OA60571 Documentation Updates

Note: Throughout this document, new support is highlighted in green, updates to existing documentation are highlighted in blue, and editorial corrections, notes, and existing support missing from the current documentation are highlighted in magenta.

1 PUBLICATIONS

1.1 z/OS: MVS Programming: Sysplex Services Guide - SA23-1400

1.1.1 Using the IXCQUERY Macro

Under the information that IXCQUERY returns when you specify REQINFO=FEATURE . .

QuReqRfIxcNoteResiliency is added after QuReqRfAsyncXi:

QuReqRfIxcNoteResiliency

IXCNOTE service support for REQUEST=NOTEPAD REQTYPE=MODIFY and the LOSSCONNDELETE keyword is available on this system

1.1.2 Using Note Pad Services (IXCNOTE)

The following changes and additions are made in the chapter titled
“Using Note Pad Services (IXCNOTE).”

Programs issue the assembler macro IXCNOTE to use the XCF Note Pad Services to:

- Create and delete an XCF note pad.
- Dynamically modify the number of notes that an XCF note pad needs to hold.
- Create and delete connections to a note pad.
- Create, update, read, and delete notes in a note pad on behalf of a connection

. . . .

1.1.2.1 Note pad concepts and terminology

Under the section titled “Note pad concepts and terminology”, the following changes are made:

Note Pad Services

These services enable your application to create, modify or delete a note pad, and get information about a note pad. The creator of the note pad determines the attributes of the note pad and the number of notes that the note pad can hold. The Note Pad Services can be used to request a dynamic increase or decrease in the number of notes in a note pad in response to increased or decreased demand on note pad capacity to avoid disruptions in application availability during times of high demand. Some specifications require users of the note pad (connections) to adhere to certain conventions or protocols. The query service can be used to determine whether a note pad exists. If it does exist, the query service can be used to get information about the note pad. The note pad will generally exist until it is explicitly deleted or fails. A note pad does not survive a sysplex outage. When deleting a note pad, you can optionally specify various conditions that must be satisfied in order for the delete request to go forward. For example, you might want the delete request to be rejected if the note pad still has connections.

1.1.2.2 Designing your application to use an XCF Note Pad

Under the section titled “Designing your application to use an XCF Note Pad”, the following changes are made:

.
.
.

- Determine how your application will respond when it is unable to create a note because the note pad is either full or constrained. A note pad is full if the maximum number of notes ~~requested by the creator of~~ allowed for the note pad ~~to~~ exist. A note pad is constrained if a note cannot be created even though the note pad is not yet full.

1.1.3 What is an XCF Note Pad

Under the section titled “What is an XCF Note Pad”, the following changes are made:

.
.

When creating a note pad, the application specifies the note pad name, the note pad description, the static note pad information, the desired number of notes, whether duplexing of the host note pad structure is desired, and various protocols that are to be applied to use of the note pad. The protocol choices include indications of whether:

- The number of connections with update access is to be limited to one
- Instance number comparisons are required when connectors update and delete notes
- The connectors are responsible for setting the note tag values
- The maximum note tag value is to be tracked by XCF
- XCF should automatically initiate note pad delete processing when a loss of connectivity to the coupling facility containing the note pad is experienced by all systems with connectors to the note pad (LOSSCONNDELETE protocol)

.
.

A note pad fails and is implicitly deleted when:

- A logically created note pad cannot be physically instantiated
  .
  .
  .
- The XCF note pad catalog fails
- A loss of connectivity to the coupling facility containing the note pad is experienced by all systems with connectors to the note pad and the LOSSCONNDELETE protocol is in effect

.
.
.

1.1.3.1 Note limit

The creator of the note pad indicates the number of notes needed for the note pad. In general, this number will be the maximum number of notes that need to reside in the note pad at the same time. Note Pad Services can be used to request a dynamic modification to increase or decrease the number of notes that can reside in a note pad after a note pad is created. For example, a request to increase the number of notes allowed for the note pad may be issued in response to a note pad full condition detected on a note request.

The note limit is one of the factors considered by XCF when choosing a CF structure to host the note pad. Structures that do not appear to have enough available space to accommodate the requested number of notes will be excluded from the candidate list. The create request is rejected if there is no candidate structure with enough space for the requested number of notes. ~~After the note pad is created,~~ XCF rejects a note request if it would cause the number of notes in the note pad to exceed the specified note limit. In such cases, the note pad is said to be full. There could also be situations where XCF is forced to reject a request that creates a new note even though the note pad is not full **. . . .**

.
.
.

1.1.3.2 Note pad protocols

The creator of the note pad determines the various protocols that are to be applied to the note pad. Choices need to be made with respect to the following protocol options.

Multi-write access

The creator of the note pad specifies the MULTIWRITE keyword to indicate whether the number of connections having update access to the note pad is to be restricted. If MULTIWRITE=NO is specified, only one connection in the entire sysplex is allowed to have update access to the note pad. If such a connection exists, a request to create a second connection with update access is rejected. If the existing connection with update access is deleted, whether explicitly by the application or implicitly by XCF when the connector terminates, a new connection with update access can be created. If MULTIWRITE=YES is specified, more than one connection can be created with update access to the note pad.

.

.

LossConnDelete

The creator of the note pad specifies the LOSSCONNDELETE keyword to indicate whether XCF should automatically initiate note pad delete processing when a loss of connectivity to the coupling facility containing the note pad is experienced by all systems with connectors to the note pad. If LOSSCONNDELETE=NO is specified or defaulted to, then XCF will not automatically initiate note pad delete processing. If LOSSCONNDELETE=YES is specified, then XCF will automatically initiate note pad delete processing when a loss of connectivity to the coupling facility containing the note pad is experienced by all systems with connectors to the note pad.

An application might use LOSSCONNDELETE=YES, if the application would like to perform recovery processing when a loss of connectivity to the coupling facility containing the note pad is experienced by all systems with connectors to the note pad. Without this protocol it would be difficult for the application to determine the scope of the loss of connectivity among the systems with connectors to the note pad and thus be difficult to start any recovery procedures. Delete processing provides a trigger point to fence access to the note pad and allow applications to perform their recovery procedures. An example recovery procedure could be failing over to a backup note pad contained in another coupling facility.

.

.

.

1.1.4 Using the IXCNOTE macro

Under the section titled “Using the IXCNOTE macro”, the following changes are made:

The XCF Note Pad Services are available on systems running z/OS V2R1 and up, or on systems running z/OS V1R13 with APAR OA38450 installed.
XCF Note Pad Resiliency functions which include support for REQUEST=NOTEPAD REQTYPE=MODIFY and LOSSCONNDELETE for REQUEST=NOTEPAD REQTYPE=CREATE are available on systems running with APAR OA60571 installed. Your program might need to determine whether the XCF Note Pad Services and XCF Note Pad Resiliency functions are available for use. To do so, issue the IXCQUERY macro specifying REQINFO=FEATURES to obtain data that indicates whether the XCF Note Pad Services and XCF Note Pad Resiliency function are available. See “Using the IXCQUERY Macro” for more

information.

. . . .

- Note Pad Requests
  - Create a note pad (see “Create note pad” on page 724)
  - Query a note pad (see “Query note pad” on page 725)
  - Modify a note pad (see “Modify note pad” on page 726)
  - Delete a note pad (see “Delete note pad” on page 727)

1.1.4.1 System Authorization Facility (SAF) requirements

Under the section titled “System Authorization Facility (SAF) requirements”,
the following change is made:

The following IXCNOTE requests require SAF authorization:

- Create note pad requires CONTROL access
- Query note pad requires READ access
- Modify note pad requires CONTROL access
- Delete note pad requires CONTROL access

1.1.4.2 Constrained Conditions

Under the section titled “Full considerations”, the complete section is replaced with the following:

If a note cannot be created because the note pad is full, consider deleting one or more notes from the note pad to free up some note space and reissue the request. You might also consider reissuing the request, perhaps after a short delay, if you believe the dynamics of your note pad are such that notes are likely to be deleted through normal activity.

Alternatively, you might try to increase the capacity of your note pad by issuing IXCNOTE REQUEST=NOTEPAD REQTYPE=MODIFY to increase the number of notes that the note pad is allowed to hold (see “Modify note pad” on page 726). If dynamically increasing the note pad capacity is unsuccessful due to note pad host structure constraints (e.g., the MODIFY request fails with a reason code ixcnoteRsnNo#NotesStrResources), manual intervention might be required to resolve the constraint condition. For example, the installation might need to update the CFRM policy to change the size of the host note pad structure. Although disruptive, you might choose to delete the note pad and create the note pad again with a higher note limit in order to increase the capacity of an existing note pad. XCF will attempt to locate the newly created note pad in a host structure with sufficient space to satisfy the higher note limit. Deleting the note pad could be disruptive to your application. Adding another note pad likely increases the complexity of your program. Note that there is no guarantee that a new note pad can be created successfully.

1.1.4.3 Timeout conditions

Under the section titled “Timeout conditions”, the following changes are made:

Note the following for note pad requests:

Create, modify, and delete note pad requests are not considered by XCF to be requests that can be safely reissued. If these requests were to be reissued after a timeout, a reissued create request might be rejected because the note pad was in fact created by the original request. Similarly, a reissued delete note pad request might be rejected because the note pad was in fact deleted by the original request. A re-issued modify request might be rejected because a previous modify request may still be in progress. Under the assumption that your program would treat these rejections as unexpected errors, XCF returns a status unknown condition instead of a timeout condition.

1.1.4.4 Status Unknown Conditions

Under the section titled “Considerations for note pad requests”, the following changes are made:

If you were creating a note pad, the note pad might or might not have been created. If you were deleting a note pad, the note pad might or might not have been deleted. If you were modifying the number of notes that a note pad is allowed to hold, the update might or might not have been made effective. Your program might need to take a recovery action to determine how to proceed. For example, you might issue a query note pad request to see if the note pad exists (for a create or delete) or whether a modify request is pending and what the current number of notes allowed value is for the note pad and then proceed appropriately. If you simply reissue the request, your program might need to deal with additional failure conditions. For example, a second create note pad request might be rejected because the note pad already exists; a second delete note pad request might be rejected because the note pad no longer exists; a second modify may be rejected because a previous modify is in progress. If you reissue a delete note pad request, consider specifying ETODCREATED to ensure that the correct instance of the note pad is deleted. If the original request actually deleted the note pad, some other thread might create a new instance of the note pad before the delete request can be reissued. Without the ETODCREATED specification, the reissued delete note pad request would unintentionally delete this new instance of the note pad.

A query note pad request can always be reissued.

1.1.5 Note pad requests

Under the section titled “Note pad requests”, the following changes are made:

Issue the IXCNOTE macro with REQUEST=NOTEPAD to manipulate the note pad as a whole. Such a request is often simply called a note pad request. Use the NOTEPAD keyword to specify the name of the note pad to be processed. The note pad can be created, queried, modified, or deleted.

The note pad must be created before connections can be created, and connections must be created before your application can manipulate notes in the note pad.

Use the REQTYPE keyword to indicate the type of operation to be performed for the note pad. Refer to the following material for specific information on each request type:

- For REQTYPE=CREATE, which is used to create a new note pad, see "Create note pad" on page 724.
- For REQTYPE=QUERY, which is used to get information about a note pad, see "Query note pad" on page 725.
- For REQTYPE=MODIFY, which is used to modify attributes of an existing note pad, see "Modify note pad" on page 726.
- For REQTYPE=DELETE, which is used to delete an existing note pad, see "Delete note pad" on page 727.

When your program issues a note pad request, the requesting work unit is suspended. In general, the note pad request is processed asynchronously under a task in the XCF address space. The optional keyword TIMEOUT can be used to control how long the work unit remains suspended waiting for results. In general, the default values used by XCF can be used. See "Timeout conditions" on page 719 for additional information. Your program needs appropriate SAF authorization to be able to create, query, modify, or delete a note pad. See "System Authorization Facility (SAF) requirements" on page 716 for more information.

1.1.5.1 Modify note pad ** NEW SECTION **

Section titled "Modify note pad" is added after section "Query note pad."

Modify note pad

Issue a modify note pad request to modify the number of notes (#NOTES attribute) that a note pad is allowed to hold after creating a note pad. You may, for example, want to request an increase in the number of notes that a note pad is allowed to hold in response to a note pad full condition or prior to an expected need for increased note pad capacity during times of high demand. Upon successful completion of the request, the requested number of notes allowed becomes effective immediately. An answer area is optional. If an answer is provided, it must be large enough to hold the answer area header.

To successfully implement and complete a dynamically effective increase or decrease to the number of notes that a note pad is allowed to hold, XCF Note Pad Services must reflect the requested change in the XCF note pad catalog and the host structure where the note pad resides. In general, several tasks must be accomplished to propagate the change in notes allowed for a note pad. For example, XCF might need to fence the note pad in the host structure to temporarily prevent note processing while decreasing the number of notes that a note pad is allowed to hold, and update the XCF note pad catalog and note pad multiple times during the course of processing an increase or decrease modify request. Usually, the XCF Note Pad Services successfully perform these operations as needed and the update to the #NOTES attribute is effective by the time the service routine returns control to your program.

However, XCF might not be able to perform some or all of these tasks if the note pad host structure or the note pad catalog structure is not accessible. If XCF Note Pad Services successfully initiated the modify request, XCF will automatically finish the necessary updates and complete the modify request when the note pad host structure and note pad catalog structure become accessible. In such cases, the modify request might complete with a return and reason code indicating:

- **MODIFY request pending completion**. XCF Note Pad Services successfully initiated the MODIFY request, but could not complete all the necessary tasks associated with completing the modify request. The request will complete when the necessary updates are made in the note pad host structure and note pad catalog structure by a system that has access to the structures. No other modify request can be initiated until the current pending modify request completes successfully. An action to consider taking when receiving a return and reason code indicating that the request is pending completion is using a connection request to pause and resume the connection when XCF completes the MODIFY request. XCF Note Pad Services can also be used to query the note pad to determine whether a MODIFY request is pending against the note pad.

- **Status unknown**. The status of the modify request is unknown. The modify request might have completed updating the number of notes the note pad is allowed to hold, might not have ever been started, or might be pending. Consider using a query note pad request to determine the state of the note pad. i.e., whether a modify request is in progress and what the current number of notes is that the note pad is allowed to hold. See “Status unknown conditions” on page 721 for more information.

If the modify request completes successfully and an answer area is provided, information returned in the answer area header details field includes the maximum number of notes that the note pad is currently allowed to hold as a result of the modify request, the number of notes that currently exist in the note pad and note pad host structure information that may be useful to the application.

1.2 z/OS: MVS Programming: Sysplex Services Reference - SA38-0658

1.2.1 IXCNOTE – XCF Note Pad Interface

1.2.1.1 Environment

Under “Minimum authorization”, the following changes are made:

Programs require appropriate SAF (System Authorization Facility) authorization to the FACILITY class resource IXCNOTE.owner.application when creating, deleting, querying, or modifying a note pad, and when creating a connection to a note pad. The "owner" and "application" are derived from the note pad name.

. . . .

- To create or delete a note pad or modify note pad attributes, the program must have CONTROL access.

If SAF is not available, or if there is no IXCNOTE.owner.application resource defined for the note pad in the FACILITY class, a request to create, delete, query, **modify** or connect to a note pad is rejected if the program is running in problem state with a PKM allowing key 8 to 15 (IXCNOTERSNNOSECPROFILE).

1.2.1.2 Restrictions

Under “Restrictions”, the following additions are:

- When using REQUEST=NOTEPAD, REQTYPE=MODIFY to modify the number of notes (#NOTES) that a note pad is allowed to hold, IBM recommends that if connections to the note pad exist, then all connections to the note pad be from systems that support REQUEST=NOTEPAD, REQTYPE=MODIFY. To determine whether the support for REQUEST=NOTEPAD, REQTYPE=MODIFY is available on the system from which you are connecting, issue IXCQUERY REQINFO=FEATURES. QuReqRflxcNoteResiliency indicates whether the support is available.

  If a connection to the note pad resides on a system that does not support REQUEST=NOTEPAD, REQTYPE=MODIFY (a "down level" system), an IXCNOTE request issued to create, read, modify, or delete notes in the note pad from the down level system while a note pad #NOTES is being modified may render the note pad unusable from the down level system for the life of the IPL. Restoring access to the note pad from the down level system will require a re-IPL of the down level system or deleting and re-creating the note pad.

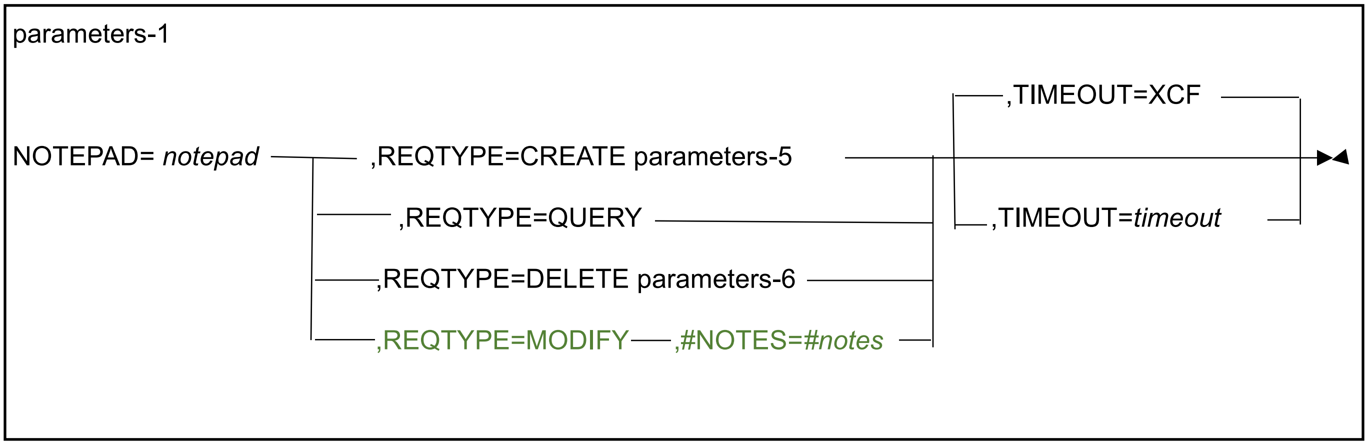
- For REQUEST=NOTEPAD, REQTYPE=CREATE, LOSSCONNDELETE=YES, IBM recommends that if connections to the note pad exist, then all connections to the note pad be from systems that support LOSSCONNDELETE=YES.

  If a connection to the note pad resides on a system that does not support LOSSCONNDELETE=YES (a "down level" system), an IXCNOTE request issued to create, read, modify, or delete notes in the note pad from the down level system may not be properly fenced when the note pad is automatically being deleted by LOSSCONNDELETE processing.

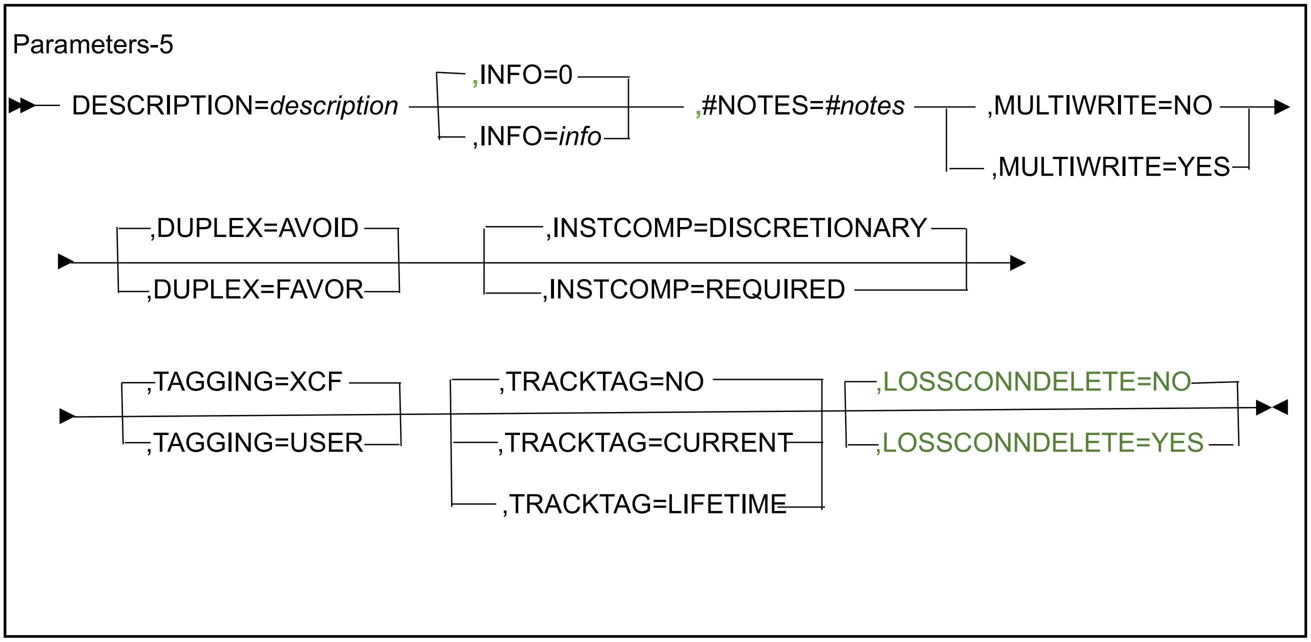
  If a "down level" system exists in the sysplex, it is possible that the LOSSCONNDELETE=YES specification will not be honored. This can occur if an "up level" system needs help from another system in the sysplex to initiate the LOSSCONNDELETE processing and the available helping system(s) are "down level". An "up level" system may need help initiating the LOSSCONNDELETE processing if the system does not have access to the note pad catalog structure.

1.2.1.3 Syntax Diagram

For parameters-1 **REQTYPE=MODIFY, #NOTES=*#notes*** is added:



For parameters-5 **LOSSCONNDELETE=YES | NO** is added:



1.2.1.4 Parameter Descriptions

Under the heading for **Parameters for REQUEST=NOTEPAD**

,#NOTES=x#notes

When REQTYPE=CREATE . . .

When REQTYPE=MODIFY is specified, use this required input parameter to change the maximum number of notes that a note pad is allowed to hold. The specified value may be greater than (increase) or less than (decrease) the current #NOTES value for the note pad.

The specified value must be at least one (1).

. . .

,INSTCOMP=DISCRETIONARY
,INSTCOMP=REQUIRED

When REQTYPE=CREATE is specified, use this optional parameter to indicate whether the users of the note pad are required to perform instance number comparisons when replacing or deleting a note (see INSTANCE# keyword for REQUEST=NOTE). The default is INSTCOMP=DISCRETIONARY.

,INSTCOMP=DISCRETIONARY

Instance number comparisons are not required.

When a note pad connector calls IXCNOTE REQUEST=NOTE to replace, write, or delete an existing note, any INSTANCE# specification is permitted. Instance number comparisons will be performed only if the connector so requests.

,INSTCOMP=REQUIRED

Instance number comparisons are required.

When a note pad connector calls IXCNOTE REQUEST=NOTE to replace, write, or delete an existing note, an instance number comparison must be performed to ensure that the correct instance of the note is being manipulated. For REQTYPE=REPLACE and REQTYPE=DELETE, the request is immediately rejected unless a nonzero INSTANCE# value is specified. For REQTYPE=WRITE, the request proceeds. If the note does not exist, the write request will create a new note (if the request is otherwise valid). If the note already exists, the write request is rejected.

,LOSSCONNDELETE=NO
,LOSSCONNDELETE=YES

When REQTYPE=CREATE is specified, use this optional parameter to indicate whether XCF can automatically initiate note pad delete processing when all systems with connectors to the note pad have lost connectivity to the coupling facility hosting the note pad. The default is NO.

,**LOSSCONNDELETE=NO**

XCF will not automatically initiate note pad delete processing when all systems with connectors to the note pad have lost connectivity to the coupling facility hosting the note pad.

,LOSSCONNDELETE=YES

XCF will automatically initiate note pad delete processing when all systems with connectors to the note pad have lost connectivity to the coupling facility hosting the note pad.

As a result of XCF automatically initiating delete processing the following will occur:

- Any existing notes will be deleted as part of deleting the note pad.
- Any existing connections will be deleted as part of deleting the note pad. Depending on the timing as to when the deletion is recognized, connections that are deleted in this manner might have their IXCNOTE requests rejected with a variety of reason codes (including for example, IXCNOTERSNNOTEPADNOTEXIST, IXCNOTERSNCONNECTIONNOTEXIST, and IXCNOTERSNNOTENOTEXIST). Any existing paused connections would be resumed and then deleted as part of deleting the note pad. If an answer area was provided on the PAUSE request, dr_ResumeCode=ixcynote_kResumeNotePadDeleted would be set in the data area mapped by ixcynote_tDetailsResumed.

. . .

,REQTYPE=CREATE
,REQTYPE=QUERY
,REQTYPE=DELETE
,REQTYPE=MODIFY

Use this required parameter to indicate the type of request to be processed against the note pad.

. . .

,REQTYPE=DELETE

Delete the indicated note pad, provided the specified conditions (if any) are satisfied.

Any existing notes will be deleted as part of deleting the note pad.

Any existing connections will be deleted as part of deleting the note pad. Depending on the timing as to when the deletion is recognized, connections that are deleted in this manner can have their IXCNOTE requests rejected with a variety of reason codes (including for example, IXCNOTERSNNOTEPADNOTEXIST, IXCNOTERSNCONNECTIONNOTEXIST, and IXCNOTERSNNOTENOTEXIST).

,REQTYPE=MODIFY

Increase or decrease the number of notes a note pad is allowed to hold.

. . .

1.2.1.5 Return and Reason Codes for IXCNOTE:

The following updates are made to *Table 14.* Return *and Reason Codes for the IXCNOTE Macro*

Table 14. Return and Reason Codes for the IXCNOTE Macro

Hexadecimal Return Code	Hexadecimal Reason Code	Equate Symbol Meaning and Action
4	xxxx0441	Equate Symbol: IXCNOTERSNPENDING Meaning: The request is pending. . . . . • When modifying the #NOTES attribute for a note pad, not all of the necessary updates could be accomplished because XCF was not able to access the relevant coupling facility structures. XCF will . automatically complete the modify request when access to the structures is restored. **Actions:** . . . . • If REQUEST=NOTEPAD, REQTYPE=MODIFY was

		specified, consider one of the following: • When a connection to the note pad exists, consider using IXCNOTE REQUEST=CONNECTION REQTYPE=PAUSE to be resumed when XCF completes the MODIFY request. • IXCNOTE REQUEST=NOTEPAD REQTYPE=QUERY can also be used to determine whether a MODIFY request is pending against the note pad. npd_Modifying#Notes is set ON if the note pad #NOTES is in the midst of being modified. When npd_Modifying#Notes is set OFF, npd_Required#Notes reflects the number of notes that the note pad is allowed to hold. npd_Required#Notes will not reflect the requested change to the #NOTES attribute while the modify request is pending.
8	xxxx0037	Equate Symbol: IXCNOTERSN#NOTESEXCEEDED Meaning: The note request cannot be processed because the note pad already contains the maximum number of notes. If an answer area was provided, the field AA_DETAILS provides additional information. The data is mapped by IXCYNOTE_TDETAILSCONSTRAINED. Action: Try the request again after deleting some notes, or after issuing an IXCNOTE REQUEST=NOTEPAD REQTYPE=MODIFY request that successfully increased the number of notes that the note pad is allowed to hold.
C	xxxx0C03	**Equate Symbol: IXCNOTERSNMODIFYINPROGRESS** **Meaning**: A REQUEST=NOTEPAD REQTYPE=MODIFY to change the #NOTES value for a note pad is rejected because a MODIFY request to change the #NOTES attribute is still in progress. A request is either currently in progress updating the note pad #NOTES attribute and CF structure control information or a previous MODIFY request is pending completion due to an inability to access the note pad catalog structure or note pad structure or both. **Action:** • When a connection to the note pad exists, consider using IXCNOTE REQUEST=CONNECTION REQTYPE=PAUSE

		to be resumed when XCF completes the MODIFY request. • IXCNOTE REQUEST=NOTEPAD REQTYPE=QUERY can also be used to determine whether a MODIFY request is pending against the note pad. npd_Modifying#Notes is set ON if the note pad #NOTES is in the midst of being modified. When npd_Modifying#Notes is set OFF, npd_Required#Notes reflects the number of notes that the note pad is allowed to hold. npd_Required#Notes will not reflect the requested change to the #NOTES attribute while the modify request is pending
C	xxxx0C0A	**Equate Symbol: IXCNOTERSNNO#NOTESSTRRESOURCES** **Meaning:** The requested #NOTES cannot be committed to the note pad because the current number of data entries reserved for XCF and for note pads in the host note pad structure plus the requested #Notes would exceed the maximum entries supported by the structure. **Action:** Retry the request after allowing time for the condition to clear. In some cases, manual intervention might be required to resolve the problem. For example, the installation might need to update the CFRM policy to change the size of the host note pad structure. In other cases, the typical dynamics of the system could cause existing note pads to be deleted or note pad notes to be deleted which in turn would allow an increase in #NOTES for the note pad.
C	xxxx0C0B	**Equate Symbol: IXCNOTERSNNOTSUPPORTMODIFY** **Meaning:** A system that has connectors to the note pad does not support IXCNOTE REQUEST=NOTEPAD, REQTYPE=MODIFY. All systems with connectors to the note pad must support REQTYPE=MODIFY to be able to modify the #NOTES definition for a note pad **Action**: Connectors on a system that does not support REQTYPE=MODIFY should temporarily disconnect from the note pad until the MODIFY #Notes request completes. . When using REQUEST=NOTEPAD, REQTYPE=MODIFY to modify the #Notes that a note pad is allowed to hold, IBM recommends that if connections to the note pad exist, then all connections to the note pad be from systems that support REQUEST=NOTEPAD, REQTYPE=MODIFY. To determine whether the support for

		REQUEST=NOTEPAD, REQTYPE=MODIFY is available on the system from which you are connecting, Issue IXCQUERY REQINFO=FEATURES. QuReqRfIxcNoteResiliency indicates whether the support is available.
C	xxxx0C0C	**Equate Symbol: IXCNOTERSNDECREASENOTVALID** **Meaning:** The requested #NOTES cannot be less than the current number of notes in use in the note pad. **Action**: Delete existing notes from the note pad, then retry the MODIFY request to decrease the #Notes allowed for a note pad.
C	xxxx0C0D	**Equate Symbol: IXCNOTERSNNOTEPADNOTUSABLE** **Meaning:** The note pad is no longer usable by the local system receiving this reason code due to an enabled note pad feature(s) that is not compatible with the level of note pad support on the local system. i.e., the current instance of the note pad contains features or exploitation of function that are not understood by the local system. **Action:** Restoring access to the note pad from the local system will require a re-IPL of the local system or deleting and re-creating the note pad.
C	xxxx0CBD	Equate Symbol: IXCNOTERSNSTATUSUNKNOWN . . . . **Action:** . . . The following is added at the end of the **Action** section : When modifying the #NOTES attribute for a note pad, processing to update the #NOTES attribute might or might not have been initiated and / or completed. The requester should take an appropriate recovery action. For example, prior to reissuing the request, the requestor might consider issuing IXCNOTE REQUEST=NOTEPAD REQTYPE=QUERY to determine whether a MODIFY request is pending against the note pad.

1.3 z/OS MVS Data Areas Volume 3 (ITK - RQE) - GA32-0937

1.3.1 IXCYNOTE

New aa_DetailsFormat value:

ixcynote_kDetailsModify#Notes Constant('0D'x)

New mapping of **aa_details** returned for REQUEST=NOTEPAD, REQTYPE=MODIFY is added:

- New DSECT ixcynote_tDetailsModify#Notes
- dm#n_#NotesInUse
- dm#n_#NotesAllowed
- dm#n_#EntriesInUse
- dm#n_#EntriesAllowed
- dm#n_#ElementsInUse
- dm#n_#ElementsAllowed
- dm#n_#NotesRequested
- dm#n_#EntriesReserved

New flags in DSECT ixcynote_tNotepadData

New field in npd_flags:
- npd_ValidLossConnDelete
  ON – when npd_LossConnDelete is valid

New field in npd_Status_Flags1:

- npd_Modifying#Notes
  ON – the #notes for the note pad is in the midst of being modified
  OFF – there are no pending requests to modify #notes for the note pad

New field in npd_Attributes:

- npd_LossConnDelete
  ON – the creator of the note pad specified LossConnDelete=Yes
  OFF – the creator of the note pad specified or defaulted LossConnDelete=No

1.3.2 IXCYCON

New Reason Codes for IXCNOTE Return Code 12. These reason codes will also be documented in the IXCNOTE macro prolog and in *z/OS: MVS Programming: Sysplex Services Reference*

For REQUEST=NOTEPAD, REQTYPE=MODIFY

xxxx0C03 ixcnoteRsnModifyInProgress

xxxx0C0A ixcnoteRsnNo#NotesStrResources

xxxx0C0B ixcnoteRsnNotSupportModify

xxxx0C0C ixcnoteRsn#NotesDecreaseNotValid

xxxx0C0D ixcnoteRsnNotepadNotUsable

1.3.3 IXCYQUAA

New features bit for MODIFY and LOSSCONNDELETE support

QuReqRflxcNoteResiliency - IXCNOTE service support for REQTYPE=MODIFY and LOSSCONNDELETE is available on this system.   .

1.4 z/OS MVS System Messages Vol. 10 (IXC-IZP)

1.4.1 IXC443I

In the message, *text* is:

```
[WARNING: INFORMATION MAY NOT BE COMPLETE]
INFO FOR NOTE PAD npname
DESCRIPTION: np_description
HOST STRUCTURE: hoststrname
STATUS: np_status
SYSTEMS CONNECTED: sysname sysname sysname sysname
                   sysname sysname sysname sysname
CREATED: np_created
LIST NUMBER: listnum
MAX TAG: npmaxtag_hex | npmaxtag_ebcdic |
CURRENT NUMBER OF NOTES: currnotes
NOTE PAD DEFINITION
REQUIRED NUMBER OF NOTES: req#notes
TAGGING: nptagging
TRACK TAG: nptracktag
MULTIWRITE: npmultwrite
LOSSCONNDELETE: nplossconndelete
DUPLEX: npduplex
INFO: info_hex | info_ebcdic |
```

In response to a DISPLAY XCF command, this message displays detailed note pad information. The system repeats the display lines as many times as necessary to provide all data.

In the message text:

hh.mm.ss

The time in hours(00-23), minutes (00-59), and seconds(00-59) for the DISPLAY XCF command.

.

.

.

nplossconndelete

One of the following:

YES

The creator of the note pad specified LOSSCONNDELETE=YES

NO

The creator of the note pad specified or defaulted LOSSCONNDELETE=NO

1.4.2 IXC473I

The message text**:**

IXC473I NOTE PAD *npname* {IS BEING DELETED | HAS BEEN DELETED}
NOTE PAD CREATION TOD: *createtod*
[REQUESTER JOB NAME: *jobname* SYSTEM NAME: *sysname*
REASON: *reason*]

Explanation:

XCF has successfully initiated delete processing for a note pad. The note pad is being deleted or has been deleted.

In the message text:

npname

Name of the note pad whose status is being reported.

.
.
.

Reason

Reasons why the note pad deletion was initiated. *reason* is one of the following:

USER REQUEST

An explicit request was made to delete the note pad, either through the IXCNOTE interface or through the note pad delete utility.

HOST STRUCTURE GONE

The structure containing the note pad has either failed or has been deallocated.

UNABLE TO FINALIZE NOTE PAD CREATION

XCF was unable to finalize the note pad creation.

LOSSCONNDELETE PROCESSING

All systems with connectors to the note pad have lost connectivity to the coupling facility hosting the note pad.

System action

The system continues processing.

Source

Cross System Coupling Facility (SCXCF)

Module

IXCN1PAD

Routing code
2, Note 13

Descriptor code
4