID. For non-data sharing environments, it is the IRLM subsystem name and the IRLM ID. If IRLM terminates normally or abnormally, the z/OS ARM automatically restarts IRLM.

**System action:** IRLM continues. However, if IRLM terminates normally or abnormally, the z/OS ARM automatically restarts IRLM.

**System programmer response:** Refer to [z/OS MVS Programming: Sysplex Services Reference](#) for the function, return code, and reason code from the IXCARM request.

---

**DXR174E**  
**irlmxxxxxxxxxxxxsect-name ARM EVENT EXIT FAILED.**

**Explanation:** IRLM ARM event exit encountered an error while attempting to define the IRLM subsystem to z/OS.

**System action:** The automatic restart of the IRLM subsystem stops.

**System programmer response:** Refer to [z/OS MVS Programming: Sysplex Services Reference](#) for return and reason codes from the IEFSSI request. You can manually start IRLM by using the S irlmproc command on any system where the IRLM subsystem is defined. If IRLM is not active and AUTO START = yes for IRLM defined in DSNZPARM, then DB2 will attempt to start IRLM when DB2 is started.

---

**DXR175E**  
**irlm xxxxxxxx IRLM IS UNABLE TO OBTAIN STORAGE - storage_type**  

**Explanation:** IRLM is attempting to acquire storage from CSA, ECSA, or extended private storage. The storage type needed is exhausted, or the monitored threshold was exceeded.

The value in field storage_type specifies the cause of the error:

**MCSA**  
MAXCSA has been exceeded.

Possible causes are:
- The value for the IRLM startup parameter is too low.
- The IRLM workload has an unexpected peak.

**CSA**  
CSA has been exhausted.

Possible causes are:
- The system was generated with too little CSA.
- The demand for CSA storage by jobs and tasks had an unexpected peak.

**ECSA**  
System ECSA has been exhausted.

Possible causes are:
- The system was generated with too little ECSA.
- The demand for ECSA storage by jobs and tasks had an unexpected peak.

**PVT**  
The extended private storage threshold has been reached.

Possible causes are:
- An application did not issue enough COMMITs.
- An application is caught in a loop.
- The IRLM region size is too small.
- A failed subsystem is not recovering retained locks.
- The MEMLIMIT value is too low.

**System action:** The IRLM continues to operate. This message stays on the console until IRLM is able to acquire the needed storage. Some database manager requests might be processed successfully, some might be rejected with out-of-storage return and reason codes, and some might hang.

**Operator response:** Notify the system programmer. Try to reduce the work load on the system or the number of transactions running in the database manager.

When the out-of-storage condition is relieved, normal IRLM operation resumes. Because storage is unavailable, the following symptoms might occur:
• Transactions might abend with out of storage indications.
• Message DXR167E might be issued.
• Transactions might hang.
• Other processes contending with a hung process also hang, potentially causing a chain reaction.
• The system hangs.
• In a data sharing environment:
  – The whole group might hang.
  – New members might not be able to join. They will time out during global initialization and eventually terminate abnormally.
  – Failed members might not get properly partitioned from the group because recovery actions for them cannot run.

System programmer response: Take one of the following actions, depending on the value of storage-type:

MCSA Specify a larger value for MAXCSA, or change the PC parameter to YES.

CSA or ECSA Generate more CSA or ECSA into the system or reduce the number of running tasks that are dependent on CSA or ECSA.

PVT Issue MODIFY irlmproc,STATUS to determine how many locks are held or if there are retained locks. Issue MODIFY irlmproc,STATUS,STOR to determine how much private region storage is available. The storage threshold can be increased by issuing MODIFY irlmproc,SET,MLT. Ensure that the amount of MEMLIMIT that you specify is less than or equal to the amount of MEMLIMIT storage that is available. IRLM does not validate that the specified amount of storage is available.

If the condition is caused by an operational error that can be corrected without shutting down IRLM, correct it as soon as possible. Otherwise, shut down IRLM, correct the problem, and restart IRLM.

DXR176I irlnx IRLM STORAGE SHORTAGE RELIEVED

Explanation: This message appears if a situation reported in a previous DXR175E message has been resolved.

System action: The IRLM continues operation.

Operator response: Notify the system programmer.

System programmer response: Try to determine what might have caused the temporary problem as reported in a previous DXR175E message. The most likely reasons are too small a value for the IRLM startup parameter MAXCSA or insufficient CSA or ECSA for the combined tasks running in z/OS. The MODIFY irlmproc,SET command can be used to change certain IRLM tuning parameters, such as MAXCSA, without restarting IRLM. If possible, correct the problem without shutting down IRLM.

DXR177I irlnx THE VALUE FOR nnnnnnn IS SET TO xxxx

Explanation: This message is issued in response to the following command:
F irlmproc,SET,TRACE=nnn

In the message text, nnnnnnn is one of the following values:
• DEADLOCK
• LTE
• PVT
• MAXCSA
• TIMEOUT
• TRACE
• MLT

System action:

DEADLOCK

When nnnnnnn is DEADLOCK, xxxx is a value between 100 and 5000 and represents the number of milliseconds used for the IRLM deadlock interval. The value displayed will be in even 100 millisecond increments.

If this IRLM is connected to the group, this message appears on each member currently connected as they update their control block during deadlock processing.

The DEADLOCK interval frequency is changed to the number of milliseconds displayed in the message as each IRLM updates its local control blocks. The global deadlock interval is aborted and global deadlock processing re-initialized to synchronize all members.

LTE

When nnnnnnn is LTE, xxxx is a value between 0 and 2048 and represents the number of Lock Table Entries available in the Coupling Facility. The value must be either zero or a power of two with each increment representing 1048576 entries.

If this IRLM is connected to the group, this message appears on each member currently connected as they update their control block during deadlock processing. The value entered with the SET command, does not take affect unless the IRLM is the first to join the group. This can be either during normal group startup or during a REBUILD. If the value specified is zero or too large for the IXQUERY policy size returned, then the IRLMPROC LTE value is used. If the IRLMPROC LTE value is blank, zero, or also too large, the number of entries is calculated...
DXR179I

by IRLM to the nearest power of two after
dividing the returned size by two.

MLT  The value of MEMLIMIT is set to xxxx
megabytes, gigabytes, terabytes, or petabytes,
as indicated by a one character unit indicator
in the set (M, G, T, or P). MLT controls the
amount of above the bar storage made
available by the operating system for the
IRLM's use in managing locks.

MAXCSA
When mmmmmn is MAXCSA, xxxx is the value
of MAXCSA set in megabytes. MAXCSA
controls storage use for lock structures. The
lock control structures are allocated from
ECSA when PC=NO. If PC=YES has been
specified, this command has no affect. The
system automatically adjusts xxxx to a number
between 0 and 999 if the number specified is
not in this range. If the new MAXCSA value is
smaller than the current IRLM CSA allocation
for accountable lock structures, IRLM will
reduce its CSA storage when the storage is no
longer needed.

PVT  The value of PVT is set to xxxx megabytes.
PVT controls the below the bar private storage
that is used by IRLM. The system
automatically adjusts xxxx to a number
between 1 and 1800, if the number specified is
not in this range. If the new PVT value is
smaller than the value that is already in use,
the command receives error message
DXR106E. No validation of the value is
performed to check if the address space
contains the storage that is specified.

TIMEOUT
When mmmmmn is TIMEOUT, xxxx is a value
between 1 and 3600 seconds. This value must
be a multiple of the local deadlock parameter.
If the value entered by the user is not an even
multiple, the next higher multiple is used. If
the value specified is out of range, the
maximum or minimum is used. Since
serialization is needed to change the internal
time-out value, it is done during the next
deadlock cycle. The DXR177I message is not
issued until the input has been validated and
the value changed during a deadlock cycle.
The message also displays the name of the
DBMS subsystem whose time-out value was
changed.

TRACES
When mmmmmn is TRACES, xxxx is a value
between 10 and 255. Each trace buffer requires
64KB of ECSA storage. IRLM can use a
maximum of 255 and a minimum number of
10 trace buffers for each trace type. The system
automatically adjusts xxxx to the maximum or
minimum allowed if the value specified is out
of range. The value specified does not apply if
the EXTERNAL CTRACE write is active. If the
number of trace buffers set by the command is
less than the current buffer allocation, IRLM
frees the oldest buffers until the number
allocated equals the number set by the
command.

Programmer response: No action is required.

DXR179I  irlmx TRACE USAGE TRACE BUFFER
STORAGE IN USE: mmmKB
MAXIMUM NUMBER OF TRACE
BUFFERS ALLOWED PER TRACE
TYPE: m Type TRACE TYPE ACTIVE
BUFFERS IN USE CTRACE WRITER

Explanation: This message is issued in response to the
following command:
F irlmproc,STATUS,TRACE

The message indicates the maximum number of 64KB
buffers that IRLM can use for each of its subcomponent
trace types.

TRACE TYPE
The trace types are:

SLM  Traces interactions with z/OS locking
component. Applicable only for data
sharing.

XIT  Traces asynchronous interactions with
the z/OS locking component.
Applicable only for data sharing.

XCF  Traces interactions with z/OS
cross-system coupling services.
Applicable only for data sharing.

DBM  Traces interactions with the DBMS
identified to this IRLM.

EXP  Traces exception conditions.

INT  Traces member and group events
outside normal locking activity.

RLE  Traces interactions between z/OS and
IRLM specific to Record List Entries.

ACTIVE  Whether the trace is active. (You cannot
deactivate the EXP and INT traces.)

BUFFERS IN USE
How many 64KB buffers are currently being
used by this trace.

CTRACE WRITE
Whether the external CTRACE writer is on.

System action: Processing continues normally.

Operator response: Trace buffers can be set using the
modify irlmproc,set,trace=nnn command. You can
activate or deactivate traces by using the TRACE CT
command of z/OS. You cannot turn off the EXP, RLE
and INT traces. The XIT, EXP, RLE and INT are
automatically activated when you start IRLM All traces
are automatically activated when TRACE=YES is
specified on the IRLMPROC.
**DXR180I irlm** AUTOMATIC RESTART MANAGER IS NOT ENABLED

**Explanation:** IRLM attempts to utilize the z/OS Automatic Restart Manager (ARM) function, but either the system does not have access to an ARM couple data set or the ARM is not set up in the user's environment.

**System action:** IRLM continues. However, if IRLM abends, the z/OS ARM cannot automatically restart IRLM.

**Operator response:** Notify the system programmer.

**System programmer response:** Determine why the system does not have access to an automatic restart manager couple data set. If there is no automatic restart manager couple data set defined, determine if the ARM should be set up so that IRLM can take advantage of the z/OS ARM function. The purpose of using the ARM is to reduce the time IRLM is down. When IRLM stops abnormally, the z/OS ARM restarts it automatically.

**DXR181I irlm** RUNTIMEO COMMAND DISABLED

**Explanation:** The IRLMs are performing automatic time-out detection and the command is not needed.

**System action:** IRLM processing continues.

**DXR182I irlm** MODIFY DIAG COMMAND PROCESSED

**Explanation:** This message is issued in response to an IRLM MODIFY irlmproc,DIAG command.

**System action:** IRLM processing continues.

**DXR183I irlm** DEADLOCK PROCESSING RESTARTED

**Explanation:** An operator issued the MODIFY irlmproc,DIAG,HANG command to collect dumps for problem analysis.

**System action:** The IRLM restarted deadlock processing following an operator MODIFY DIAG,HANG command to collect dumps for problem analysis.

**Operator response:** If this message is NOT issued by any IRLM after this command has been used, that IRLM must be terminated and restarted.

**DXR184I irlm** REBUILDING LOCK STRUCTURE AT REQUEST OF DBMS IRLM QUERY

**Explanation:** This message is issued when IRLM starts a REBUILD as the result of a special request from the DBMS. For DB2 this occurs if a normal RESTART stall is detected by the restart monitor agent. If the stall is caused by incompatible LOCK information between the SYSPLEX members, a REBUILD is in place and the restart continues.

**System action:** The REBUILD is started and should complete normally.

**DXR185I csect-name** IXLFORCE SUCCESSFUL FOR IRLM LOCK STRUCTURE

**Explanation:** IRLM has deleted its lock structure due to a request made by the associated database subsystem. If DB2 is the associated subsystem, lock structure deletion is done if the setting of the DEL_CFSTRUCT_ON_RESTART ZPARM is YES.

**System action:** The lock structure is deleted and recovered by restart process.
Chapter 2. IRLM return and reason codes

The IRLM return and reason codes are listed below. The database management system (DBMS) referred to throughout the return and reason codes is either DB2 or IMS. Some return and reason codes do not apply to DB2 operation.

The values in the 'return code' column show the request completion status:

00  The request completed successfully.
04  The request completed successfully, but the state of lock remained unchanged.
08  The request completed unsuccessfully because of a system error or condition, not a problem in the request.
12  The request completed unsuccessfully because of a logic error in the request.
16  The request completed unsuccessfully because of an invalid request specification.
20  The request completed unsuccessfully because IRLM resources are not available.

CHANGE request

The following table shows the return codes, reason codes, and descriptions for CHANGE requests:

<table>
<thead>
<tr>
<th>Return code</th>
<th>Reason code (byte 1)</th>
<th>Reason code (byte 2)</th>
<th>Description</th>
</tr>
</thead>
</table>
| 04 (X'04') | X'80'                |                      | Ownership or duration, or both was specified, but no locks were changed for one of these reasons:  
|             |                      |                      | • The 'set of locks' specified included no held locks.  
|             |                      |                      | • All specified locks had a duration value greater than or equal to the specified duration.  
<p>|             |                      |                      | • PSTATE=TSTATE was specified. |
| X'20'      |                      |                      | Other holders of the lock exist. |
| X'10'      |                      |                      | The CHANGE ALL request specified the from-duration, but the from-duration did not match the held-duration for at least one lock. The mismatched lock is not altered, but other locks might have been altered. |
| X'08'      |                      |                      | This lock was involved in a deadlock condition, but was not selected as the victim. |
| X'02'      |                      |                      | The CHANGE STATE request specified a 'tstate' value that was less restrictive than the current held state of the lock. |
| X'04'      |                      |                      | At least one other work unit holds the lock with the repeatable read (RR) attribute. |
| X'02'      |                      |                      | The work unit that submitted this lock request holds the lock with the repeatable read (RR) attribute. |</p>
<table>
<thead>
<tr>
<th>Return code</th>
<th>Reason code (byte 1)</th>
<th>Reason code (byte 2)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>08 (X'08')</td>
<td>X'80'</td>
<td></td>
<td>A non-recoverable system error occurred while processing this request.</td>
</tr>
<tr>
<td></td>
<td>X'40'</td>
<td></td>
<td>The lock is not available in the requested state. MODE=COND was specified.</td>
</tr>
<tr>
<td></td>
<td>X'20'</td>
<td></td>
<td>The request was rejected because of a deadlock.</td>
</tr>
<tr>
<td>X'10'</td>
<td></td>
<td></td>
<td>The lock request cannot be granted because another DBMS failed. That DBMS holds a retained lock in a state that is incompatible with the requested state of this request. This subcode is also issued when the lock cannot be granted because of an incompatibility with a lock in NOWAIT status. A lock is placed in this status when a QUIT,HELD=WORKUNIT is issued to the IRLM.</td>
</tr>
<tr>
<td>X'08'</td>
<td></td>
<td></td>
<td>The DBMS is operating in CLEANUP mode because a system failure occurred or because retained locks were deleted without a guarantee that database recovery was done. The DBMS must issue a PURGE request before any global lock request is allowed to wait.</td>
</tr>
<tr>
<td>X'01'</td>
<td></td>
<td></td>
<td>There is insufficient storage for the IRLM to process the CHANGE request. After all other subsystems identified to the IRLM have quit, terminate the IRLM and increase the storage available to the IRLM before restarting it.</td>
</tr>
<tr>
<td>X'80'</td>
<td></td>
<td></td>
<td>This request is in lock negotiation with other DBMSs and was denied.</td>
</tr>
<tr>
<td>X'40'</td>
<td></td>
<td></td>
<td>This request is involved in lock negotiation with other DBMSs. A DBMS downgraded its held state to a still incompatible state or a reason code to deny the conflicting request was not provided.</td>
</tr>
<tr>
<td>X'20'</td>
<td></td>
<td></td>
<td>The request specified the repeatable read (RR) attribute, but not a resource name (RNA) or token.</td>
</tr>
<tr>
<td>X'10'</td>
<td></td>
<td></td>
<td>No more record list entries are available.</td>
</tr>
<tr>
<td>X'08'</td>
<td></td>
<td></td>
<td>An environmental error occurred in the System Lock Manager (SLM)) component of z/OS.</td>
</tr>
<tr>
<td>X'04'</td>
<td></td>
<td></td>
<td>At least one other work unit holds the lock with the repeatable read (RR) attribute.</td>
</tr>
<tr>
<td>X'02'</td>
<td></td>
<td></td>
<td>The work unit that submitted this lock request holds the lock with the repeatable read (RR) attribute.</td>
</tr>
<tr>
<td>Return code</td>
<td>Reason code (byte 1)</td>
<td>Reason code (byte 2)</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>---------------------</td>
<td>---------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>12 (X'0C')</td>
<td>X'80'</td>
<td></td>
<td>An invalid request to change a work unit’s Compatibility Class Token was issued. The value in RLPOLDCT did not match what was specified on a previous lock or change request for this work unit.</td>
</tr>
<tr>
<td>X'40'</td>
<td></td>
<td></td>
<td>The work unit does not hold the lock.</td>
</tr>
<tr>
<td>X'20'</td>
<td></td>
<td></td>
<td>No lock exists for the specified resource hash and name values.</td>
</tr>
<tr>
<td>X'10'</td>
<td></td>
<td></td>
<td>From-duration was specified on a change single request, but the from-duration did not match the held-duration for that lock. The lock is not altered.</td>
</tr>
<tr>
<td>X'08'</td>
<td></td>
<td></td>
<td>The lock is not held in the state specified by 'fstate'.</td>
</tr>
<tr>
<td>X'04'</td>
<td></td>
<td></td>
<td>The resource is not locked by the work unit in the specified class.</td>
</tr>
<tr>
<td>X'02'</td>
<td></td>
<td></td>
<td>A suspended LOCK or CHANGE request exists against the lock. If an ownership/duration change of a group of locks was requested, some locks might have changed before this condition was detected.</td>
</tr>
<tr>
<td>X'01'</td>
<td></td>
<td></td>
<td>The target work unit (TWU) already holds a lock or is waiting for a lock on a resource for which this request is attempting to transfer lock ownership. Ownership/duration change of other locks specified by this request might have occurred before this condition was detected.</td>
</tr>
<tr>
<td>16 (X'10')</td>
<td>X'80'</td>
<td></td>
<td>An invalid 'tstate' was specified. The value is greater than 11.</td>
</tr>
<tr>
<td>X'40'</td>
<td></td>
<td></td>
<td>The CLASS is invalid. The value is greater than 31.</td>
</tr>
<tr>
<td>X'20'</td>
<td></td>
<td></td>
<td>The count-by-state lock has a valid 'tstate', but the 'fstate' is zero.</td>
</tr>
<tr>
<td>X'10'</td>
<td></td>
<td></td>
<td>Both STATE and TWU were specified.</td>
</tr>
<tr>
<td>X'08'</td>
<td></td>
<td></td>
<td>STATE was specified, but TOKEN and RNA were omitted.</td>
</tr>
<tr>
<td>X'04'</td>
<td></td>
<td></td>
<td>The TOKEN is invalid. It cannot be resolved to a lock held by the specified owning work unit.</td>
</tr>
<tr>
<td>X'02'</td>
<td></td>
<td></td>
<td>The resource name length is invalid. The length is less than 2 or greater than 32 bytes.</td>
</tr>
<tr>
<td>X'01'</td>
<td></td>
<td></td>
<td>The 'fstate' is invalid for a count-by-state lock. The value is greater than 11.</td>
</tr>
<tr>
<td>X'20'</td>
<td></td>
<td></td>
<td>A target work unit was given but no owner was specified or the target &amp; owner are the same.</td>
</tr>
<tr>
<td>X'40'</td>
<td></td>
<td></td>
<td>Repeatable reader was specified and no token or resource name was given.</td>
</tr>
<tr>
<td>20 (X'14')</td>
<td>N/A</td>
<td></td>
<td>IRLM services are not available.</td>
</tr>
</tbody>
</table>
### IDENT request

The following table shows the return codes, reason codes, and descriptions for IDENT requests:

<table>
<thead>
<tr>
<th>Return code</th>
<th>Reason code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>04 (X'04')</td>
<td>X'80'</td>
<td>Scope=NoDiscon specified on the irlmproc, identify is successful.</td>
</tr>
<tr>
<td></td>
<td>X'40'</td>
<td>IRLM is registered with the Automatic Restart Manager (ARM).</td>
</tr>
<tr>
<td></td>
<td>X'10'</td>
<td>Locks are held from the previous IDENT request. Locks held by the identifying DBMS from a previous failure are still in protected status. For additional information, see the MODIFY (F) irlmproc,STATUS command.</td>
</tr>
<tr>
<td>08 (X'08')</td>
<td>X'80'</td>
<td>A non-recoverable system error occurred while processing this request.</td>
</tr>
<tr>
<td></td>
<td>X'40'</td>
<td>A z/OS subsystem interface (SSI) error occurred during identification. Ensure that the 4-character IRLM name is defined as a z/OS subsystem, and that the IRLM subsystem with the same name was started.</td>
</tr>
<tr>
<td></td>
<td>X'04'</td>
<td>Identify for recovery (SSTYPE=RECON) was requested, but this IRLM has no retained information about this DBMS.</td>
</tr>
<tr>
<td></td>
<td>X'01'</td>
<td>There is insufficient storage for the IRLM to process the IDENT request. After all other subsystems identified to the IRLM have quit, terminate the IRLM and increase the storage available to the IRLM before restarting it.</td>
</tr>
<tr>
<td>16 (X'10')</td>
<td>X'80'</td>
<td>The lock table name is invalid. It probably contains blanks.</td>
</tr>
<tr>
<td></td>
<td>X'40'</td>
<td>The SSNAME is invalid. An active DBMS in the data sharing group is already using this name.</td>
</tr>
<tr>
<td></td>
<td>X'20'</td>
<td>A required exit is missing. If a TIMEOUT exit is specified, TIMEV must be greater than 0.</td>
</tr>
<tr>
<td></td>
<td>X'10'</td>
<td>A read-only request is not allowed. Locks are retained for this DBMS.</td>
</tr>
<tr>
<td></td>
<td>X'08'</td>
<td>This request violates the single DBMS/IRLM rule.</td>
</tr>
<tr>
<td></td>
<td>X'04'</td>
<td>The address of the BACKUP subsystem name pointed to inaccessible storage.</td>
</tr>
<tr>
<td></td>
<td>X'02'</td>
<td>This IRLM does not support the requested function level. A probable cause is the version/release of this IRLM does not support this version/release of the DBMS. IRLM is back-level.</td>
</tr>
<tr>
<td></td>
<td>X'01'</td>
<td>The IRLM failed to connect to the data sharing group and abended. After fixing the connection error, reconnect the IRLM to the data sharing group. This error can also occur when two versions of IRLM with the same IRLMID, running on separate images, are using the same IRLM JOBNAME.</td>
</tr>
<tr>
<td>20 (X'14')</td>
<td>N/A</td>
<td>IRLM services are not available.</td>
</tr>
</tbody>
</table>
LOCK request

The following table shows the return codes, reason codes, and descriptions for LOCK requests:

<table>
<thead>
<tr>
<th>Return code</th>
<th>Reason code (byte 1)</th>
<th>Reason code (byte 2)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>04 (X'04')</td>
<td>X'80'</td>
<td></td>
<td>A restart lock successfully reacquired a retained lock.</td>
</tr>
<tr>
<td>X'40'</td>
<td></td>
<td></td>
<td>A modify lock was granted, but no Record List Entry (RLE) was requested in the CF.</td>
</tr>
<tr>
<td>X'20'</td>
<td></td>
<td></td>
<td>Other lock holders exist.</td>
</tr>
<tr>
<td>X'10'</td>
<td></td>
<td></td>
<td>The lock is already held.</td>
</tr>
<tr>
<td>X'08'</td>
<td></td>
<td></td>
<td>This lock was involved in a deadlock, but was not selected as a victim.</td>
</tr>
<tr>
<td>X'04'</td>
<td></td>
<td></td>
<td>Another work unit holds the lock private.</td>
</tr>
<tr>
<td>X'04'</td>
<td></td>
<td></td>
<td>At least one other work unit holds the lock with the repeatable read (RR) attribute.</td>
</tr>
<tr>
<td>X'02'</td>
<td></td>
<td></td>
<td>The work unit that submitted this lock request holds the lock with the repeatable read (RR) attribute.</td>
</tr>
<tr>
<td>Return code</td>
<td>Reason code (byte 1)</td>
<td>Reason code (byte 2)</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>---------------------</td>
<td>---------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>08 (X'08')</td>
<td>X'80'</td>
<td></td>
<td>A non-recoverable system error occurred while processing this request.</td>
</tr>
<tr>
<td></td>
<td>X'40'</td>
<td></td>
<td>The lock is not available in the requested state. MODE=COND was specified.</td>
</tr>
<tr>
<td></td>
<td>X'20'</td>
<td></td>
<td>The request was rejected because of a deadlock or time-out.</td>
</tr>
<tr>
<td></td>
<td>X'10'</td>
<td></td>
<td>The lock request cannot be granted because of the failure of another DBMS that holds a retained lock in a state incompatible with the requested state of this request. This subcode is also issued when the lock cannot be granted because of an incompatibility with a lock in NOWAIT status. A lock is placed in this status when a QUIT,HELD=WORKUNIT command is issued to the IRLM.</td>
</tr>
<tr>
<td></td>
<td>X'08'</td>
<td></td>
<td>The DBMS is operating in CLEANUP mode because a system failure occurred or because retained locks were deleted without a guarantee that database recovery was done. The DBMS must issue a PURGE request before any global lock request is allowed to wait. The same reason code is presented if the LOCK request is canceled by a SYNC request.</td>
</tr>
<tr>
<td></td>
<td>X'04'</td>
<td></td>
<td>Another holder holds the restart lock in an incompatible state.</td>
</tr>
<tr>
<td></td>
<td>X'02'</td>
<td></td>
<td>This work unit exceeded the maximum number of locks that it can hold. The LOCKMAX parameter specifies the maximum.</td>
</tr>
<tr>
<td></td>
<td>X'01'</td>
<td></td>
<td>There is insufficient storage for the IRLM to process the LOCK request. After all other subsystems identified to the IRLM have quit, terminate the IRLM and increase the storage available to the IRLM before restarting it.</td>
</tr>
<tr>
<td></td>
<td>X'80'</td>
<td></td>
<td>This request was involved in lock negotiation with other DBMSs and was denied.</td>
</tr>
<tr>
<td></td>
<td>X'40'</td>
<td></td>
<td>This request was involved in lock negotiation with other DBMSs. Either a DBMS downgraded its held state to a still incompatible state, or a reason code to deny the conflicting request was not provided.</td>
</tr>
<tr>
<td></td>
<td>X'20'</td>
<td></td>
<td>Two different work units executing on the same IRLM tried to get the same P-lock. Only one work unit is allowed for each resource name on an IRLM.</td>
</tr>
<tr>
<td></td>
<td>X'10'</td>
<td></td>
<td>No more record list entries are available.</td>
</tr>
<tr>
<td></td>
<td>X'08'</td>
<td></td>
<td>An environmental error occurred in the System Lock Manager (SLM) component of z/OS.</td>
</tr>
<tr>
<td></td>
<td>X'04'</td>
<td></td>
<td>One other work unit holds this lock with the repeatable read (RR) attribute.</td>
</tr>
<tr>
<td></td>
<td>X'02'</td>
<td></td>
<td>The work unit that submitted this request also holds the lock with the repeatable read (RR) attribute.</td>
</tr>
<tr>
<td>12 (X'0C')</td>
<td>X'01'</td>
<td></td>
<td>This IRLM already has a modify lock.</td>
</tr>
<tr>
<td></td>
<td>X'80'</td>
<td></td>
<td>A work unit within this IRLM already holds the P-lock. Only one owning work unit is allowed to hold interest in a P-lock within an IRLM.</td>
</tr>
<tr>
<td></td>
<td>X'40'</td>
<td></td>
<td>An illegal attempt was made to change a Compatibility Class Token (CCT). A previous lock or change request had specified a CCT (non-zero), and the most recent request specified a CCT that did not match.</td>
</tr>
</tbody>
</table>
### NOTIFY request

<table>
<thead>
<tr>
<th>Return code</th>
<th>Reason code (byte 1)</th>
<th>Reason code (byte 2)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>04 (X'04')</td>
<td>X'80'</td>
<td></td>
<td>An asynchronous request was accepted.</td>
</tr>
<tr>
<td></td>
<td>X'40'</td>
<td></td>
<td>The receiver returned additional return code information to the requester.</td>
</tr>
<tr>
<td></td>
<td>X'20'</td>
<td></td>
<td>A selective notify request was made, but there were no DBMSs with the right qualifications to receive this request.</td>
</tr>
<tr>
<td></td>
<td>X'02'</td>
<td></td>
<td>One or more DBMSs that are holding locks failed and could not be notified.</td>
</tr>
<tr>
<td>08 (X'08')</td>
<td>X'80'</td>
<td></td>
<td>A non-recoverable system error occurred while processing this request.</td>
</tr>
<tr>
<td></td>
<td>X'08'</td>
<td></td>
<td>The NOTIFY request is purged by sync.</td>
</tr>
<tr>
<td></td>
<td>X'01'</td>
<td></td>
<td>There is insufficient storage for the IRLM to process the NOTIFY request. After all other subsystems identified to the IRLM have quit, terminate the IRLM and increase the storage available to the IRLM before restarting it.</td>
</tr>
<tr>
<td>12 (X'0C')</td>
<td>X'40'</td>
<td></td>
<td>The lock is not held by the specified owning work unit.</td>
</tr>
<tr>
<td></td>
<td>X'20'</td>
<td></td>
<td>No lock exists for the specified resource hash and name values.</td>
</tr>
<tr>
<td>16 (X'10')</td>
<td>X'80'</td>
<td></td>
<td>The message area address is invalid.</td>
</tr>
<tr>
<td></td>
<td>X'08'</td>
<td></td>
<td>The message length is zero.</td>
</tr>
<tr>
<td></td>
<td>X'04'</td>
<td></td>
<td>The TOKEN is invalid.</td>
</tr>
<tr>
<td></td>
<td>X'02'</td>
<td></td>
<td>An invalid resource name was specified. The length is less than 2 or greater than 32 bytes.</td>
</tr>
<tr>
<td>20 (X'14')</td>
<td>N/A</td>
<td></td>
<td>IRLM services are not available.</td>
</tr>
</tbody>
</table>
### PLEXIT request

The following table shows the return codes, reason codes, and descriptions for PLEXIT requests:

<table>
<thead>
<tr>
<th>Return code</th>
<th>Reason code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 (X'04')</td>
<td>X'80'</td>
<td>The P-lock exit is already enabled.</td>
</tr>
<tr>
<td></td>
<td>X'40'</td>
<td>The P-lock exit is already disabled.</td>
</tr>
<tr>
<td>8 (X'08')</td>
<td>X'80'</td>
<td>A non-recoverable system error occurred while processing this request.</td>
</tr>
<tr>
<td></td>
<td>X'01'</td>
<td>There is insufficient storage for the IRLM to process the PLEXIT request. After all other subsystems identified to the IRLM have quit, terminate the IRLM and increase the storage available to the IRLM before restarting it.</td>
</tr>
<tr>
<td>12 (X'0C')</td>
<td>X'80'</td>
<td>The P-lock exit was not defined at IDENTIFY time.</td>
</tr>
<tr>
<td>20 (X'14')</td>
<td>N/A</td>
<td>IRLM services are not available.</td>
</tr>
</tbody>
</table>

### PURGE request

The following table shows the return codes, reason codes, and descriptions for PURGE requests:

<table>
<thead>
<tr>
<th>Return code</th>
<th>Reason code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>04 (X'04')</td>
<td>X'10'</td>
<td>The locks are already released.</td>
</tr>
<tr>
<td></td>
<td>X'08'</td>
<td>The PURGE-required count was decremented.</td>
</tr>
<tr>
<td>08 (X'08')</td>
<td>X'80'</td>
<td>A non-recoverable system error occurred while processing this request.</td>
</tr>
<tr>
<td></td>
<td>X'01'</td>
<td>There is insufficient storage for the IRLM to process the PURGE request. After all other subsystems identified to the IRLM have quit, terminate the IRLM and increase the storage available to the IRLM before restarting it.</td>
</tr>
<tr>
<td>12 (X'0C')</td>
<td>X'80'</td>
<td>A PURGE TYPE=TAKEOVER was issued, but the requester did not issue a corresponding TAKEOVER request.</td>
</tr>
<tr>
<td></td>
<td>X'40'</td>
<td>A PURGE TYPE=SFAIL was issued, but this IRLM is still part of the data sharing group. An IRLM cannot be in system failed state while it is part of a data sharing group. This code indicates that the DBMS has invalid information about its status in the group.</td>
</tr>
<tr>
<td></td>
<td>X'20'</td>
<td>A PURGE-BY-WORK-UNIT was issued but no owning work unit was specified.</td>
</tr>
<tr>
<td></td>
<td>X'10'</td>
<td>A PURGE-BY-WORK-UNIT was issued but is not supported by all members.</td>
</tr>
<tr>
<td>16 (X'10')</td>
<td>X'01'</td>
<td>A GENERIC PURGE request was made, but the resource name is zero or the name length is out of range.</td>
</tr>
<tr>
<td>20 (X'14')</td>
<td>N/A</td>
<td>IRLM services are not available.</td>
</tr>
</tbody>
</table>

### QUERY request

The following table shows the return codes, reason codes, and descriptions for QUERY requests:

<table>
<thead>
<tr>
<th>Return code</th>
<th>Reason code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>04 (X'04')</td>
<td>X'80'</td>
<td>There is no information meeting the criteria of the specified parameters.</td>
</tr>
<tr>
<td></td>
<td>X'01'</td>
<td>The QUERY request did not specify an area large enough to contain all the data requested.</td>
</tr>
<tr>
<td>Return code</td>
<td>Reason code</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td>08 (X'08')</td>
<td>X'80'</td>
<td>A non-recoverable system error occurred while processing this request.</td>
</tr>
<tr>
<td>X'01'</td>
<td></td>
<td>There is insufficient storage for the IRLM to process the QUERY request. After all other subsystems identified to the IRLM have quit, terminate the IRLM and increase the storage available to the IRLM before restarting it.</td>
</tr>
<tr>
<td>16 (X'10')</td>
<td>X'80'</td>
<td>The AREA address is invalid or zero.</td>
</tr>
<tr>
<td>X'40'</td>
<td></td>
<td>An invalid continuation token was specified.</td>
</tr>
<tr>
<td>X'08'</td>
<td></td>
<td>The AREA length is invalid. The AREA length is less than 4.</td>
</tr>
<tr>
<td>X'04'</td>
<td></td>
<td>The TOKEN is invalid. The TOKEN value does not correspond to an existing lock, or the lock is not held by the designated work unit.</td>
</tr>
<tr>
<td>X'02'</td>
<td></td>
<td>The resource name length is invalid. The length specified is less than 2 or greater than 32 bytes.</td>
</tr>
<tr>
<td>X'01'</td>
<td></td>
<td>The DATA=RETAIN parameter is invalid. SCOPE=WU, OWU, or TOKEN was specified.</td>
</tr>
<tr>
<td>20 (X'14')</td>
<td>N/A</td>
<td>IRLM services are not available.</td>
</tr>
</tbody>
</table>

**QUIT request**

The following table shows the return codes, reason codes, and descriptions for QUIT requests:

<table>
<thead>
<tr>
<th>Return code</th>
<th>Reason code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>04 (X'04')</td>
<td>X'10'</td>
<td>Modify locks are retained.</td>
</tr>
<tr>
<td>08 (X'08')</td>
<td>X'80'</td>
<td>A non-recoverable system error occurred while processing this request.</td>
</tr>
<tr>
<td>X'01'</td>
<td></td>
<td>There is insufficient storage for the IRLM to process the QUIT request. After all other subsystems identified to the IRLM have quit, terminate the IRLM and increase the storage available to the IRLM before restarting it.</td>
</tr>
<tr>
<td>12 (X'0C')</td>
<td>X'80'</td>
<td>The owning work unit (OWU) specified on a HELD=WORKUNIT request was not found.</td>
</tr>
<tr>
<td>X'40'</td>
<td></td>
<td>A HELD=WORKUNIT request did not specify the owning work unit (OWU) parameter.</td>
</tr>
<tr>
<td>X'20'</td>
<td></td>
<td>A QUIT HELD=RETAIN request was already processed.</td>
</tr>
<tr>
<td>X'10'</td>
<td></td>
<td>This IRLM has no knowledge of this DBMS. Either a QUIT HELD=RELEASE was already processed or the DBMS was never identified to this IRLM.</td>
</tr>
<tr>
<td>X'01'</td>
<td></td>
<td>Locks are held, but the HELD parameter was omitted.</td>
</tr>
<tr>
<td>20 (X'14')</td>
<td>N/A</td>
<td>IRLM services are not available.</td>
</tr>
</tbody>
</table>

**RETLOCK request**

The following table shows the return codes, reason codes, and descriptions for QUIT requests:

<table>
<thead>
<tr>
<th>Return code</th>
<th>Reason code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>04(X'04')</td>
<td>X'80'</td>
<td>A retained lock already exists for this resource by this DBMS.</td>
</tr>
<tr>
<td>X'40'</td>
<td></td>
<td>A retained lock on the resource in a different state already exists. The states merged and the resultant state updated the existing retained lock.</td>
</tr>
<tr>
<td>X'20'</td>
<td></td>
<td>The child retained lock was not created because it is covered by the parent retained lock specified in PTOKEN or PNAME.</td>
</tr>
</tbody>
</table>
### STAT request

The following table shows the return codes, reason codes, and descriptions for STAT requests:

<table>
<thead>
<tr>
<th>Return code</th>
<th>Reason code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>16 (X'10')</td>
<td>X'80'</td>
<td>DBMS name specified on the request is either zero or blank.</td>
</tr>
</tbody>
</table>

### SYNC request

The following tables show the return codes, reason codes, and descriptions for SYNC requests:

<table>
<thead>
<tr>
<th>Return code</th>
<th>Reason code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>04 (X'04')</td>
<td>X'80'</td>
<td>At least one request matched the cancel criteria. IRLM canceled the request.</td>
</tr>
<tr>
<td>X'01'</td>
<td></td>
<td>There is insufficient storage for the IRLM to process the SYNC request. After all IRLM identified subsystems have quit, terminate the IRLM and increase the storage available to the IRLM before restarting it.</td>
</tr>
<tr>
<td>08 (X'08')</td>
<td>X'80'</td>
<td>A nonrecoverable system error occurred while processing this request.</td>
</tr>
<tr>
<td>X'01'</td>
<td></td>
<td>There is insufficient storage for the IRLM to process the SYNC request. After all IRLM identified subsystems have quit, terminate the IRLM and increase the storage available to the IRLM before restarting it.</td>
</tr>
<tr>
<td>12 (X'0C')</td>
<td>X'80'</td>
<td>CANCEL=YES was specified, but the work unit held no locks.</td>
</tr>
<tr>
<td>X'40'</td>
<td></td>
<td>The caller specified RESUME=NO, but did not give the TAGET RLPL of the request to cancel.</td>
</tr>
<tr>
<td>14 (X'0E')</td>
<td>X'10'</td>
<td>The work unit was not found.</td>
</tr>
<tr>
<td>20 (X'14')</td>
<td>N/A</td>
<td>IRLM services are not available.</td>
</tr>
</tbody>
</table>

### TAKEOVER request

The following table shows the return codes, reason codes, and descriptions for TAKEOVER requests:

<table>
<thead>
<tr>
<th>Return code</th>
<th>Reason code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>00 (X'00')</td>
<td>N/A</td>
<td>Takeover completed successfully for a DBMS that was identified either to this IRLM or to the IRLM in XCF communication with this IRLM. Restart locks might be requested.</td>
</tr>
</tbody>
</table>
### IRLM return and reason codes

<table>
<thead>
<tr>
<th>Return code</th>
<th>Reason code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>04 (X'04')</td>
<td>X'80'</td>
<td>Takeover completed successfully for a DBMS that this IRLM did not know.</td>
</tr>
<tr>
<td></td>
<td>X'20'</td>
<td>Takeover completed successfully for a DBMS that this IRLM did not know.</td>
</tr>
<tr>
<td>08 (X'08')</td>
<td>X'80'</td>
<td>A non-recoverable system error occurred while processing this request.</td>
</tr>
<tr>
<td></td>
<td>X'01'</td>
<td>There is insufficient storage for the IRLM to process the TAKEOVER request. After all other subsystems identified to the IRLM have quit, terminate the IRLM and increase the storage available to the IRLM before restarting it.</td>
</tr>
<tr>
<td>12 (X'0C')</td>
<td>X'80'</td>
<td>The DBMS that issued the request was not identified as an alternate DBMS.</td>
</tr>
<tr>
<td></td>
<td>X'40'</td>
<td>The DBMS being taken over identified itself as an alternate DBMS, or another DBMS already issued a takeover request.</td>
</tr>
<tr>
<td></td>
<td>X'20'</td>
<td>The DBMS being taken over was not identified to either this IRLM or to any other IRLM that is participating in data sharing.</td>
</tr>
<tr>
<td>20 (X'14')</td>
<td>N/A</td>
<td>IRLM services are not available.</td>
</tr>
</tbody>
</table>

### UNLOCK request

The following table shows the return codes, reason codes, and descriptions for UNLOCK requests:

<table>
<thead>
<tr>
<th>Return code</th>
<th>Reason code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>04 (X'04')</td>
<td>X'80'</td>
<td>One or more locks are held. Some locks did not satisfy the generic resource specification.</td>
</tr>
<tr>
<td></td>
<td>X'20'</td>
<td>Additional locks are held in other states. This applies only to count-by-state locks.</td>
</tr>
<tr>
<td></td>
<td>X'10'</td>
<td>The IRLM cannot release one or more parent locks because child locks exist.</td>
</tr>
<tr>
<td></td>
<td>X'08'</td>
<td>The lock was not fully released.</td>
</tr>
<tr>
<td></td>
<td>X'04'</td>
<td>The work unit holds locks in other classes.</td>
</tr>
<tr>
<td></td>
<td>X'02'</td>
<td>Locks could not be released because the duration specified on the request was lower than the duration previously set.</td>
</tr>
<tr>
<td>08 (X'08')</td>
<td>X'80'</td>
<td>A non-recoverable system error occurred while processing this request.</td>
</tr>
<tr>
<td></td>
<td>X'40'</td>
<td>A P-lock conditional request would have had to wait on a held resource latch. The resource latch could have been held by an in-progress P-lock negotiation or a Notify with Latch request.</td>
</tr>
<tr>
<td></td>
<td>X'01'</td>
<td>There is insufficient storage for the IRLM to process the UNLOCK request. After all other subsystems identified to the IRLM have quit, terminate the IRLM and increase the storage available to the IRLM before restarting it.</td>
</tr>
<tr>
<td>12 (X'0C')</td>
<td>X'40'</td>
<td>The work unit does not hold the lock.</td>
</tr>
<tr>
<td></td>
<td>X'20'</td>
<td>No lock exists for the specified resource hash and name values.</td>
</tr>
<tr>
<td></td>
<td>X'10'</td>
<td>The IRLM cannot release the parent lock because child locks exist.</td>
</tr>
<tr>
<td></td>
<td>X'08'</td>
<td>The lock is not held in the specified STATE.</td>
</tr>
<tr>
<td></td>
<td>X'04'</td>
<td>The lock is not held in the specified class.</td>
</tr>
</tbody>
</table>

---

Chapter 2. IRLM return and reason codes
<table>
<thead>
<tr>
<th>Return code</th>
<th>Reason code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>16 (X'10')</td>
<td>X'40'</td>
<td>The CLASS is invalid. The value is greater than 31.</td>
</tr>
<tr>
<td></td>
<td>X'20'</td>
<td>The STATE is invalid. The value is greater than 11.</td>
</tr>
<tr>
<td></td>
<td>X'10'</td>
<td>The owning work unit (OWU) parameter is not specified.</td>
</tr>
<tr>
<td></td>
<td>X'04'</td>
<td>The TOKEN is invalid.</td>
</tr>
<tr>
<td></td>
<td>X'02'</td>
<td>The resource name (RNA) length is invalid. The length specified is less than 2 or greater than 32 bytes.</td>
</tr>
<tr>
<td></td>
<td>X'01'</td>
<td>TYPE=GENERIC was specified, but no resource name (RNA) was specified.</td>
</tr>
<tr>
<td>20 (X'14')</td>
<td>N/A</td>
<td>IRLM services are not available.</td>
</tr>
</tbody>
</table>

**VERIFY request**

The following table shows the return codes, reason codes, and descriptions for VERIFY requests:

<table>
<thead>
<tr>
<th>Return code</th>
<th>Reason code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>00 (X'00')</td>
<td>X'80'</td>
<td>The VERIFY request completed successfully, but IRLM did not recognize one entry in the VERIFY list.</td>
</tr>
<tr>
<td></td>
<td>X'40'</td>
<td>IRLM recognized at least one entry in the VERIFY list, but the entry did not have matching DBRC RECON status.</td>
</tr>
<tr>
<td>08 (X'04')</td>
<td>X'80'</td>
<td>A non-recoverable system error occurred while processing this request.</td>
</tr>
<tr>
<td></td>
<td>X'01'</td>
<td>There is insufficient storage for the IRLM to process the VERIFY request. After all other subsystems identified to the IRLM have quit, terminate the IRLM and increase the storage available to the IRLM before restarting it.</td>
</tr>
<tr>
<td>16 (X'10')</td>
<td>X'80'</td>
<td>The VERIFY SSNAREA area address is invalid.</td>
</tr>
<tr>
<td>20 (X'14')</td>
<td>N/A</td>
<td>IRLM services are not available.</td>
</tr>
</tbody>
</table>
Chapter 3. IRLM abend codes

This section describes the abend codes issued by the internal resource lock manager (IRLM). The database management system (DBMS) referred to throughout the abend codes is either DB2 or IMS. Some abend codes do not apply to DB2 operations.

2017

**Explanation:** This is an internal error. IRLM processing of a request from a database management system (DBMS) failed. The IRLM issues message DXR123E containing the error ID.

**Analysis:** IRLM established a functional recovery routine (FRR) to intercept abends that occur during its execution under the IMS execution unit (TCB or SRB) during the processing of an RLMREQ request.

The RLMREQ request or IMS exit routine processing terminated abnormally. An IRLM FRR intercepted the abend and requested a z/OS SYS1.LOGREC entry and an SDUMP of the failure. The FRR issued a CALLRTM to terminate the IRLM with abend code 2017. Console message DXR123E, issued when the IRLM terminates, contains the z/OS error ID recorded in the SYS1.LOGREC record and in the SDUMP.

This is a standard abend issued by DXRRL1D0 and DXRRL170. User abend 2017 terminated the IRLM. The original problem was intercepted by the FRR to avoid abnormal termination of the IMS system.

Analyze and, if necessary, report the original failure, not abend 2017.

Analyze the SDUMP to determine the original problem. The problem usually is a program check within IRLM code.

1. Locate the program status word (PSW) and register contents at entry to abend, either from the software LOGREC entry or from the RTM2WA summary in the formatted section of the SDUMP.
   
   If the program status word (PSW) is not within an IRLM module, determine the system component where the abend occurred, and use the diagnostic procedure for that component to resolve the problem. IRLM modules are prefixed with DXR.

2. Use the software LOGREC entry or the RTM2WA summary entry for the original error in the related SRB for problem diagnosis.

3. Register 9 normally contains the address of the RLMCB if the error occurred during IRLM processing.

4. Register 12 normally contains the base register contents for the module in control at the time of the error.

**System action:** The IRLM terminates abnormally. An SDUMP of the IRLM address space is requested.

**System programmer response:** Obtain the SDUMP and check the error ID in the message. If you are unable to correct the problem, contact your IBM Support Center for assistance.

**Problem determination:** The problem is usually a program check within the IRLM code. The dump title includes a module name and an offset into the module where the abend was issued.

**Source:** IMS abends

**Module:** DXRRL1D0, DXRRL170

2018

**Explanation:** An unrecoverable error occurred during IRLM initialization. The IRLM issues message DXR116E containing an error code.

**Analysis:** This is a standard abend issued by DXRRL010.

An analysis of the SDUMP or z/OS dump is necessary to determine the reasons for specific failures. Register 15 at the time the abend command was issued contained one of the following reason codes:

<table>
<thead>
<tr>
<th>Register Contents</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'01'</td>
<td>An invalid EXEC parameter was issued.</td>
</tr>
<tr>
<td>X'02'</td>
<td>A parameter or option was invalid, or a required option was missing or invalid.</td>
</tr>
<tr>
<td>X'03'</td>
<td>An invalid SCOPE parameter value was issued.</td>
</tr>
<tr>
<td>X'04'</td>
<td>IRLM was already active.</td>
</tr>
<tr>
<td>X'05'</td>
<td>IRLM was not defined as a z/OS subsystem.</td>
</tr>
<tr>
<td>X'06'</td>
<td>An error building IRLM RLMCB occurred.</td>
</tr>
<tr>
<td>X'07'</td>
<td>A failure occurred in an IRLM initialization routine.</td>
</tr>
<tr>
<td>X'09'</td>
<td>The IRLM is already active.</td>
</tr>
<tr>
<td>X'0B'</td>
<td>A COMPARE-and-SWAP instruction failed to activate the IRLM SSVT. The IRLM is already active.</td>
</tr>
<tr>
<td>X'0C'</td>
<td>The SETDIE service failed to establish a first deadlock time interval.</td>
</tr>
<tr>
<td>X'0D'</td>
<td>Invalid PC parameter specified.</td>
</tr>
<tr>
<td>X'0E'</td>
<td>PC=yes specified at z/OS and does not support close memory.</td>
</tr>
<tr>
<td>X'0F'</td>
<td>Invalid ITRACE keyword.</td>
</tr>
<tr>
<td>X'10'</td>
<td>Too many APPL parameters specified.</td>
</tr>
</tbody>
</table>
APPL parameter not specified in pairs.
Invalid RULES parameter specified.

System action: The IRLM terminates abnormally. An SDUMP of the IRLM address space is requested.

System programmer response: Analyze the error code in message DXR116E or DXR008E to determine the reason for this failure. If you are unable to correct the problem, contact your IBM Support Center for assistance.

Source: IMS abends
Module: DXRRL010

2019

Explanation: This is an internal error. The IRLM detected either a destroyed storage pool or an invalid request. The IRLM storage manager detected invalid or inconsistent control information within its storage pool structure and did not attempt to complete the request that detected this error. The IRLM issues message DXR122E to the console.

Analysis: This is a standard abend issued by DXRRL400. Analyze the SDUMP to determine the problem.

Register 9 contains the address of the RLMCB. Register 12 contains the base register contents for the module in control at the time of the error. Register 14 contains the return address of the caller of storage management. Register 15 at the time the abend was issued contained one of the following reason codes:

Register Contents
X'01' A GET serialized request detected a destroyed storage pool.
X'02' A GET unserialized request detected a destroyed storage pool.
X'03' A FREE serialized request attempted to free an element that was already freed.
X'04' A FREE serialized request detected a destroyed storage pool.
X'05' A FREE unserialized request attempted to free an element that was already freed.
X'06' A FREE unserialized request detected a destroyed storage pool.

System action: The IRLM terminates abnormally. A SYS1.LOGREC entry and an SDUMP of the IRLM address space is requested.

System programmer response: Obtain the SDUMP. If you are unable to correct the problem, contact your IBM Support Center for assistance.

Problem determination: The dump title includes a module name and an offset into the module where the abend was issued.

Source: IMS abends
Module: DXRRL400

2020

Explanation: The IRLM was abended by the Firlmproc, ABEND command and global sharing with the IRLM was consequently terminated. The IRLM issues message DXR124E to the console.

Analysis: This is a standard abend issued by DXRRL020. The IRLM control block structure is formatted when the SDUMP is printed. This structure shows the IRLM status at the time the SDUMP was taken.

Register 9 contains the address of the RLMCB.

System action: The IRLM terminates abnormally. A z/OS SYS1.LOGREC record and an SDUMP are requested. If the NODUMP parameter was omitted, an SDUMP is requested.

System programmer response: The SDUMP shows the status of the IRLM.

Source: IMS abends
Module: DXRRL020

2022

Explanation: This is an internal error. IRLM issues this abend because one of its subtasks abended. The end-of-task (ETXR) routine specified for the subtask by an ATTACH parameter was entered. The end-of-task routine issued abend 2022 to force IRLM to terminate and issues message DXR122E to the console.

Analysis: This is a standard abend issued by DXRRL010 and DXRRL020. When the GLS TASK subtask abnormally terminates, its ESTAE requests the recording of the failure in SYS1.LOGREC and an SDUMP. Abend 2022 does not take a second SDUMP.

When other IRLM subtasks abnormally terminate, no ESTAE exists, so z/OS takes an abend dump (the default action). In these cases, the SYSABEND or SYSUDUMP dump can provide additional information. In addition, an SDUMP is taken.

Analyze and, if necessary, report the original abend on the subtask (GLS TASK, DXRRL080 or DXRRL0B0), not abend 2022. Analyze the SDUMP to determine the reason for the original abend.

1. Locate the program status word (PSW) and register contents at entry to abend either from the LOGREC entry or from the RTM2WA summary in the formatted section of the SDUMP.

If the program status word (PSW) is not within an IRLM module, determine the system component where the abend occurred. Use the diagnostic procedure for that component to resolve the problem. IRLM modules are prefixed with DXR.
Chapter 3. IRLM abend codes

2023

Register 12 normally contains the base register contents for the module in control at the time of the error.

System action: The IRLM terminates abnormally. Global sharing with the IRLM terminated and a SYS1.LOGREC record and an SDUMP were requested.

System programmer response: Obtain the SDUMP. If you are unable to correct the problem, contact your IBM Support Center for assistance.

Problem determination: The dump title includes a module name and an offset into the module where the abend was issued.

Source: IMS abends

Module: DXRRL010, DXRRL020

Explanation: The IRLM encountered an out-of-storage condition from either a logic error, a subsystem definition error, or an error from a z/OS service that must be performed for the IRLM to continue processing. An example of an IRLM internal error is the detection of an invalid function code in an internal queue element. An example of a z/OS service failure is a nonzero return code from SETDIE. The IRLM deadlock and Intersystem Communication (ISC) functions cannot be performed without SETDIE services. Do not force termination of the IRLM. The IRLM issues message XR175E to the console to help identify the error, followed by the message XR122E, which describes the abend.

Analysis: This is a standard abends issued by various modules. Analyze the SDUMP to determine the reasons for the specific failures.

1. Locate the program status word (PSW) and register contents at entry to abend either from the LOGREC entry or from the RTM2WA summary in the formatted section of the SDUMP. If the program status word (PSW) is not within an IRLM module, determine the system component where the abend occurred. Use the diagnostic procedure for that component to resolve the problem. IRLM modules are prefixed with DXR.

2. Register 9 normally contains the address of the RLMCB if the error occurred during IRLM processing.

3. Register 12 normally contains the base register contents for the module in control at the time of the error.

Register 15 contains one of the following reason codes:

- Register X'01': Module DXRRL210 GETMAIN for the ISL-SEND RLMQE element failed.
- Register X'02': Module DXRRL210 ISL-MERGE detected the same IMS subsystem identified to both IRLMs.
- Register X'03': Module DXRRL210 GETMAIN for merged ISL storage failed.
- Register X'04': Module DXRRL300 SETDIE return code for initial PTB process-delay-time interval failed.
- Register X'05': Module DXRRL310 SETDIE return code for PTB wait-time interval failed.
- Register X'06': Module DXRRL310 SETDIE return code for PTB process-delay-time interval failed.
- Register X'07': Module DXRRL330 detected an unexpected RLMQE element while processing request queues.
- Register X'08': Module DXRRL330 attempted to free an RH-owned RLMQE element. A logic error in the PTB-RH protocols was found.
- Register X'09': Module DXRRL330 request for a storage pool for the RLMPTBQ element failed.
- Register X'0A': Module DXRRL370 issued a request to obtain storage for an RLMPTBQ element that failed.
- Register X'0B': Module DXRRL370 issued a request to obtain storage for an RLMQCE60 element that failed.
- Register X'0C': Module DXRRL370 issued a request to obtain storage for an RLMPNCB element that failed.
- Register X'0D': Module DXRRL370 GETMAIN for an RLMQE element failed.
- Register X'0E': Module DXRRL240 GETMAIN for the deadlock workspace failed.
- Register X'0F': Module DXRRL250 GETMAIN for the deadlock parameter list (DPL) storage failed.
- Register X'10': Module DXRRL250 SETDIE for the deadlock-time interval failed.
- Register X'11': Module DXRRL2A0 or DXRRL2E0 issued a request to obtain storage for an RMLPQCOMQ element that failed.
- Register X'12': Module DXRRL2A0 issued a request to obtain storage for an RMLPQEQ8 element that failed.
- Register X'14': Module DXRRL1F0 issued a request to obtain storage for an MLB that failed.
- Register X'15': Module DXRRL2K0 issued a request to obtain storage for an 11-25 RLMQE that failed.
- Register X'16': Module DXRRL2K0 issued a request to obtain storage for an MLB that failed.
- Register X'17': Module DXRRL2K0 issued a request to obtain storage for an 11-20 RLMQE that failed.
- Register X'18': Module DXRRL2K0 issued a request to obtain storage for a dummy WHB that failed.
- Register X'19': Module DXRRL220 issued a request to obtain storage for a dummy WHB that failed.
- Register X'1A': Module DXRRL2E0 issued a request to obtain storage for a dummy WHB that failed.
- Register X'1B': Module DXRRL2C0 issued a request to obtain storage for an RLSPL that failed.
- Register X'1C': Module DXRRL2C0 issued a request to obtain storage for an SRB that failed.
- Register X'1D': Module DXRRL2G0 issued a request to obtain storage for an 07-05 RLMQE that failed.
Module DXRRL350 issued a double DXRRFSAV macro.

Module DXRRL350 failed to obtain PQE28 storage.

Module DXRRL200 encountered a zero secondary latch use count (RLMUCNT/RLMRHSTA).

Module DXRRL200 failed to get storage for an SRB or RHWKA.

**System action:** Global sharing with this IRLM terminated. A z/OS SYS.LOGREC record and an SDUMP of the IRLM address space are requested and issued unless superseded or suppressed.

**System programmer response:** Refer to the DXR175E message to determine which storage was in error and what the probable cause is. If IBM service is required, obtain the SDUMP. If you are unable to correct the problem, contact your IBM Support Center for assistance.

**Problem determination:** The dump title includes a module name and an offset into the module where the abend was issued.

**Source:** IMS abends

**Module:** DXRRL210, DXRRL220, DXRRL240, DXRRL250, DXRRL300, DXRRL310, DXRRL330, DXRRL370, DXRRL1F0, DXRRL2A0, DXRRL2E0, DXRRL2G0, DXRRL2J0, DXRRL2K0.

---

**2024**

**Explanation:** This is an internal error. The database management system (DBMS) parameter list is invalid.

IRLM established a functional recovery routine (FRR) to intercept abends that occur while executing under the IMS execution unit (TCB or SRB), during the processing of some RLMREQ request.

The IRLM encountered a program check while attempting to access an IMS-owned storage area, either an RLPL (RLMREQ interface control block), or some other storage address passed to the IRLM by the RLPL (that is, a VERIFY list). The error, as well as symptom string data, was recorded in SYS1.LOGREC, and the FRR retried to DXRRL100 to clean up the IRLM-owned structures.

**Analysis:** The user abend did not terminate the IRLM, but abended the IMS execution unit.

Examine the SYS1.LOGREC entry recorded by the IRLM FRR to extract the program status word (PSW) and registers at the time of the abend. At the time the abend was issued, register 10 contained one of the following codes:

**Register Contents**

| X’05’ | An invalid IMS owned RLPL storage address was detected after completion of the IRLM request. The request was processed in cross-memory mode (PC=NO). |
| X’07’ | An invalid IMS owned RLPL storage address was detected before completion of the IRLM request. The request was processed in cross-memory mode (PC=NO). |
| X’09’ | An invalid IMS owned storage address (other than the RLPL) was detected, while processing the request in cross-memory mode (PC=NO). |

In all cases, the invalid IMS-owned storage address is available from the variable recording area (VRA) of the SDWA, recorded within the SYS1.LOGREC entry.

**System action:** The requesting DBMS is abended with user abend 2024. The IRLM continues processing.

**System programmer response:** Obtain the dump of the requesting DBMS. If you are unable to correct the problem, contact your IBM Support Center for assistance.

**Source:** IMS abends

**Module:** DXRRL100

---

**2025**

**Explanation:** This is an internal error. The IRLM request to the cross-system extended services (XES) or the cross-system coupling facility (XCF) failed. The job step ESTAE issued message DXR139E with the failing return and reason codes. Message DXR122E is also issued.

**Analysis:** See messages DXR122E and DXR139E for more information. Obtain the SDUMP of the requesting IMS subsystem. The dump title includes a module name and an offset into the module where the abend was issued.

**System action:** The IRLM terminates abnormally. An SDUMP of the IRLM address space is requested.

**System programmer response:** Obtain the SDUMP. If you are unable to correct the problem, contact your IBM Support Center for assistance.

**Problem determination:** The dump title includes a module name and an offset into the module where the abend was issued.

**Source:** IMS abends

**Module:** DXRCHNGP, DXRRL120, DXRRL2E0, DXRRL2R0, DXRRL2R1, DXRRL2R2, DXRRL2R3, DXRRL2R4, DXRRL2T0, DXRRL2T1, DXRRL2U0, DXRRL2V0, DXRRL7C1, DXRRL7C2, DXRRL711, DXRRL712, DXRRL730, DXRRL732, DXRRL750, DXRRL752, DXRRL753, DXRRL754, DXRRL760, DXRRL770, DXRRL780
2027

**Explanation:** This is an internal error. The IRLM detected a logical inconsistency in either its processing or the local lock structure. IRLM issues message DXR122E to the console.

**Analysis:** Obtain the SDUMP. The dump title includes a module name and an offset into the module where the abend was issued. Contact the IBM Support Center for help in determining the problem.

**System action:** The IRLM terminates abnormally. An SDUMP of the IRLM address space is requested.

**System programmer response:** Obtain the SDUMP. If you are unable to correct the problem, contact your IBM Support Center for assistance.

**Problem determination:** The dump title includes a module name and an offset into the module where the abend was issued.

**Source:** IMS abends

**Module:** DXRGLBL, DXRRRL2R1, DXRRRL2R4, DXRRRL2S0, DXRRRL2T0, DXRRRL2T2, DXRRRL2V0, DXRRRL200, DXRRRL220, DXRRRL248, DXRRRL7B1, DXRRRL700, DXRRRL711, DXRRRL712, DXRRRL770

---

2031

**Explanation:** This is an internal error. The IRLM requested that z/OS terminate the IRLM address space because of a previous error. An abend occurred while processing an SRB dispatched to a cross-system coupling facility (XCF) exit or to an System Lock Manager (SLM) exit in the IRLM address space. The exit functional recovery routine (FRR) issued a CALLRTM with a completion code of 2031 to terminate the IRLM address space. The z/OS error ID was placed in the IRLM RLMCB control block. The job step ESTAE issued message DXR122E.

**Analysis:** See message DXR122E for more information. Obtain the SDUMP. The dump title includes a module name and an offset into the module where the abend was issued.

**System action:** The IRLM terminates abnormally. A SYS1.LOGREC record was written. An SDUMP of the IRLM address space is requested.

**System programmer response:** Obtain the SDUMP. If you are unable to correct the problem, contact your IBM Support Center for assistance.

**Problem determination:** The dump title includes a module name and an offset into the module where the abend was issued.

**Source:** IMS abends

**Module:** DXRRRL2F0
Notices

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

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

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

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

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

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

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

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

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.
IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

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

IBM Corporation
J46A/G4
555 Bailey Avenue
San Jose, CA 95141-1003
U.S.A.

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

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

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

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

**Programming Interface Information**

This information is intended to help you interpret messages that are issued by IMS and DB2 for z/OS. This information also documents General-use Programming Interface and Associated Guidance Information, and Diagnosis, Modification or Tuning information provided by IMS and DB2 for z/OS.

General-use Programming Interfaces allow the customer to write programs that obtain the services of IMS and DB2 for z/OS.

Diagnosis, Modification or Tuning information is provided to help you diagnose, modify, or tune IMS and DB2 for z/OS. Do not use this Diagnosis, Modification or Tuning information as a programming interface.

Diagnosis, Modification or Tuning Information is identified where it occurs, either by an introductory statement to a section or topic, or by the following marking: Diagnosis, Modification or Tuning Information.
Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.