File Manager for z/OS V8R1

Addenda to V8R1 User's Guides and Customization Guide

fmv8apar

File Manager for z/OS V8R1

Addenda to V8R1 User's Guides and Customization Guide

fmv8apar

Contents

About this document

This document provides details of all the APAR service fixes that impact upon documentation, for IBM File Manager for z/OS Version 8.1, since the most recent edition of the product manuals in September 2007. These editions are:

- Customization Guide (SC19-1238-01) - Second Edition
- User's Guide and Reference (SC19-1239-00) - First Edition
- User's Guide and Reference for DB2 Data (SC19-1240-00) - First Edition
- User's Guide and Reference for IMS Data (SC19-1241-00) - First Edition
- User's Guide and Reference for CICS (SC19-1242-00) - First Edition

The Addendum document is divided into two parts:

- **Part One: PTF/APAR documentation changes**

  This section lists the changes to the File Manager for z/OS Version 8 Release 1 documentation that are required to reflect new behavior resulting from the application of APAR fixes.

  The fixes are listed by PTF number, in reverse date order, so that the most recently released fix appears at the beginning of the document. Each description shows:

  – The set of PTF numbers in the release
  – The date of the PTF release
  – The APARs included in the released fix
  – Details of those APAR changes that affect documentation
  – Page references for the manuals affected by the change

  Notes:

  1. This document does NOT describe those APAR fixes that do not have an impact upon documentation.
  2. The enhancements and corrections described in this section are only available after applying the listed PTFs for the APAR.

- **General documentation changes**

  This section describes enhancements, corrections and updates in the documentation for File Manager for z/OS Version 8 Release 1. These changes are not associated with PTF numbers, as they do not require the application of any code updates.

  The changes are grouped by manual and listed within each section in reverse date order. That is, the most recent documentation change appears at the beginning of each manual section.

Part 1. PTF/APAR documentation changes

UK47198, UK47210, UK47238, UK47239, UK47240, UK47241, UK47242

Release Date: **12 June 2008**

This set of PTFs contains the following APAR fixes:

APAR #	APAR Abstract	Doc Impact
PK85997	File Manager miscellaneous problems.	Customization Guide (SC19-1238-01)

PK85997

Initial problem description

In sample job FMN2CGEN, the description of changes to the ’AS TEMP’ clause does not explicitly state that removal of the clause is an option, for DB2 version 9 systems only.

Outline of solution

File Manager DB2 component has been updated to correct the problem.

Documentation impact

This APAR requires changes to be made to:

- Customization Guide (SC19-1238-01)

Changes to the Customization Guide

Chapter 13, ″Customizing File Manager/DB2″

In the section, ″Creating a segmented table space (required)″, after the second paragraph insert the following text:

Note: For DB2 version 9, the use of temporary databases is discontinued. For DB2 version 9, you should either remove the ’AS TEMP’ clause, or change the clause to ’AS WORKFILE’. The latter option is only available in a DB2 data-sharing environment.

UK43612, UK43614, UK43619, UK43655, UK43670, UK43686, UK43688, UK43709, UK43710, UK43718, UK43719, UK43720

Release Date: **6 February 2009**

This set of PTFs contains the following APAR fixes:

APAR #	**APAR Abstract**	**Doc Impact**
PK69959	FM/DB2 issues error messages attempting to access a remote DB2 location when select access to SYSIBM.LOCATIONS is not available.	User's Guide and Reference for DB2 Data (SC19-1240-00)
PK72083	Batch FM/IMS does not seem to provide the same functionality regarding template/views as foreground does.	User's Guide and Reference for IMS (SC19-1241-00).

PK69959

Initial problem description

In FM/DB2, message ″FM/DB2 could not access catalog table SYSIBM.LOCATIONS, LUNAMES.″ is displayed when:

- a. Access to one or more of the DB2 catalog tables defined in the SYSDDF databases has not been granted to PUBLIC, or to FM/DB2 users.
- b. An attempt is made to access a remote DB2 server, or display a selection list showing the available remote servers.

Outline of solution

File Manager DB2 component has been enhanced to allow operation when access to DB2 catalog tables defined in the SYSDDF database is not available. In this situation, the location selection list function is not available.

Documentation impact

This APAR requires changes to be made to the User's Guide and Reference for DB2 Data (SC19-1240-00).

Changes to the User's Guide and Reference for DB2 Data

Chapter 2, ″Getting started with FM/DB2″

In the section, ″Using the FM/DB2 interface″, subsection ″Specifying a DB2 Object Name″, in the description for **Location**, after the existing text:

″If you specify an asterisk on its own or a pattern, FM/DB2 displays a selection list showing the locations that match the input specification. To select the location you want, type S against the required entry and press Enter.″

add this text:

"The location selection list is not available unless access to certain DB2 catalog tables has been granted to FM/DB2 users. When the access is not available, you can still access a remote DB2 system by specifying the exact location name."

PK72083

Initial problem description

In IEB/IBB batch the field references can only be specified on the:

- 1. WHERE clause of the DELIMS, GETIMS, ISRTIMS, REPLIMS and VIEWPOSN functions for the first layout of a segment.
- 2. SET clause of the ISRTIMS function for the first layout of a segment.
- 3. SET clause of the REPLIMS function for the current layout associated with the segment being replaced.

In IEB/IBB batch the VIEWIMS function does not allow:

- 4. Segment layouts to be deselected.
- 5. Selection criteria to be specified for individual segment layouts. The criteria will be placed on the first layout for a segment.

Miscellaneous issues:

- 6. The SETRC IEB/IBB batch function does not allow a value of zero to be specified.
- 7. In browse or edit, after accessing a terminal MSDB data base and attempting to access a HALDB, DEDB, full function or non-terminal MSDB database using a dynamic PSB, FM/IMS receive message 'DFS0919I KEYLEN SPECIFICATION TOO SMALL. KEY LEN SHOULD BE --xxxxx BYTES'. FM/IMS then abends U919.

Outline of solution

FM/IMS has been modified to:

- Allow the user to qualify the field references or field names names with either a relative layout number or layout name on the:
  - 1. WHERE clause for the DELIMS, GETIMS,ISRTIMS, REPLIMS and VIEWPOSN IEB/IBB batch functions.
  - 2. SET clause for the ISRTIMS IEB function.
  - 3. SET clause for the REPLIMS IEB function.
- 4. Allow the specification of the layouts that are selected for a segment when creating the view using the VIEWIMS IEB/IBB batch function.
- 5. Allow the specification of the selection criteria for individual segment layouts when creating the view using the VIEWIMS IEB/IBB batch function.
- 6. Allow a value of '0' to be specified for the SETRC IEB/IBB batch function.
- 7. Ensure the flag which specifies a database is a terminal MSDB is switched off and hence the dynamic PSB which is generated has the correct keylen specified.

Documentation impact

This APAR requires changes to be made to:

- User's Guide and Reference for IMS (SC19-1241-00)

Changes to the User's Guide and Reference for IMS

Chapter 11, ″Batch Reference″, subsection ″Batch Edit (IEB)″, ″Retrieval using a view″

Replace these syntax diagrams and the description of parameters that follow each diagram:

Syntax

```
►►──ISRTIMS──(─SEGMENT=inssegm──────────────────────────────────────────────►

►─┬──────────────────────────────────────────────────────────────┬─)──────►◄
  │            ┌──────────────────────────────────────────────┐  │
  │    (1)     ▼                                       (2)    │  │
  └─SET──────────#─┬─┬─────────────┬─field_ref=field_value──┬─┴──┘
                   │ └─layout_ref.─┘                        │
                   └─┬──────────────────┬─field_name=field_value─┘
                     │ ┌──────────────┐ │
                     │ ▼  (3)         │ │
                     └────qualifier──.─┴─┘
```

Notes:

1 To use *field_ref* or *field_name*, you must have already loaded a view by preceding this function call with a call to VIEWIMS.

2 Specification of a *field_ref* without qualification of a *layout_ref* results in the *field_ref* being associated with the first layout for the *inssegm* segment.

3 To fully qualify a field_name, the layout name must be the first *qualifier* specified. If the layout name is not specified, FM/IMS searches all the layouts for the *inssegm* segment in relative order until the group qualifiers and *field_name* are found. If not found, an error message is displayed and processing terminates.

field_name
A field name.

field_ref
A field reference number of the field within the layout. This can be obtained by printing the view/template using the VIEWIMS DESCRIBE clause or browsing the view/template online.

field_value
The value of the field specified by *field_ref* or *field_name*.

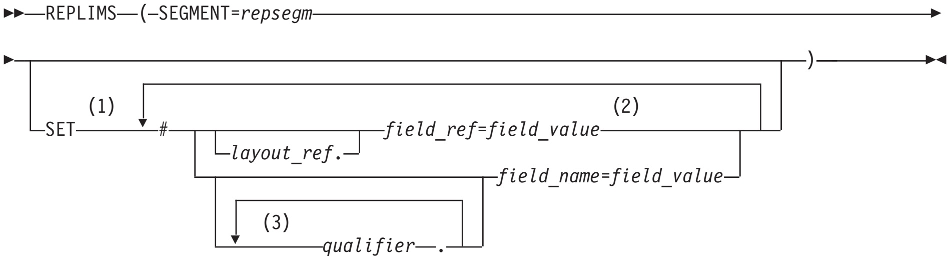
inssegm
The name of the segment you want to insert. This must be a child of the current segment.

layout_ref
The relative number of layout within a segment. This can be obtained by printing the view/template using the VIEWIMS DESCRIBE clause.

qualifier
One or more qualifiers that identify the field name (*field_name*). The first qualifier is the segment layout name. Subsequent qualifiers (if any) are the group fields used to identify the specified field name. Omitting the layout name or group fields causes FM/IMS to search the segment layouts for the specified qualifiers and field name.

Note: The layout names and group fields can be obtained by printing the view/template using the VIEWIMS DESCRIBE clause or browsing the view/template online.

Syntax



Notes:

1 To use *field_ref* or *field_name*, you must have already loaded a view by preceding this function call with a call to VIEWIMS.

2 Specification of a *field_ref* without qualification of a *layout_ref* results in the *field_ref* being associated with the current layout for the *repsegm* segment.

3 To fully qualify a field name, the layout name must be the first *qualifier* specified. If the layout name is not specified, FM/IMS searches the current the layout for the *repsegm* segment being replaced. If the group qualifiers and field name are not found, FM/IMS searches all the layouts for the *repsegm* segment in relative order until the group qualifiers and field name are found. If not found, an error message is displayed and processing terminates.

field_name
A field name.

field_ref
A field reference number of the field within the layout. This can be obtained by printing the view/template using the VIEWIMS DESCRIBE clause or browsing the view/template online.

field_value
The value of the field specified by *field_ref* or *field_name*.

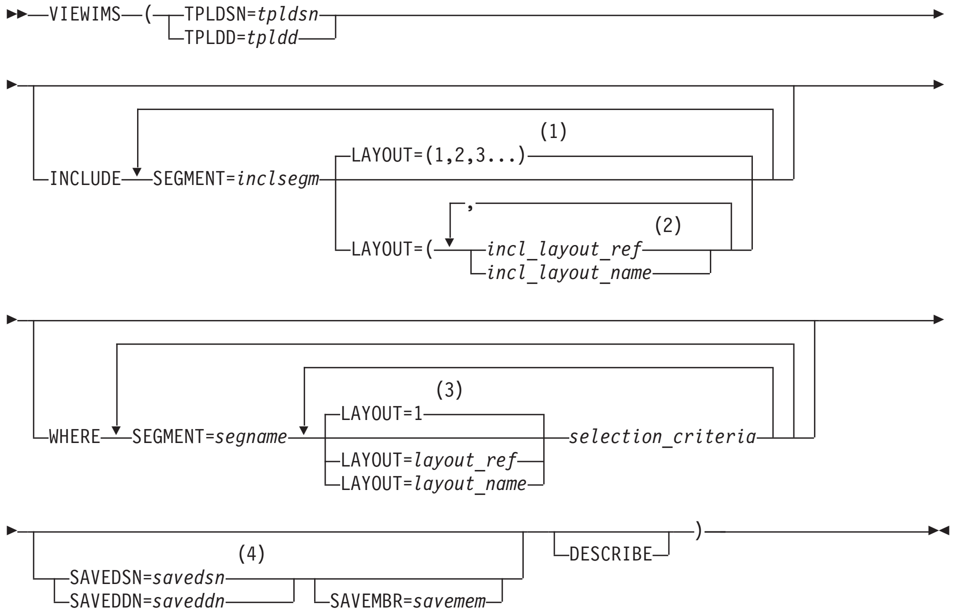
layout_ref
The relative number of layout within a segment. This can be obtained by printing the view/template using the VIEWIMS DESCRIBE clause.

qualifier
One or more qualifiers that identify the field name (*field_name*). The first qualifier is the segment layout name. Subsequent qualifiers (if any) are the group fields used to identify the specified field name. Omitting the layout name or group fields causes FM/IMS to search the current *repsegm* segment layout for the specified qualifiers and field name. If not found, then all *repsegm* segment layouts are searched for the specified qualifiers and field name.

Note: The layout names and group fields can be obtained by printing the view/template using the VIEWIMS DESCRIBE clause or browsing the view/template online.

repsegm
The name of the segment you want to replace.

Syntax (to create a new view from an existing template)



Notes:

1 If no layout parameter is specified, all layouts for *inclsegm* are selected.

2 If only one *incl_layout_ref* or *incl_layout_name* is specified, the parentheses are not required.

3 If no layout parameter is specified, then the selection criteria applies to the first layout for the *segname* segment.

4 The old format of SAVE is still supported (`SAVE DSN=`*`savedsn`* or `SAVE DDN=`*`saveddn`* with, optionally, `MBR=`*`savembr`*).

If you specify the DESCRIBE parameter, then VIEWIMS prints a report showing the structure of the view or template for each segment in the database.

incl_layout_name
The name of the *inclsegm* segment layout to be selected in the view. You can obtain this by printing the view/template with the VIEWIMS DESCRIBE clause or browsing the view/template online.

incl_layout_ref
The relative number of the layout within the *inclsegm* segment that is selected in the view. You can obtain this by printing the view/template with the VIEWIMS DESCRIBE clause.

inclsegm
The name of a segment to be included in the view.

If you call the GETIMS function with the USING_VIEW parameter, then only the included in the view is retrieved.

However, if you omit the INCLUDE clause entirely, then all segments are included in the view.

layout_name
The name of the layout within the *segname* segment to which the selection criteria are to apply. This can be obtained by printing the view/template using the VIEWIMS clause or browsing the view/template online.

layout_ref
The relative number of the layout within a segment to which the selection criteria are to apply. This can be obtained by printing the view/template using the VIEWIMS DESCRIBE clause.

saveddn
The DDNAME containing the data set into which the view is to be saved. If the data set is a PDS, you must also specify the SAVEMBR=*savemem* parameter.

savedsn
The name of the data set into which the view is to be saved. If the data set is a PDS, you must also specify the MBR=*savemem* parameter.

savemem
The member of the PDS specified by the DSN, SAVEDSN, DDN, or SAVEDDN parameter into which the view is to be saved.

segname
The name of a segment for which you want to specify field selection criteria.

selection_criteria
Field selection criteria for the *segname* segment. Determines which segment occurrences are retrieved by calls to the GETIMS function with the USING_VIEW parameter.

tpldd The DDNAME containing the template data set that is used as a basis to create a view. The template for the DBD must have already been created online in FM/IMS.

tpldsn The name of the data set containing the template that is used as a basis to create a view. The template for the DBD must have already been created on-line in FM/IMS.

viewdd The DDNAME containing one of the following:

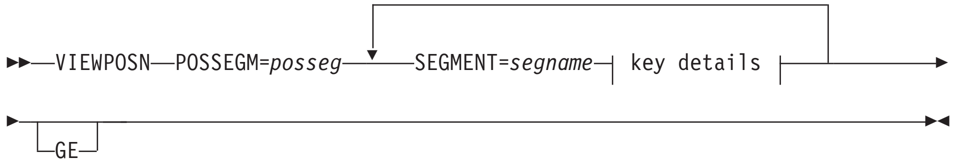
- The sequential data set containing the view.
- The PDS containing the view. You then specify the member name in *viewmem*.

viewdsn
One of:

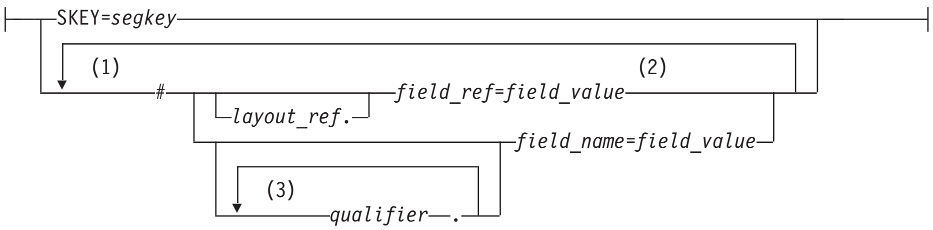
- The name of the sequential data set containing the view.
- The name of the PDS containing the view. (You then specify the member name in *viewmem*.)

viewmem
The member name of the view.

Syntax



key details:



Notes:

1 To use *field_ref* or *field_name*, you must have already loaded a view by preceding this function call with a call to VIEWIMS.

2 Specification of a *field_ref* without qualification of a *layout_ref* results in the *field_ref* being associated with the first layout for the *segname* segment.

3 To fully qualify a field_name, the layout name must be the first *qualifier* specified. If the layout name is not specified, FM/IMS searches all the layouts for the *segname* segment in relative order until the group qualifiers and *field_name* are found. If not found, an error message is displayed and processing terminates.

field_name
A field name.

field_ref
A field reference number of the field within the layout. This can be obtained by printing the view/template using the VIEWIMS DESCRIBE clause or browsing the view/template online.

field_value
The value of the field specified by *field_ref* or *field_name*.

layout_ref
The relative number of layout within a segment. This can be obtained by printing the view/template using the VIEWIMS DESCRIBE clause.

posseg The name of the segment you want to position on.

qualifier
One or more qualifiers that identify the field name (*field_name*). The first qualifier is the segment layout name. Subsequent qualifiers (if any) are the

group fields used to identify the specified field name. Omitting the layout name or group fields causes FM/IMS to search the segment layouts for the specified qualifiers and field name.

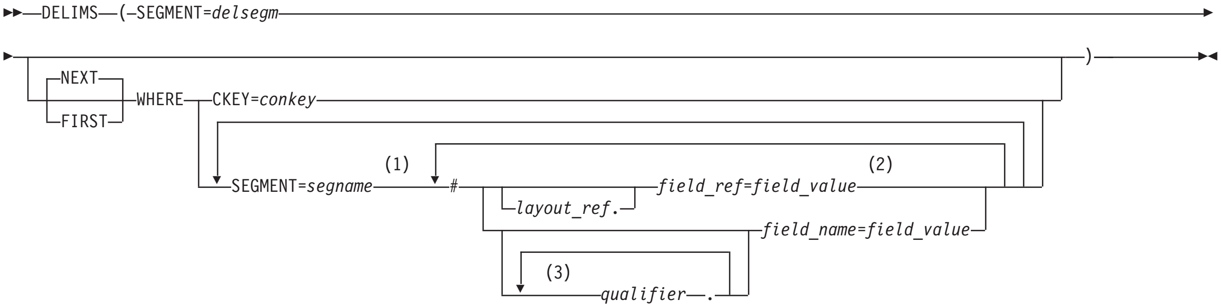
Note: The layout names and group fields can be obtained by printing the view/template using the VIEWIMS DESCRIBE clause or browsing the view/template online.

segname
The name of a segment in the hierarchical path of the segment you want to position on.

Chapter 11, ″Batch Reference″, subsection ″Batch Edit (IEB)″, ″Retrieval without a view″

Replace these syntax diagrams and the description of parameters that follow each diagram:

Syntax



Notes:

1 To use *field_ref* or *field_name*, you must have already loaded a view by preceding this function call with a call to VIEWIMS.

2 Specification of a *field_ref* without qualification of a *layout_ref* results in the *field_ref* being associated with the first layout for the *delsegm* segment.

3 To fully qualify a field_name, the layout name must be the first *qualifier* specified. If the layout name is not specified, FM/IMS searches all the layouts for the *delsegm* segment in relative order until the group qualifiers and *field_name* are found. If not found, an error message is displayed and processing terminates.

conkey The concatenated key of the segment you want to delete.

If you specify the concatenated key of a segment that you have already deleted, then DELIMS produces a nonzero return code, and does not delete any segment.

delsegm
The name of the segment you want to delete.

field_name
The name of a key field in the *segname* segment, as specified in the template or view.

field_ref
The reference number of a key field in the *segname* segment, as specified in

the template or view. This can be obtained by printing the view/template using the VIEWIMS DESCRIBE clause or browsing the view/template online.

field_value
The value of the key field specified by *field_ref* or *field_name*.

FIRST Deletes the first segment in the database that satisfies the WHERE clause.

layout_ref
The relative number of layout within *segname* segment. This can be obtained by printing the view/template using the VIEWIMS DESCRIBE clause.

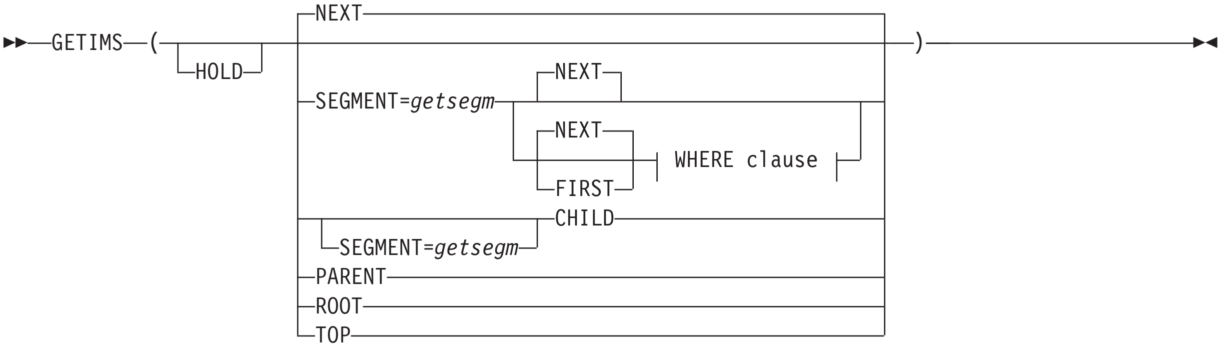
NEXT Deletes the next segment in the database after the current segment that satisfies the WHERE clause.

qualifier
One or more qualifiers that identify the field name (*field_name*). The first qualifier is the segment layout name. Subsequent qualifiers (if any) are the group fields used to identify the specified field name. Omitting the layout name or group fields causes FM/IMS to search the segment layouts for the specified qualifiers and field name.

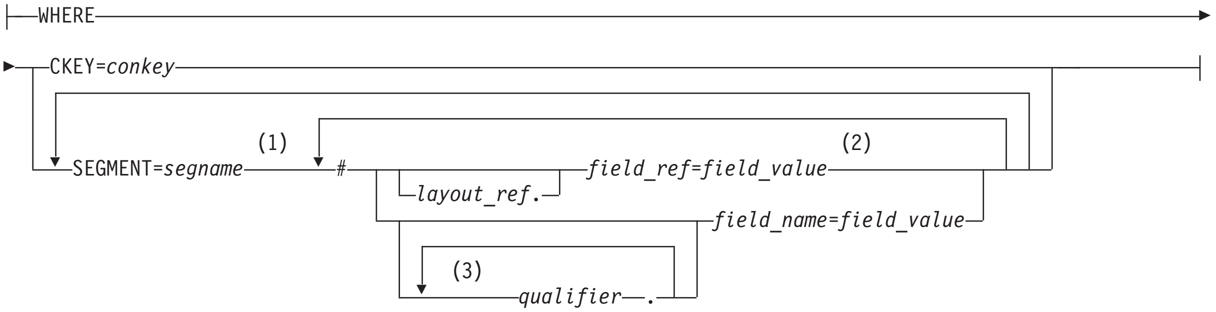
Note: The layout names and group fields can be obtained by printing the view/template using the VIEWIMS DESCRIBE clause or browsing the view/template online.

segname
The name of a segment in the hierarchical path of the segment that you want to delete.

Syntax (without a view)



WHERE clause:



Notes:

1 To use *field_ref* or *field_name*, you must have already loaded a view by preceding this function call with a call to VIEWIMS.

2 Specification of a *field_ref* without qualification of a *layout_ref* results in the *field_ref* being associated with the first layout for the *segname* segment.

3 To fully qualify a field_name, the layout name must be the first *qualifier* specified. If the layout name is not specified, FM/IMS searches all the layouts for the *segname* segment in relative order until the group qualifiers and *field_name* are found. If not found, an error message is displayed and processing terminates.

CHILD

Retrieves the next segment in the hierarchical path of the current segment. If you also specify SEGMENT=*getsegm*, then GETIMS retrieves the next segment of that name in the hierarchical path of the current segment.

conkey The concatenated key of the segment you want to retrieve.

field_name

The name of a key field in the segname segment, as specified in the template or view. This can be obtained by printing the view/template using the VIEWIMS DESCRIBE clause or browsing the view/template online.

field_ref

A field reference number of the field within the layout. This can be obtained by printing the view/template using the VIEWIMS DESCRIBE clause or browsing the view/template online.

field_value
The value of the key field specified by *field_ref* or *field_name*.

FIRST Retrieves the first segment in the database that satisfies the WHERE clause.

getsegm
The name of the segment you want to retrieve. If you also specify CHILD, then GETIMS retrieves the next hierarchically dependent segment with this name. Otherwise, the default value is NEXT, and GETIMS retrieves the next segment with this name, regardless of hierarchy.

HOLD
Retrieves the segment with a hold. Code this parameter when the GETIMS call is followed by a REPLIMS call without a WHERE clause, or a DELIMS call without a WHERE clause.

If the segment or one of its parents has a nonunique key or no key, this parameter is mandatory. Without it, the REPLIMS call or DELIMS call fails. The REPLIMS call or DELIMS call will not fail if the segment and its parents have unique keys but, if you do code it, your EXEC executes less DL/I calls and runs more efficiently.

layout_ref
The relative number of layout within *segname* segment. This can be obtained by printing the view/template using the VIEWIMS DESCRIBE clause.

NEXT Retrieves the next segment in the database after the current segment.

PARENT
Retrieves the parent segment of the current segment.

If the current segment is a root segment, then GETIMS retrieves that same segment. To avoid looping in this situation, you can use the FMSEGNM REXX variable to check the name of the current segment.

qualifier
One or more qualifiers that identify the field name (*field_name*). The first qualifier is the segment layout name. Subsequent qualifiers (if any) are the group fields used to identify the specified field name. Omitting the layout name or group fields causes FM/IMS to search the segment layouts for the specified qualifiers and field name.

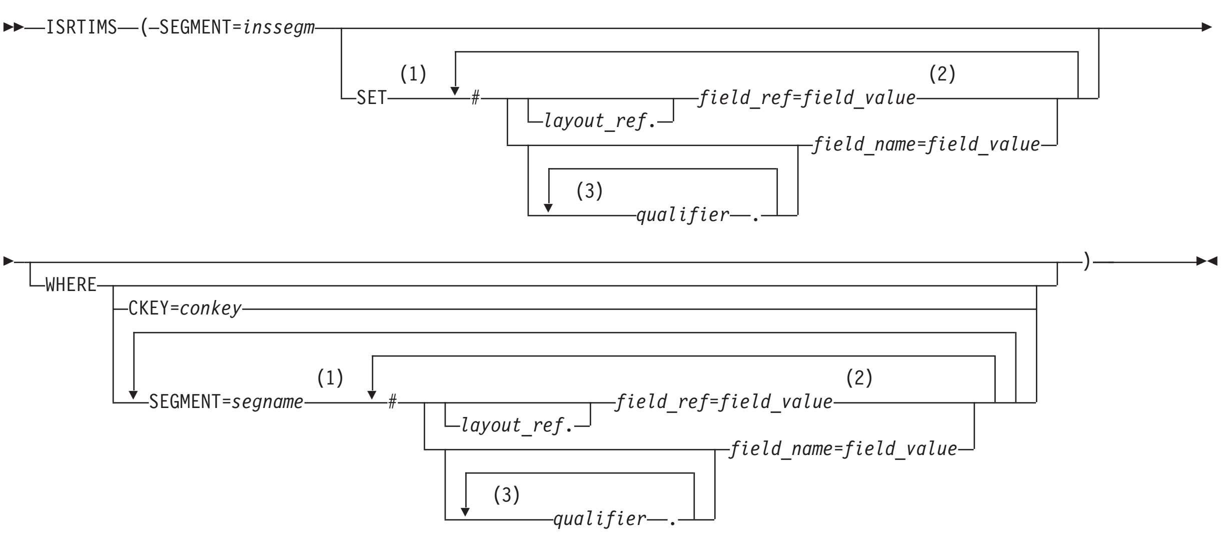
Note: The layout names and group fields can be obtained by printing the view/template using the VIEWIMS DESCRIBE clause or browsing the view/template online.

ROOT
Retrieves the root segment in the hierarchical path of the current segment.

segname
The name of a segment in the hierarchical path of the segment that you want to retrieve.

TOP Retrieves the first root segment in the database.

Syntax



Notes:

1 To use *field_ref* or *field_name*, you must have already loaded a view by preceding this function call with a call to VIEWIMS.

2 Specification of a *field_ref* without qualification of a *layout_ref* results in the *field_ref* being associated with the first layout for the *inssegm* segment if specified on the SET clause, or the *segname* segment if specified on the WHERE clause.

3 To fully qualify a field_name, the layout name must be the first *qualifier* specified. If the layout name is not specified, FM/IMS searches all the layouts for the segment in relative order until the group qualifiers and *field_name* are found. If not found, an error message is displayed and processing terminates.

conkey The concatenated key of the parent of the segment you want to insert.

field_name
A field name.

field_ref
A field reference number of the field within the layout. This can be obtained by printing the view/template using the VIEWIMS DESCRIBE clause or browsing the view/template online.

field_value
The value of the field specified by *field_ref* or *field_name*.

inssegm
The name of the segment you want to insert. This must be a child of the segment that you specify in the WHERE clause, or, if you omit the WHERE clause, a child of the current segment.

layout_ref
The relative number of layout within *segname* segment. This can be obtained by printing the view/template using the VIEWIMS DESCRIBE clause.

qualifier

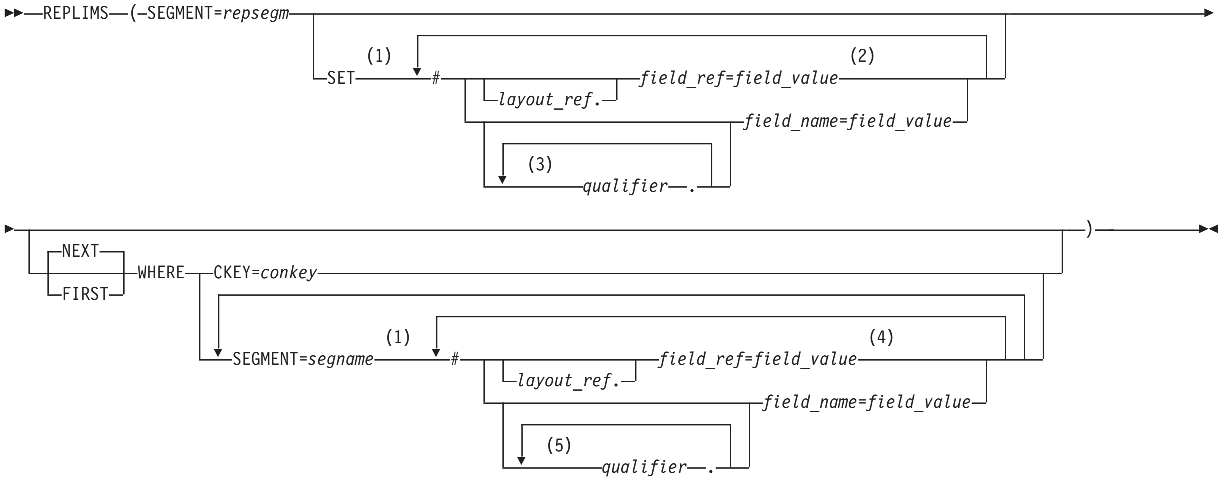
One or more qualifiers that identify the field name (*field_name*). The first qualifier is the segment layout name. Subsequent qualifiers (if any) are the group fields used to identify the specified field name. Omitting the layout name or group fields causes FM/IMS to search the segment layouts for the specified qualifiers and field name.

Note: The layout names and group fields can be obtained by printing the view/template using the VIEWIMS DESCRIBE clause or browsing the view/template online.

segname

The name of a segment in the hierarchical path of the parent of the segment that you want to insert.

Syntax



Notes:

1 To use *field_ref* or *field_name*, you must have already loaded a view by preceding this function call with a call to VIEWIMS.

2 Specification of a *field_ref* without qualification of a *layout_ref* results in the *field_ref* being associated with the current layout for the *repsegm* segment that is being replaced

3 To fully qualify a field name, the layout name must be the first *qualifier* specified. If the layout name is not specified, FM/IMS searches the current the layout for the *repsegm* segment being replaced. If the group qualifiers and field name are not found, FM/IMS searches all the layouts for the *repsegm* segment in relative order until the group qualifiers and field name are found. If not found, an error message is displayed and processing terminates.

4 Specification of a *field_ref* without qualification of a *layout_ref* results in the *field_ref* being associated with the first layout for the *segname* segment.

5 To fully qualify a field name, the layout name must be the first *qualifier* specified. If the layout name is not specified, FM/IMS searches all the layouts for the *segname* segment in relative order until the group qualifiers and field name are found. If not found, an error message is displayed and processing terminates.

conkey The concatenated key of the segment you want to update.

field_name A field name.

field_ref
A field reference number of the field within the layout. This can be obtained by printing the view/template using the VIEWIMS DESCRIBE clause or browsing the view/template online.

field_value
The value of the field specified by *field_ref* or *field_name*.

FIRST
Replaces the first segment in the database that satisfies the WHERE clause.

layout_ref
The relative number of layout within *segname* segment. This can be obtained by printing the view/template using the VIEWIMS DESCRIBE clause.

NEXT
Replaces the next segment in the database after the current segment that satisfies the WHERE clause.

qualifier
One or more qualifiers that identify the field name (*field_name*). The first qualifier is the segment layout name. Subsequent qualifiers (if any) are the group fields used to identify the specified field name.

Omitting the layout name or group fields causes FM/IMS:

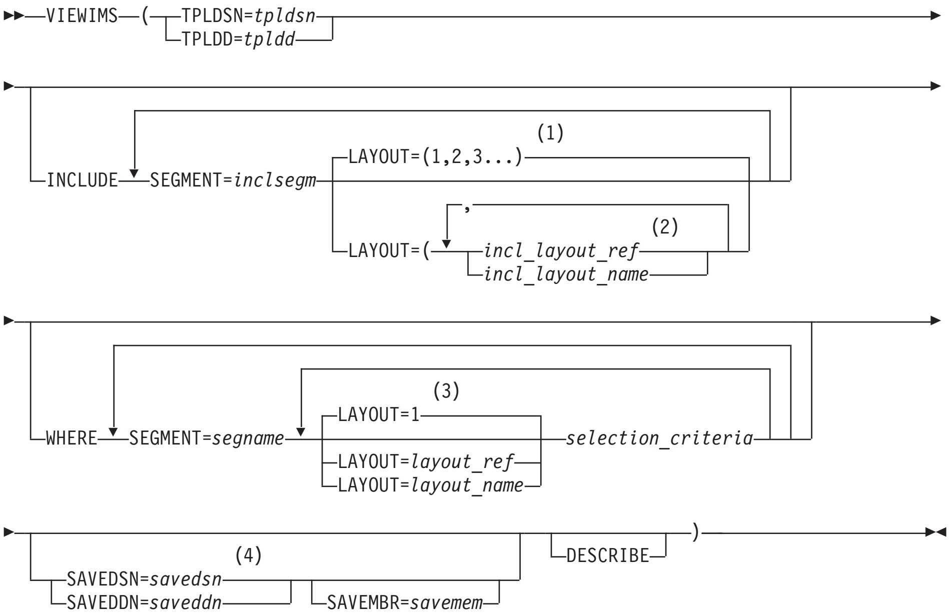
- For the SET clause, to search the current *repsegm* segment layout for the specified qualifiers and field name. If not found, then all *repsegm* segment layouts are searched for the specified qualifiers and field name.
- For the WHERE clause, to search all the *segname* segment layouts for the specified qualifiers and field name.

Note: The layout names and group fields can be obtained by printing the view/template using the VIEWIMS DESCRIBE clause or browsing the view/template online.

repsegm
The name of the segment you want to update.

segname
The name of a segment in the hierarchical path of the segment that you want to update.

Syntax (to create a new view from an existing template)



```
►►──VIEWIMS──(──┬─TPLDSN=tpldsn─┬──────────────────────────────────────────►
                └─TPLDD=tpldd───┘

►──┬────────────────────────────────────────────────────────────────────┬──►
   │          ┌──────────────────────────────────────────────────────┐  │
   │          ▼                             (1)                      │  │
   └─INCLUDE────SEGMENT=inclsegm─┬─LAYOUT=(1,2,3...)──────────────┬──┘──┘
                                 │         ┌─,──────────────────┐  │
                                 │         ▼               (2)  │  │
                                 └─LAYOUT=(──┬─incl_layout_ref──┬┴──┘
                                             └─incl_layout_name─┘

►──┬────────────────────────────────────────────────────────────────────┬──►
   │        ┌────────────────────────────────────────────────────────┐  │
   │        │                    ┌────────────────────────────────┐  │  │
   │        ▼                    ▼        (3)                     │  │  │
   └─WHERE────SEGMENT=segname────┬─LAYOUT=1─────────┬─selection_criteria─┴──┴──┘
                                 ├─LAYOUT=layout_ref──┤
                                 └─LAYOUT=layout_name─┘

►──┬──────────────────────────────────────────────┬──┬──────────┬──)──────►◄
   │                  (4)                          │  └─DESCRIBE─┘
   └─┬─SAVEDSN=savedsn─┬──┬──────────────────┬─────┘
     └─SAVEDDN=saveddn─┘  └─SAVEMBR=savemem──┘
```

Notes:

1 If no layout parameter is specified, all layouts for *inclsegm* are selected.

2 If only one *incl_layout_ref* or *incl_layout_name* is specified, the parentheses are not required.

3 If no layout parameter is specified, then the selection criteria applies to the first layout for the *segname* segment.

4 The old format of SAVE is still supported (`SAVE DSN=`*`savedsn`* or `SAVE DDN=`*`saveddn`* with, optionally, `MBR=`*`savembr`*).

If you specify the DESCRIBE parameter, then VIEWIMS prints a report showing the structure of the view or template for each segment in the database.

incl_layout_name
The name of the *inclsegm* segment layout to be selected in the view. You can obtain this by printing the view/template with the VIEWIMS DESCRIBE clause or browsing the view/template online.

incl_layout_ref
The relative number of the layout within the *inclsegm* segment that is selected in the view. You can obtain this by printing the view/template with the VIEWIMS DESCRIBE clause.

inclsegm
The name of a segment to be included in the view.

If you call the GETIMS function with the USING_VIEW parameter, then only the included in the view is retrieved.

However, if you omit the INCLUDE clause entirely, then all segments are included in the view.

layout_name
The name of the layout within the *segname* segment to which the selection criteria are to apply. This can be obtained by printing the view/template using the VIEWIMS clause or browsing the view/template online.

layout_ref
The relative number of the layout within a segment to which the selection criteria are to apply. This can be obtained by printing the view/template using the VIEWIMS DESCRIBE clause.

saveddn
The DDNAME containing the data set into which the view is to be saved. If the data set is a PDS, you must also specify the SAVEMBR=*savemem* parameter.

savedsn
The name of the data set into which the view is to be saved. If the data set is a PDS, you must also specify the MBR=*savemem* parameter.

savemem
The member of the PDS specified by the DSN, SAVEDSN, DDN, or SAVEDDN parameter into which the view is to be saved.

segname
The name of a segment for which you want to specify field selection criteria.

selection_criteria
Field selection criteria for the *segname* segment. Determines which segment occurrences are retrieved by calls to the GETIMS function with the USING_VIEW parameter.

tpldd
The DDNAME containing the template data set that is used as a basis to create a view. The template for the DBD must have already been created on-line in FM/IMS.

tpldsn
The name of the data set containing the template that is used as a basis to create a view. The template for the DBD must have already been created on-line in FM/IMS.

viewdd
The DDNAME containing one of the following:

- The sequential data set containing the view.
- The PDS containing the view. You then specify the member name in *viewmem*.

viewdsn
One of:

- The name of the sequential data set containing the view.
- The name of the PDS containing the view. (You then specify the member name in *viewmem*.)

viewmem
The member name of the view.

UK41178, UK41185, UK41197, UK41198, UK41200, UK41201, UK41204, UK41210, UK41222, UK41226, UK41227, UK41228

Release Date: **7 November 2008**

This set of PTFs contains the following APAR fixes:

APAR #	APAR Abstract	Doc Impact
PK68099	File Manager - various problems found internally Q3 2008	User's Guide and Reference (SC19-1239-00)
PK70109	File Manager generated wrong JCL using FCH function.	User's Guide and Reference (SC19-1239-00)
PK70113	ABEND0C4 occurred at FMNMAIN during table edit.	None.
PK73481	ABEND0C4 in FMN2CATQ at offset X'527C' processing a table with default values.	None.

PK68099

Initial problem description

1. An improper message may be displayed while inserting new IMS segments.

2. Load module compare prints incorrect output for DATE and MODE options.

3. Base audit report does not contain any template information.

4. 'Additional print options' panel not displayed when option selected on 'Print Temporary Storage Entry' panel.

5. When PDS and nonPDS datasets are included as input to batch DSC, DSP or FCH functions job may terminate with RC=16 and no message to indicate what the error is.

6. File Manager will abend S978 when using the DSC utility to generate a batch job using a member selection list to identify the member to be copied.

7. If a copybook is used when comparing data sets and the copybook contains multiple record definitions with no selection criteria only the first record in the copybook is being used when comparing record lengths for a match.

8. When browsing the compile listing after an error under FM/CICS the the message 'Value out of range' is produced and scrolling is not possible.

9. Template edit is not detecting a change when a scrambling value list has been deleted.

10. Workbench copybook member name being incorrectly updated with member name provided in data set name field after a CC command.

11. ABENDS978-04 running template build utility with batch execution.

12. When a start key is entered on the Data Set Print utility entry panel for a NONVSAM data set the message ’Must be record number’ is displayed on the cursor is position on the command line.

13. The pulldown option in not protected when in the Temporary and Output Data Set Allocation option panels.

14. The Print Audit Trail Report may abends0c4 when an incomplete audit log data set is provided as input.

15. Select command requires extra enter when selecting single member.

16. IMS segment length changes via the length field are not honored during an insert or repeat.

17. ABENDSOC4 running a procedure with TFLD function and no template.

18. The member name field is not being cleared on the ’Advanced Member Selection’ panel when it is cleared on the previous panel.

19. File Manager CICS may abends0c4 when first started.

20. An expanding change for a variable length record may incorrectly retain trailing blanks.

21. The ’sorted by’ field on the ’Display VTOC’ panels may be incorrectly interpreted when abbreviated.

22.If a column range is specified when using the Find/Change utility to change JCL the from column is being ignored.

23.The PARM parameter may be split incorrectly causing invalid JCL to be generated when a JCL statement is changed using the Find/Change utility.

24.When using the FCH utility and the length of ’change to’ value is longer that the length of the ’change from’ value and the JCL option has been selected changes will not be made to statements other than JCL statements.

25.When using the FCH function in batch with multiple Change commands supplied and these commands include a column range the column range may not be honored.

26.When using the FCH utility and the length of the ’change to’ value is longer than the length of the ’change from’ value and the JCL option has been selected, changes will not be made to statements other than JCL statements.

27.When using the edit or browse function and the ’Edit Template’ option has been selected and the template contains long names File Manager may abendS0C4.

28.When using the Find/Change utility to change JCL and a trailing space is specified in the Change command no match will be found because trailing spaces are not included in the logical JCL statement built by File Manager. The same problem would occur for the Find command.

29.The Find/Change utility does not recognize the VCONTEXT command when the JCL option has been selected.

30.When using a formatted compare numeric fields which have the same attributes but do not contain valid numeric data are being flagged as invalid and are not compared.

31.File Manager is not including invalid data fields as unmatched when the ’Changed data only’ only option has been selected.

32.File Manager may ABEND0C4 when comparing data sets using templates and the old template contains more record definitions than the new template(PFM03312).

33.Unmatched numeric fields are not being reported when comparing data sets using a template.

34.XML tag </RECORD> in wrong position.

35.In FM/DB2 Object List utility, when listing plans and issuing the prefix command RBI against a plan and entering a ’?’ in the PKLIST field an ISPF Dialog Error occurs.

36.The fields CACHE and ACQUIRE are incorrectly populated in the DB2 ’Bind Application Plan’ panel.

37.Compare utility may abends0c4 when comparing multiple members of PDSs or PDSEs.

38.Compare utility may abends978 when the ’Batch execution’ option has been selected and the old data set is a PDS or PDSE and a member mask is specified.

Outline of solution

File Manager has been updated such that:

1. Navigation is not attempted during an IMS insert operation. 2. DATE and MODE in the CSECTC or LMODC parameters now produce the correct output.

3. Template information, if it exists, is now printed on first page of the Detailed Statistics and on the Summary Statistics page.

4. The ’Additional print options’ panel will be displayed when the option is selected on the ’Print Temporary Storage Entry’ panel.

5. When processing a concatenated data set File Manager will check to see that the data set organisations are the same.

6. File Manager will not abend when requested to generate a batch job when using the DSC utility with a member selection list.

7. File Manager will check all records in the copybook when selecting records.

8. The insert record length will not be displayed as zero under CICS when a browse listing function is running.

9. The removal of value lists will be detected as a template change when editing the scrambling attributes.

10.Fix copybook member name updating to correctly check data set name for member before updating.

11.Freemain for selection list has been corrected.

12.The File Manager utilities print, copy and compare will not accept a Start key value when processing a nonvsam data set.

13.The correct pulldown option will be protect when in the Base and DB2 Temporary and Output Data Set Allocation option panels.

14.File Manager will not abends0c4 when producing an Audit Trail Report from an incomplete audit log file.

15.When selecting a single member off a list hold processing will not apply.

16. The IMS segment header (for variable length IMS segments) will be correctly updated, honoring the length field update.

17.TFLD function corrected to check for template existance.

18.The member name field in the 'Advanced Member Selection' panel will be cleared if no member name was supplied on the previous panel.

19.File Manager CICS will not abends0c4 when initially started.

20.A record wide change command that potentially expands a variable length record with trailing blanks will consider the trailing blanks before setting the new record length.

21.The minimum length of some 'sorted by' values on the 'Display VTOC' panels has been increased to ensure the correct value is selected when an alternate value has been entered.

22.The from column will be taken into consideration when finding/changing non-JCL records when the JCL option has been selected.

23a.If a PARM statement is enclosed in quotes and there are multiple quoted parameters separated by a comma and the statement needs to be continued, the parameters after the comma will be continued on a second JCL record.

23b.If there is more than one keyword on the last record of a changed JCL statement and the record has been expanded and will no longer fit into one record, the record will be split on the last keyword.

24.File Manager has been updated in order to correctly calculate the to column for non JCL statements, when the JCL option is selected.

25.The column range supplied with batch Change commands will be honored.

26.File Manager has been updated in order to correctly calculate the to column for non JCL statements, when the JCL option is selected.

27.File Manager will no longer abends0c4 when editing a DB2 table containing containg a number of long names.

28.Trailing spaces will be included in the logical JCL statement against which the Find or Change parameter is compared. Note that this occurs only for the last statement of a (continued or single) JCL statement. Trailing spaces in other statements forming part of a continued JCL statement continue to be ignored (they are not part of the logical JCL statement).

29.The VCONTEXT command parameters will be taken into consideration when producing the Find/Change utility report and the LIST=LONG and JCL options have been specified.

30.When using a formatted compare numeric fields which have the same attributes will be compared as character strings. This prevents fields with the same underlying data from flagging the record as containing invalid data.

31.File Manager will include invalid data fields in the unmatched report when the 'Changed data only' only option has been selected. 32.The 01 level comparison loop has been corrected.

33.File Manager will report unmatched numeric fields when data sets are compared using a template.

34.XML output has been corrected.

35-36. File Manager DB2 component has been changed to correct the problems.

37.The Compare utility will no longer abends0c4 when comparing members in PDSs or PDSEs.

38.The Compare utility will no longer abends978 when comparing members in PDSs or PDSEs.

Documentation impact

This APAR requires changes to be made to the User's Guide and Reference (SC19-1239-00).

Changes to the User's Guide and Reference

Chapter 6, ″Managing data sets″

In the section ″Comparing data sets″, section ″Comparison output″, subsection ″Body section flags″, add the following to the last entry in this section:

″When arrays are being compared and the number of array elements in the old record is different to the number of array elements in the new record, the array elements which are not matched are reported as changed fields.″

In the section ″Comparing data sets″, subsection ″Comparing data sets using templates″, change the paragraph beginning with

″If a formatted comparison is performed then, by default...″

to read:

″If a formatted comparison is performed then, by default, the fields in the ″Old″ template are mapped to fields in the ″New″ template that have the same name. If a field appears in either the ″Old″ or ″New″ template and does not have a field with the same name in the other template it is not mapped and is not reported.

You can edit the field mapping in the ″New″ template, and either:

- Delete the mapping for a field (so that it is not compared), or
- Specify which field in the ″Old″ template is mapped to a field in the ″New″ template

You can use the template mapping reports to identify those fields which are, and are not, mapped.″

Chapter 13, ″Enhancing File Manager processing″

In the section ″Creating and supplying a procedure″, subsection ″Coding REXX procedure statements″, subsection ″Using internal (FASTREXX) processing″, add the following note:

Note: Default length for packed fields: If you omit the length for a function that refers to a packed field then the length for FASTREXX is calculated as the length of the first valid occurrance of that field on a record. For REXX processing the length is re-calculated for each record processed.

Chapter 14, ″Panels and fields″

In the section ″Compare Utility: Options panel″, replace the existing description for the **Always show SELECTed fields** option to:

Always show SELECTed fields

Specify / to show SELECTed fields in formatted comparison reports. The option is intended to allow you to ensure that certain fields that you have SELECTed are always printed, regardless of whether they are changed or not. This option has effect only if the **Show changed fields only** option has been selected and only if the SELECTed fields have been mapped.

PK70109

Initial problem description

1. If a column range is specified when using the Find/Change utility to change JCL the from column is being ignored.

2. The PARM parameter may be split incorrectly causing invalid JCL to be generated when a JCL statement is changed using the Find/Change utility.

3 .When using the FCH utility and the length of ′change to′ value is longer that the length of the ′change from′ value and the JCL option has been selected changes will not be made to statements other than JCL statements.

4 .When using the Find/Change utility to change JCL and a trailing space is specified in the Change command no match will be found because trailing spaces are not included in the logical JCL statement built by File Manager. The same problem would occur for the Find command.

5 .The Find/Change utility does not recognize the VCONTEXT command when the JCL option has been selected.

Outline of solution

1. The starting position to be used when finding/changing data records when the JCL option has been specified was not including the starting column if specified.

2. If the last record of a continued changed JCL statement will not fit in the space provided it is always being continued following the JCL continuation guidelines, which may lead to undesired results when substitution variables are present.

3. The wrong to column was being calculated. The routine FMNCHJCL appends a character to the JCL string and this character was being included in the to column calculation.

4. Trailing spaces are not included in the logical JCL statement against which the Find or Change parameter is compared.

5. The VCONTEXT command is ignored when THE JCL option has been selected.

Documentation impact

This APAR requires changes to be made to the User's Guide and Reference (SC19-1239-00).

Changes to the User's Guide and Reference

Chapter 6, ″Managing data sets″

In the section, ″Working with files or members containing JCL″, change the definition for *Parameter part* to:

Parameter part

Contains the contents of the first character after the *name part*, including any blanks after the last character of the *parameter part*.

UK38687, UK08097, UK08098, UK08099

Release Date: **15 August 2008**

This set of PTFs contains the following APAR fixes:

APAR #	APAR Abstract	Doc Impact
PK68093	File Manager usability improvements - criteria display, KSDS KEY command, REFD, and options processing.	User's Guide and Reference (SC19-1239-00)

PK68093

Initial problem description

Useability issues as follows:

1. Data Set Compare - processing options are not preserved across sessions.

2. Data Set Compare, Data Set Copy, Data Set Create and Data Set Print - Reference data sets are not available as per browse/view/edit functions.

3. Data Set View, Data Set Edit - No KEY command exists for allowing retrieval of last used key position.

4. Template Edit - Cannot see the structure when entering criteria by field.

Outline of solution

File Manager has been enhanced to provide the following.

1a. Batch execution option will now be preserved wherever it is set in the base product.

1b. Compare processing options with the exception of template edit will be preserved across sessions.

2. Reference data sets will be updated and selectable from functions Compare (option 3.11), Create (option 3.1) Copy (option 3.3) and Print (option 3.2).

3. A new command, KEY, for edit/view as an alternative way of navigating a KSDS file - allowing the use of the ISPF retrieve command to restore last key command.

4. A new command STR has been added to the template edit criteria displays to toggle the display between showing the structure or field levels and not showing them. The current setting is preserved across sessions.

Documentation impact

This APAR requires changes to be made to the User's Guide and Reference (SC19-1239-00).

Changes to the User's Guide and Reference

Chapter 3, ″Viewing and changing data sets″

For the section ″Using data set lists″:

- Move the section to become the first subsection in Chapter 6, ″Managing data sets″.
- Change the section heading to ″Working with data set lists″.
- Replace the first sentence with:

  The data set list facility provides an easy way to select frequently-used data sets within File Manager and is available for functions whose entry panel allows you to specify the primary data set and, optionally, a copybook or template.

  The data set list facility is available with these functions:
  – Browse
  – Compare
  – Copy
  – Create
  – Edit
  – Print
  – View
- Throughout the section, replace the term ″editor entry panel″ with ″function entry panel″.

For the section ″Starting an editor session without using templates″:

- After the last step, add:

  ″If your data set is a KSDS file, you can specify a starting position by issuing the KEY primary command.″

Chapter 14, ″Panels and fields″

- For the section ″Record Identification Criteria panel″, in the description for **Field Name**, add this paragraph:

  ″Use the STR primary command to toggle the displaying of the structure (level information) in the **Field Name** area of the display. The current setting is remembered for future sessions.″
- For the section ″Record Selection Criteria panel″, in the description for **Field Name**, add this paragraph:

  ″Use the STR primary command to toggle the displaying of the structure (level information) in the **Field Name** area of the display. The current setting is remembered for future sessions.″

Chapter 15, ″Primary commands″

Add this new command:

KEY primary command

You use the KEY primary command to specify a starting position when browsing, viewing, or editing a KSDS.

The KEY primary command positions to the record containing, or beginning with, the specified keyvalue. This is similar to the ability provided by the **Key** field on the editor entry panels.

Syntax

```
►►──KEY──keyvalue────────────────────────────────────────────────►◄
```

keyvalue

In a KSDS, File Manager positions to the record containing, or beginning with, this value.

To specify blanks or special characters, enclose the value in quotes. For example:

```
KEY '02 A'
```

For undisplayable characters, you can specify the value in hexadecimal. For example:

```
KEY 'F0F240C1'X
```

Availability

- Browse panel
- Edit panel
- View panel

Related tasks and examples

- ″Starting an editor session without using templates″

PK65892

Release Date: **8 August 2008**

APAR PK65892:

Change to the following document:

- User's Guide and Reference for DB2 Data (SC19-1240-00)

Replace Chapter 17 of this document with the following chapter:

UK38131, UK38132

Release Date: **21 July 2008**

This set of PTFs contains the following APAR fixes:

APAR #	APAR Abstract	Doc Impact
PK61281	GIM23911E when applying service.	Customization Guide (SC19-1238-01)

PK61281

Initial problem description

Two problems may occur when File Manager product is installed via Serverpac. 1.When applying maintenance (PTFs) for either FM/CICS or FM/IMS, SMP/E MSGGIM23901E Link edit failed may be produced for various modules including FMN1DLI, FMN3CICS. Link edit error MSGIEW2456E is produced for missing symbols including AIBTDLI,ASMTDLI,DFHEAI0,DFHEI1. 2.When applying maintenance (PTFs) affecting module FMNTRTBD, having previously deleted the FM/CICS component (JAC7816), MSGGIM23911E Linkedit failed for FMN3CICS is produced.

Outline of solution

1. File Manager IMS and CICS components have been repackaged so that they contain the required optional IMS and CICS modules as part of the File Manager product so that CALLLIBS are no longer required. 2. File Manager CICS has been changed to remove the reference to the module FMNTRTBD. It now contains its own version of this module. Note that this will only correct the problem if the FM/CICS FMID has not already been deleted. If the FM/CICS FMID has been deleted then the solution provided in the circumvention must be used. This is documented in APAR.

Documentation impact

This APAR requires changes to be made to the Customization Guide (SC19-1238-01).

Changes to the Customization Guide

Appendix E, ″Maintaining File Manager″

At the end of the section ″ServerPac customers″ add the following note:

Note: If you plan to delete FM/CICS, you **must** apply PTF UK38132 first.

UK37644, UK37645, UK37646, UK37647, UK37665, UK37666, UK37667, UK37674, UK37675, UK37680

Release Date: **30 June 2008**

PK61281

This set of PTFs contains the following APAR fixes:

APAR #	APAR Abstract	Doc Impact
PK60356	Load module processing does not work for identically named modules/libraries.	User's Guide and Reference (SC19-1239-00)
PK60577	FILE MANAGER does not allow enough HLQ variables for audit file names.	Customization Guide (SC19-1238-01)
PK64452	Security settings for function DBI are duplicated for both IMS and DB2.	(Same changes as for PK66242.)
PK64936	Input data set empty error for extended dataset.	None.
PK65359	RFIND command not working in File Manager DB2 object list utility.	None.
PK65978	System abend code 0C4 in DB2/FM.	None.
PK66102	Appending to criteria in an existing template changes original criteria.	None.
PK66116	File Manager various problems found internally Q2 2008.	None.
PK66189	File Manager enhancement to support an optional tally register on external functions	User's Guide and Reference (SC19-1239-00)
PK66193	FM/DB2 gen stmt: SQL statements are generated for all the indexes in the database instead of only one.	None.
PK66242	FM/DB2 update facility class does not prevent access.	Customization Guide (SC19-1238-01)
PK66295	File Manager ignore record length mismatch option ignored when set for data set compare (DSM).	None.
PK66305	Failed to change HLQ for auxiliary dataset.	None.
PK66436	FM/IMS IEB REPL function is not using the correct layout when multiple layouts are involved.	None.
PK66447	Trying to view database in FM/IMS with 2 dbd libs, results in error.	None.
PK66451	DB with some logical relationships (local to the db): extract + load of a record: some relationship are not loaded properly.	None.

APAR #	APAR Abstract	Doc Impact
PK66610	FM inplace edit of non-VSAM data set with pad on corrupts data.	None.
PK67407	File Manager MSGFMN0333I issued when processing unlike concatenated data sets.	None.

PK60356

Initial problem description

1. Message FMN1301I DDNEW/DSNNEW=DDOLD/DSNOLD is produced when the Old data set and New Data set names are the same and one of the data sets is uncatalogued.

2.When a Load module comparison is performed and no Load module criteria is specified, File Manager reporting gives the impression that there is no difference between unlike modules.

Outline of solution

1. File Manager has been updated to correct the problem.

2. File Manager has been updated to report when a Load Module Name comparison is performed.

Documentation impact

This APAR requires changes to be made to the User's Guide and Reference (SC19-1239-00).

Changes to the User's Guide and Reference

Chapter 6, ″Managing data sets″: In the section ″Comparing data sets″, subsection ″Comparing load modules″, in item ″**6. Load module criteria**: specify the load module properties you want to compare.″, add the sentence:

″If no criteria is selected, then only a comparison of the load module name is performed.″.

PK60577

Initial problem description

1. File Manager does not allow enough HLQ variables for audit data set names.

2. In FM/DB2 Explain utilities, the upgrade plan table option does not work.

Outline of solution

1. File Manager has been enhanced to allow greater user control over the format of File Manager audit log data set names. These changes apply to File Manager base, IMS and DB2 components.

2. File Manager DB2 component has been changed to correct the problem. SYSROUTE of PK60747.

Documentation impact

This APAR requires changes to be made to:

- Customization Guide (SC19-1238-01)

Changes to the Customization Guide

Chapter 5, ″Customizing the File Manager audit facility″: In the section, ″Recording an audit trail to a data set″, replace the text starting with ″The data set name will be of the format...″ with the following:

The format of the audit log data set name is determined by the setting of the AUDITHLQ parameter in the FMN0POPI definition in FMN0POPT. See ″AUDITHLQ″ on page 194 for more information about the AUDITHLQ option.

The following data set name formats may be generated

userid.FMNLOG.D*yymmdd*.T*hhmmss* (when AUDITHLQ= (blank))

audithlq.*userid*.FMNLOG.D*yymmdd*.T*hhmmss* (when AUDITHLQ=*audithlq*)

qual1.<*qual2*.><*qual3*.>D*yymmdd*.T*hhmmss* (when AUDITHLQ=*qual1*.<*qual2*.><*qual3*>)

where:

audithlq
Is any 1-8 character constant that is valid in the context of a data set name.

userid Is the userid creating the data set.

D*yymmdd*
Is the date of the activity.

T*hhmmss*
Is the time of the activity.

When AUDITHLQ contains one or more periods, the AUDITHLQ value is treated as a data set prefix, with one, two, or three levels. Each level of the prefix can be:

XXX Any 1-8 character constant that is valid in the context of a data set name.

&&PREFIX
Indicates that the user’s TSO prefix should be used. This will be null if TSO NOPREFIX is in effect and, after substitution, the appropriate level of the audit log data set name prefix will also be null.

&&USER
Indicates that the user’s logonid (ISPF system variable ZUSER, stored in the shared pool) should be used.

&&UID
Indicates that the user’s TSO prefix should be used, when the value is non-blank. When TSO NOPREFIX is in effect, the user’s TSO logonid (ISPF system variable ZUSER, stored in the shared pool) should be used.

&&FUNCOD
Indicates that the File Manager internal function code should be used. Specifying this parameter allows the File Manager function that generated the audit log data set to be included in the audit log data set name.

Chapter 13, ″Customizing the File Manager/DB2 audit facility″: In the section, ″Recording an audit trail to a data set″, replace the first paragraph and parameter descriptions with the following:

If you do not provide an SMF record type in your options macro, or do not specify SMF for audit logging in your FMN2SSDM macro, but audit logging is to take place, (either optional or required), then an audit trail will be written to a user data set.

The format of the audit log data set name is determined by the setting of the AUDITHLQ parameter in the FMN0POPI definition in FMN2POPT. See ″AUDITHLQ″ on page 194 for more information about the AUDITHLQ option.

The following data set name formats may be generated:

userid.FMN2AUD.<*ssid*>.D*yymmdd*.T*hhmmss* (when AUDITHLQ= (blank)

audithlq.FMN2AUD.<*ssid*>.D*yymmdd*.T*hhmmss* (when AUDITHLQ=*audithlq*)

qual1.<*qual2*.><*qual3*.>D*yymmdd*.T*hhmmss* (when AUDITHLQ=*qual1*.<*qual2*.><*qual3*>)

where:

audithlq
: Is any 1-8 character constant that is valid in the context of a data set name.

ssid
: Is the name of the currently connected (local) DB2 system.

userid
: Is the userid creating the data set.

D*yymmdd*
: Is the date of the activity.

T*hhmmss*
: Is the time of the activity.

When AUDITHLQ contains one or more periods, the AUDITHLQ value is treated as a data set prefix, with one, two, or three levels. Each level of the prefix can be:

XXX
: Any 1-8 character constant that is valid in the context of a data set name.

&&PREFIX
: Indicates that the user's TSO prefix should be used. This will be null if TSO NOPREFIX is in effect and, after substitution, the appropriate level of the audit log data set name prefix will also be null.

&&USER
: Indicates that the user's logonid (ISPF system variable ZUSER, stored in the shared pool) should be used.

&&UID
: Indicates that the user's TSO prefix should be used, when the value is non-blank. When TSO NOPREFIX is in effect, the user's TSO logonid (ISPF system variable ZUSER, stored in the shared pool) should be used.

&&FUNCOD
: Indicates that the File Manager internal function code should be used. Specifying this parameter allows the File Manager function that generated the audit log data set to be included in the audit log data set name.

&&SSID
: Indicates that the currently connected (local) DB2 subsystem name should be used.

Chapter 20, ″Customizing the File Manager/IMS audit facility″: In the section, ″Recording an audit trail to a data set″, replace the text starting with ″The data set name will be of the format...″ with the following:

The format of the audit log data set name is determined by the setting of the AUDITHLQ parameter in the FMN0POPI definition in FMN1POPT. See ″AUDITHLQ″ on page 194 for more information about the AUDITHLQ option.

The following data set name formats may be generated:

userid.IMSAUDIT.D*yymmdd*.T*hhmmss* (when AUDITHLQ= (blank)

audithlq.IMSAUDIT.D*yymmdd*.T*hhmmss* (when AUDITHLQ=*audithlq*)

qual1.<*qual2*.><*qual3*.>D*yymmdd*.T*hhmmss* (when AUDITHLQ=*qual1*.<*qual2*.><*qual3*>)

where:

audithlq
Is any 1-8 character constant that is valid in the context of a data set name.

userid Is the userid creating the data set.

D*yymmdd*
Is the date of the activity.

T*hhmmss*
Is the time of the activity.

When AUDITHLQ contains one or more periods, the AUDITHLQ value is treated as a data set prefix, with one, two, or three levels. Each level of the prefix can be:

XXX Any 1-8 character constant that is valid in the context of a data set name.

&&PREFIX
Indicates that the user’s TSO prefix should be used. This will be null if TSO NOPREFIX is in effect and, after substitution, the appropriate level of the audit log data set name prefix will also be null.

&&USER
Indicates that the user’s logonid (ISPF system variable ZUSER, stored in the shared pool) should be used.

&&UID
Indicates that the user’s TSO prefix should be used, when the value is non-blank. When TSO NOPREFIX is in effect, the user’s TSO logonid (ISPF system variable ZUSER, stored in the shared pool) should be used.

&&FUNCOD
Indicates that the File Manager internal function code should be used. Specifying this parameter allows the File Manager function that generated the audit log data set to be included in the audit log data set name.

&&SSID
Indicates that the currently connected IMS subsystem name should be used.

Appendix A, ″File Manager options″: For the AUDITHLQ option, change the description to the following:

Syntax

```
►►──AUDITHLQ──=──┬─────────────────────┬──────────────────────────────►◄
                 ├─user-defined hlq────┤
                 └─user-defined prefix─┘
```

AUDITHLQ

Specifies an optional high-level qualifier, or data set prefix, for File Manager audit log data sets.

The format of the generated data set names is shown below:

Table 1. Batch update status and action

File Manager component	AUDITHLQ value	Audit log data set name format
Base	Blank (none)	*userid*.FMNLOG.D*yymmdd*.T*hhmmss*
Base	Audithlq	audithlq.*userid*.FMNLOG.D*yymmdd*.T*hhmmss*
Base	Prefix	*qual1*.<*qual2*.><*qual3*.>D*yymmdd*.T*hhmmss*
DB2	Blank (none)	*userid*.FMN2AUD.<*ssid*>.D*yymmdd*.T*hhmmss*
DB2	Audithlq	*audithlq*.FMN2AUD.<*ssid*>.D*yymmdd*.T*hhmmss*
DB2	Prefix	*qual1*.<*qual2*.><*qual3*.>D*yymmdd*.T*hhmmss*
IMS	Blank (none)	u*serid*.IMSAUDIT.D*yymmdd*.T*hhmmss*
IMS	Audithlq	*audithlq*.IMSAUDIT.D*yymmdd*.T*hhmmss*
IMS	Prefix	*qual1*.<*qual2*.><*qual3*.>D*yymmdd*.T*hhmmss*

Where:

audithlq
Is any 1-8 character constant that is valid in the context of a data set name.

userid Is the userid creating the data set.

D*yymmdd*
Is the date of the activity.

T*hhmmss*
Is the time of the activity.

Prefix *qual11*.<*qual2*.><*qual3*.> as defined below

The following values can be specified for each level (*qual1*, *qual2* or *qual3*) of the data set name prefix:

XXX Any 1-8 character constant that is valid in the context of a data set name.

&&PREFIX
Indicates that the user's TSO prefix should be used. This will be null if TSO NOPREFIX is in effect.

&&USER
Indicates that the user's logonid (ISPF system variable ZUSER, stored in the shared pool) should be used.

&&UID
Indicates that the user’s TSO prefix should be used, when the value is non-blank. When TSO NOPREFIX is in effect, the user’s logonid (ISPF system variable ZUSER, stored in the shared pool) should be used.

&&FUNCOD
Indicates that the File Manager internal function code should be used. Specifying this parameter allows the File Manager function that generated the audit log data set to be included in the audit log data set name.

&&SSID
Indicates that the currently connected subsystem name (IMS or DB2 only) should be used. If this symbolic parameter is specified in FMN0POPT, the value is ignored, the substituted value is null.

When a symbolic parameter is specified (any value preceded by &&), a trailing period is required, except for *qual3*.

You cannot specify a combination of characters and symbolic parameters at the same level of the data set name prefix.

You cannot use more than one symbolic parameter at the same level of the data set name prefix.

You cannot specify a prefix value with more than three levels.

You can specify a trailing period to distinguish between a high-level qualifier value and a prefix value comprising a single level.

Examples:

`AUDITHLQ=,`
Default.

`AUDITHLQ=FMNAUD,`
HLQ value is FMNAUD.

`AUDITHLQ=FMNAUD.,`
Prefix value is FMNAUD.

`AUDITHLQ=&&USER..FMN1AUD,`
Two-level prefix.

`AUDITHLQ=FMNAUD.&&FUNCOD..&&USER`
Three-level prefix.

`AUDITHLQ=FMNAUD..&&USER..&&SSID`
Three-level prefix, SSID non-blank DB2 and IMS only.

PK66189

Initial problem description
File Manager does not produce summary statistics for functions like CHG_OUT, OVLY_OUT, FLD_OUT etc

Outline of solution
File Manager has been enhanced to add an optional tally register to the external functions it provides for procedure processing. If a tally literal is specified with the following functions CHANGE, CHG_OUT, CONTAINS, FLD_CO, FLD_OUT, FLD_TM, FLD_TYPE, NCONTAIN, OFLD_CO, OVLY_OUT, SET_OLEN, TFLD and TM, then the literal will be displayed on the tally report with a respective count. The tally report has also been modified to align the counts and to remove line spacing between report lines.

Documentation impact

This APAR requires changes to be made to the User's Guide and Reference (SC19-1239-00).

Changes to the User's Guide and Reference

Chapter 16, ″Functions″: In the section ″External REXX functions″, after the subsection ″Specifying relative positions″, add the new section shown here.

Tally register for external REXX functions: Table 2 shows the functions that support a tally register that allows you to report on the function activity.

Table 2. Functions supporting a TALLY register

Function name	Counts number of	Sample coding for tally literal
CHANGE	Strings changed	`(fld(1),'a','c',0,,,'Change 'a' to 'c' ')`
CHG_OUT	Strings changed	`chg_out('a','c',0,,,, 'Change 'a' to 'c' ')`
CONTAINS	True results	`co(fld(1,2),'aa','bb','cc',,'Contains 'aa','bb','cc'')`
FLD_CO	True results	`fld_co(1,2,c,'aa','bb','cc',, 'Contains 'aa','bb','cc'')`
FLD_OUT	Invocations	`fld_out(1,2,3,2,,'Move Columns 1,2 to Columns 3,4  ')`
FLD_TM	True results	`fld_tm(1,'01'x,,'Test under mask column 1 for '01'x  ')`
FLD_TYPE	True results	`fld_type(36,1,Z,'Check Column 36 for valid zoned')`
NCONTAIN	True results	`nco(fld(36,1),1,4,3,2,,Column 36 contains 1,4,3,2'))`
OFLD_CO	True results	`ofld_co(1,2,c,'aa','bb','cc',, 'Output contains 'aa','bb','cc'')`
OVLY_OUT	Invocations	`ovly_out('**',1,2,,,'Overlay columns 1,2 with '**'   ')`
SET_OLEN	Invocations	`Set_olen(84,'b','Change output record length to 84')`

TFLD	True results	`tfld('Age','>',64,'People over 64')`
		`tfld('Age','NN','Non-Numeric Age fields')`
		`tfld('Age','RG',21,75,'People between 21 and 75' )`
		`tfld('Name','CU','Smith','Jones',,'Common surnames')`

TM	True results	`tm(fld(1,1),'01'x,,'Test under mask column 1 for '01'x ')`

Specifying your tally register: The tally register is defined when you provide a literal value as an additional operand to the functions in Table 2. For functions that have a fixed number of operands, the literal operand is the next positional operand beyond the defined operands for the given function. For functions that have an indefinite number of operands, a null operand is required to delimit the function operands and to denote the next operand as a tally literal.

Example 1

```
IF FLD_CO(1,8,c,'a',,'Number of records with "a"') then
  chg_out('a','c',0,,,,'Number of strings changed from "a" to "c"')
```

produces this tally report:

```
TALLY summary report
----------------------------------------------------------
Number of records with "a"                     4
Number of strings changed from "a" to "c"     32
```

Note: FLD_CO can have an indefinite number of search literals. As a result, the tally register is specified by `,,'Number of records with "a"`.

For CHG_OUT, no null positional delimiter is required and the TALLY literal must be the seventh operand.

Example 2

```
*FASTREXX
if fld_tm(1,'01'x) then do;
  OVLY_OUT('**',1,2,,, 'Count of first 2 chars set to "**"')
  return
end;
```

produces this tally report:

```
TALLY summary report
-------------------------------------------------------
Count of first 2 chars set to "**"     4
```

Using a tally register: Each tally register is defined by the literal description. If you code the same literal description in a number of functions, then the same tally registered is incremented as determined by the function being invoked.

PK66242

Initial problem description

In FM/DB2, the edit function is not protected with SEC=YES (in FMN2POPT) and a facility class rule of FILEM.DB2.UPDATE that prevents access.

Outline of solution

File Manager DB2 component has been updated to correct the problem.

Documentation impact

This APAR requires changes to be made to:

- Customization Guide (SC19-1238-01)

Changes to the Customization Guide

Chapter 13, ″Customizing FM/DB2″: Replace the following section:

Customizing to protect update functions in File Manager/DB2: You can use an external security product to write facility class rules to protect update functions within FM/DB2.

The following FM/DB2 functions are considered to be update functions:

Table 3.

Function	Menu option	Description
D2E	2	DB2 edit
DBC	3.3	Copy utility
D2I	3.6	Import utility
D2G	3.8	DB2 data create
DBSBSP	4.1	Basic select prototyping
DBSASP	4.2	Advanced select prototyping
DBSEDX	4.4	DB2 edit and execute SQL
DBSENX	4.3	DB2 enter and execute SQL

If you require protection of update functions, specify SEC=YES in the FMN2POPT module (see the SEC parameter in Appendix A, File Manager options).

The facility class rules that are required are:
FILEM.DB2.UPDATE
FILEM.FUNCTION.*function_code*

Example 1

To protect all DB2 update functions:
Specify SEC=YES in the FMN2POPT
Write a facility class rule for FILEM.DB2.UPDATE

Example 2

To protect the DB2 editor function only:
Specify SEC=YES in the FMN2POPT
Write a facility class rule for FILEM.FUNCTION.D2E

The FILEM.FUNCTION facility class rule is checked first. If access is denied via this rule then no further checking is done.

If there is no FILEM.FUNCTION.*function_code* rule, the more general facility class rule FILEM.DB2.UPDATE is checked.

Appendix A, ″File Manager options″: In the section ″SEC″, replace the note following the description of the SEC parameter with:

Note: To protect update functions in File Manager base function set SEC=YES in FMN0POPT.

To protect update functions in FM/DB2 set SEC=YES in FMN2POPT.

To protect update functions in FM/CICS set SEC=YES in FMN3POPT.

Setting this option in FMN1POPT has no effect in FM/IMS.

UK34079, UK34083, UK34084, UK34085, UK34086, UK34099, UK34100, UK34102, UK34104, UK34141, UK34142

Release Date: **10 March 2008**

This set of PTFs contains the following APAR fixes:

APAR #	**APAR Abstract**	**Doc Impact**
PK59101	Generated JCL from template update utility is incorrect.	None.
PK59559	Unable to open a VSAM file under FM/CICS When file is in a different region.	None.
PK60207	Using FM 3.5 on indirect cataloged datasets results in S0C4.	None.

APAR #	APAR Abstract	Doc Impact
PK60448	When trying to use record resynchronization with a DSCMP, File Manager receives CC=8.	User's Guide and Reference (SC19-1239-00)
PK60457	IRX0251E user abend code 0901, display VSAM file entry info.	None.
PK60477	ABEND0C4 PIC04 occurred when FINDNOT command was executed from File Manager FCH 3.6 Find/Change utility.	None.
PK60490	File Manager USERLMS term error.	None.
PK60567	Compare of a array.	None.
PK60571	FM getting 878-10 abend trying to browse a large file.	None.
PK60649	Inconsistent behavior for Preserve copybook library option.	User's Guide and Reference (SC19-1239-00) User's Guide and Reference for IMS (SC19-1241-00)
PK60773	SQLCODE : -508 DSNTIAR CODE : 0 at delete of a line in a table.	None.
PK60895	File Manager ddname FMNIN not freed by PL/I compiler.	None.
PK61280	File Manager Korean service translations Q1 2008.	None.

PK60648

Initial problem description

1. DSCMP returning CC=8 when using readahead synchronization.

2. Incorrect results may be returned when using readahead synchronization (PFM03282 & PFM03283).

Outline of solution

File Manager has been updated to correctly position read ahead buffers, no longer issuing message ″Record resynchronization was not possible within the read-ahead limit.″

Documentation impact

This APAR requires changes to be made to the User's Guide and Reference (SC19-1239-00).

Changes to the User's Guide and Reference

Chapter 6, ″Managing data sets″: In the section ″Comparing data sets″, subsection ″Record synchronization″, in item **2. Read-ahead, section How the records are synchronized, remove the words ″and the comparison is terminated″ from the last sentence.**

PK60649

Initial problem description

1. The Preserve copybook library option and Use library data sets option are both being ignored when a template is updated from the member list with File Manager base Option 7.4.

2. The Preserve copybook library option and Specify library data sets option are both being ignored when a template is updated from the member list with File Manager/IMS Option 4.4.

3. The Override compiler options on the Compiler Language Selection panel is not persistant.

Outline of solution

The interactive interface for template update will now honor the Preserve copybook library and Use library data sets options when invoked from option 7.4.

For IMS, the interactive interface will honor the Preserve copybook library and Specify library data sets options when invoked from option 4.4.

The Preserve copybook library option will be able to be set from the Compiler language selection options panel to make it generally available to control the behavior of the template update when invoked from places other than the template update utility.

All values on the Compiler language selection panel will persist after being set.

Documentation impact

This APAR requires changes to be made to:
- User's Guide and Reference (SC19-1239-00)
- User's Guide and Reference for IMS (SC19-1241-00)

Changes to the User's Guide

Chapter 4, ″Creating and editing templates″: In the section ″Managing templates″, subsection ″Setting your template processing options″, after subsection ″Overriding Compiler options″, add this new subsection:

Determining the search order for your copybooks

The **Preserve copybook library** option allows you to determine the search order File Manager uses when looking for a specified copybook.

To determine the search order for your copybooks:

1. Select the **Compiler Language Selection** option from the **Options** pull-down menu.

   File Manager displays the Compiler Language Selection pull-down menu.
2. Select the **Preserve copybook library** option.

   Selecting this option causes the template update to attempt to locate the copybook in the data set in which it was originally found.

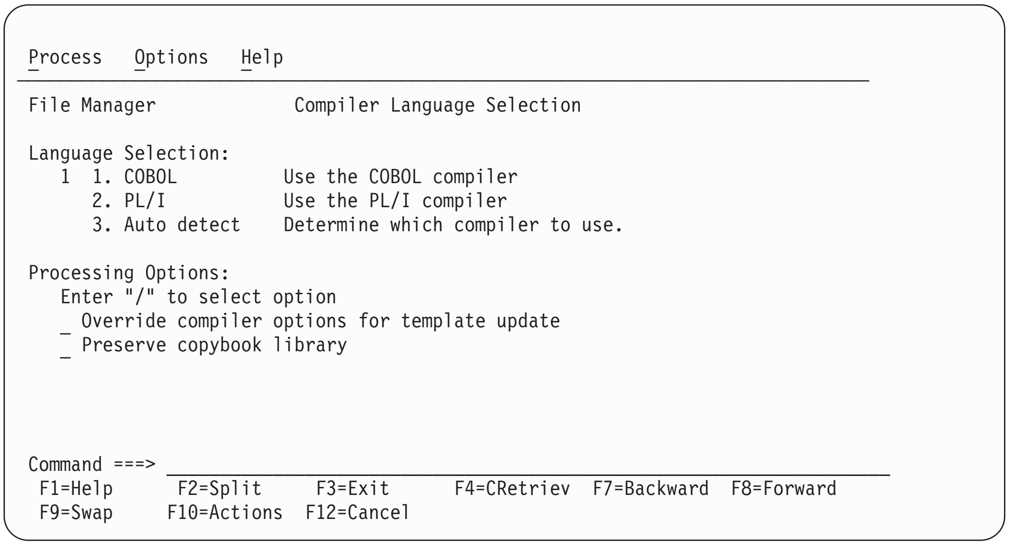
   Not selecting this option, or if the copybook cannot be located, causes the template update to search the library data sets in order for the first occurrence of the copybook.

The option only applies to template update.

You can also access the **Preserve copybook library** option on the Compiler Language Selection panel.

Chapter 14, ″Panels and fields″: In the section ″Compiler Language Selection panel″:

- Replace the existing screen capture with:



```
 Process   Options   Help
────────────────────────────────────────────────────────────────────────────────
 File Manager             Compiler Language Selection

 Language Selection:
    1  1. COBOL          Use the COBOL compiler
       2. PL/I           Use the PL/I compiler
       3. Auto detect    Determine which compiler to use.

 Processing Options:
    Enter "/" to select option
    _ Override compiler options for template update
    _ Preserve copybook library




 Command ===> ____________________________________________________________________
  F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
  F9=Swap     F10=Actions  F12=Cancel
```

Figure 1. Compiler Language Selection

- After the description for the processing option **Override compiler options for template update**, add″:

  Preserve copybook library

  Ensures that, if a copybook still exists in the library that it was previously found in and that library is in the list that the update is using, then that version of the copybook is used if:

  – This option is not selected, or

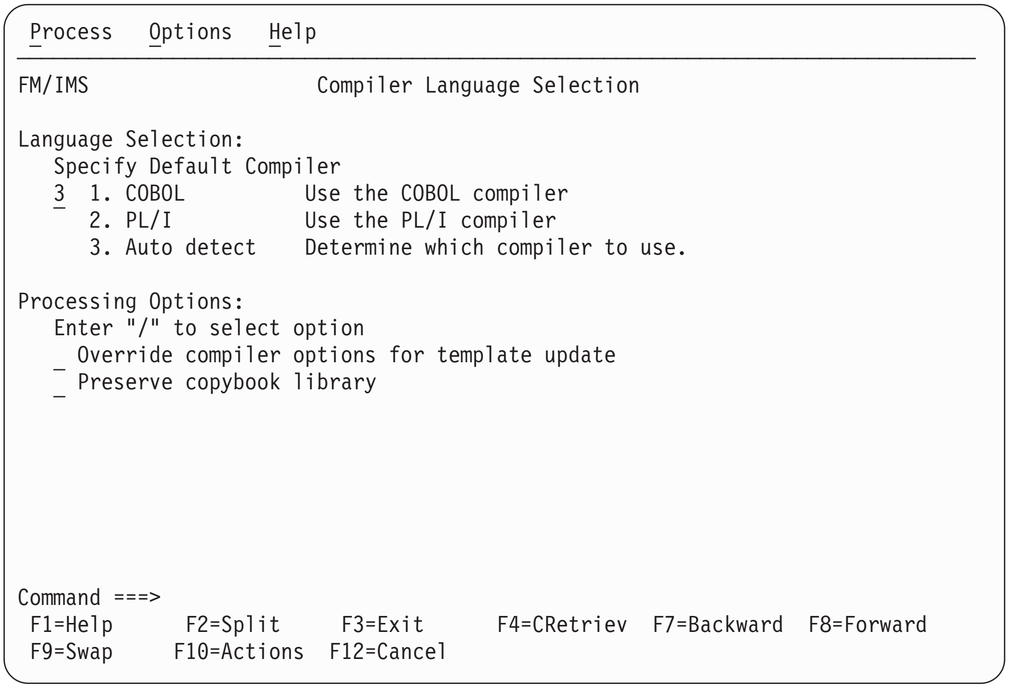
  – The copybook no longer exists in the library it was previously found in, or

  – That library is not in the list the update is using then the utility searches the libraries in the order they are listed and uses the first version of the copybook that it finds.

Changes to the User’s Guide and Reference for IMS

Chapter 9, ″Panels and fields″: In the section ″Compiler Language Selection panel″:

- Replace the existing screen capture with:



```
 Process   Options   Help
─────────────────────────────────────────────────────────────────────────────────
FM/IMS                  Compiler Language Selection

Language Selection:
   Specify Default Compiler
   3  1. COBOL          Use the COBOL compiler
      2. PL/I           Use the PL/I compiler
      3. Auto detect    Determine which compiler to use.

Processing Options:
   Enter "/" to select option
   _ Override compiler options for template update
   _ Preserve copybook library







Command ===>
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

Figure 2. Compiler Language Selection panel

- After the description for the processing option **Override compiler options for template update**, add″:

  Preserve copybook library
  Ensures that, if a copybook still exists in the library that it was previously found in and that library is in the list that the update is using, then that version of the copybook is used if:
  – This option is not selected, or
  – The copybook no longer exists in the library it was previously found in, or
  – That library is not in the list the update is using then the utility searches the libraries in the order they are listed and uses the first version of the copybook that it finds.

UK33104, UK33105, UK33106, UK33107

Release Date: **31 January 2008**

This set of PTFs contains the following APAR fixes:

APAR #	**APAR Abstract**	**Doc Impact**
PK58738	File Manager support of unlike concatenation in FM batch.	User's Guide and Reference (SC19-1239-00)
PK58753	FM problems found internally in support of spanned VSAM files with large record lengths.	None.
PK58760	Incorrect handling of extended volume names.	None.

APAR #	APAR Abstract	Doc Impact
PK58825	Garbled display results upon return from 80-byte field information screen to the database browse screen.	None.
PK58847	FM/IMS incorrect result using V3 template.	None.
PK58885	File Manager shows NOCOMP in job output.	None.
PK58993	Date fields in FMNPSCKL Japanese panel are truncated.	None.
PK59027	Selecting specific fields for XML export is ignored.	None.
PK59034	Loop in FM trying to exit the DFSVSAMP member selection panel by using ′=X′.	None.
PK59263	Ignore settings for RSR and BTCHFREQ and issue a message.	User’s Guide and Reference for IMS Data (SC19-1241-00)
PK59416	Msg FMN0256I IRLM PARAMETER IGNORED, INSTALLATION VALUE IRLM=N USED.	None.

PK58738

Initial problem description

″Function terminated″ or other errors may be reported when using concatenated unlike data sets.

Outline of solution

File Manager has been enhanced to support dataset concatenation with unlike attributes.

Documentation impact

This APAR requires changes to be made to:

- User’s Guide and Reference (SC19-1239-00)

Changes to the User’s Guide and Reference

Chapter 1, ″File Manager overview″: In the section, ″Data sets and records″, replace the existing text for this section with:

File Manager provides production and development logical file manipulation for UNIX files, sequential (PS and PDS(E)) and VSAM (including IAM) data sets.

You can view and edit or otherwise process these types of data sets:

- QSAM data sets: A member of a partitioned data set or a sequential data set with any of these data set attributes:
  - PS data set type
    - Basic
    - Library
    - Extended
    - Large

    - Compressed
  - Record format
    - Fixed, variable, or undefined
    - Blocked or unblocked
    - Spanned
    - ISPF packed format data
  - Record length
    - For fixed-length data: 1–32760, 10-65535 (for tape using LBI)
    - For variable-length data: 1–32752, 14-65531 (for tape using LBI)
    - For spanned variable-length data: 1–16MB
- VSAM data sets:
  - ESDS, including spanned format
  - KSDS, including spanned format
  - RRDS (fixed-length or variable-length)
  - PATH (PATHs related to a non-unique alternate index are restricted to browse only)
  - AIX
- UNIX files: A file contained in the z/OS UNIX File System, including z/FS and HFS
  - Record format
    - Text (delimited by CR,LF or NL)
    - Binary
  - Record length
    - Text mode files are variable length: 1-32752
    - Binary mode files are fixed length: 1-32760

File Manager Version 8 is able to use COBOL and PL/I copybooks that are stored in a PDS, a PDSE, a CA-Panvalet library, or a library accessed using the Library Management System Exit.

Chapter 6, ″Managing data sets″: In the section, ″Copying data sets″, in the paragraph ″You can use the Copy Utility to:″, after

- Copy data from any partitioned, sequential, or VSAM data set to any other partitioned, sequential, or VSAM data set.

add (in the same bullet point)″:

″Concatenated like and unlike sequential data sets are supported. Note that, under some conditions (with tape data sets), File Manager may not be able to detect unlike data set attributes and still invoke DFSORT for copy processing. Such invocation may fail since DFSORT does not allow for unlike concatenation of data sets. In such cases, the DFSORT use may be disabled with the NOSORT function to allow for successful processing of concatenated datasets with unlike attributes.

Concatenated partitioned data sets with like attributes are fully supported. Mixed partitioned data sets with like and unlike attributes are processed correctly providing that the members being selected for processing come from libraries with matching attributes, otherwise errors may be reported.″

Chapter 6, ″Managing data sets″: In the section, ″Printing from File Manager″, subsection ″Printing data sets″, after the first sentence in the first paragraph (″You can use the Print Utility to print sequential, partitioned data set members, and VSAM files in a selected format.″), add:

″Concatenated like and unlike sequential data sets are supported. Note that, under some conditions (with tape data sets), File Manager may not be able to detect unlike data set attributes and still invoke DFSORT for copy processing. Such invocation may fail as DFSORT does not allow for unlike concatenation of data sets. In such cases, you can disable DFSORT with the NOSORT function to allow for successful processing of concatenated datasets with unlike attributes.″

Chapter 16, ″Functions″:

- In the section, ″General tips about performance when you use File Manager functions″, replace the bullet point starting ″File Manager can use DFSORT technology ...″ with:
  - File Manager can use DFSORT technology even if it needs to process individual records, such as when using a PROC or a template or copybook. However, some things can prevent File Manager from using DFSORT I/O technology on a particular sequential input file:
    - An IO exit is being used on the input data set.
    - The input data set:
      - Is a concatenated data set with unlike attributes.
      - Is an LRECL=X data set.
      - Is a UNIX file.
      - Contains records of undefined length (RECFM=U).
      - Is being processed with a template that contains segmented records.

    File Manager makes the determination about DFSORT use based on the above conditions and other factors which may affect performance. Note that, under some conditions (with tape data sets), File Manager may not be able to detect unlike data set attributes and still invoke DFSORT for copy processing. Such invocation may fail since DFSORT does not allow for unlike concatenation of data sets. In such cases, you can disable DFSORT with the NOSORT function to allow for successful processing of concatenated datasets with unlike attributes.
  - Add the description of the NOSORT function:

    NOSORT (Disable the use of DFSORT)

    Purpose
    Disable the use of DFSORT for copy or print processing of data sets.

    Usage notes
    File Manager may use DFSORT internally for processing to achieve better performance. Under certain conditions (for example, unlike concatenated input data sets), such use may result in DFSORT processing errors and should be disabled.

    Syntax

    ```
    ►►──NOSORT─────────────────────────────────────────────────►◄
    ```
  - Add the following under the ″Usage notes″ for the DSC function (as the fifth bullet point):
    - Copy concatenated data sets with like or unlike attributes. Note that, under some conditions (with tape data sets), File Manager may not be able to detect unlike data set attributes and still invoke DFSORT for processing. Such invocation may fail as DFSORT does not allow for unlike

concatenation of data sets. In such cases, you can disable the DFSORT with the NOSORT function to allow for successful processing of concatenated datasets with unlike attributes.

– Add the following under the ″Usage notes″ for the DSP function:

  You can Print concatenated data sets with like or unlike attributes. Note that, under some conditions (with tape data sets), File Manager may not be able to detect unlike data set attributes and still invoke DFSORT for processing. Such invocation may fail as DFSORT does not allow for unlike concatenation of data sets. In such cases, you can disable DFSORT with the NOSORT function to allow for successful processing of concatenated datasets with unlike attributes.

PK59263

Initial problem description

FM/IMS version 8 Print (IPR), Extract (IXB), Load (ILB), Batch Edit (IEB), Batch Browse (IBB) and Initialize (DIB) jobs fail with a return code of 16 and message ″FMN0091I Missing or invalid control card″ when the FMIMSIN input includes the line:

```
$$FILEM     RSR=N,
or
$$FILEM     RSR=Y,
```

FM/IMS version 7 and earlier generated JCL that included these lines. This parameter is not used in FM/IMS version 8. So currently, if you want your old JCL to work with version 8, you must delete these lines.

Outline of solution

FM/IMS has been modified to ignore the RSR specification. So these lines will no longer cause the job to fail. When FM/IMS encounters them, it now issues the message: ″FMN0116I Extraneous parameter RSR ignored″.

Documentation impact

This APAR requires changes to be made to:

- User’s Guide and Reference for IMS Data (SC19-1241-00)

Changes to the User’s Guide and Reference for IMS Data

Chapter 11, ″Batch reference″: Replace the entire chapter with what is shown here.

Batch reference

This chapter describes the FM/IMS batch functions. FM/IMS provides batch functions to:

- Update templates
- Update views
- Update criteria sets
- Extract data from your IMS databases
- Load your IMS databases
- Print data from your IMS databases
- Edit data in your IMS databases (Batch edit)
- View data in your IMS database (Batch browse)
- Initialize your IMS database

How to use this reference chapter: This chapter lists the FM/IMS batch functions.

Function reference: How to use this chapter

The following subsections are supplied, where relevant, for each function:

Purpose
A short summary of what you can do with the function.

Usage notes
A more detailed explanation of the function.

Return Codes
Lists the return codes that are specific to the function.

Related functions
Other functions that are related in some way.

Syntax
A syntax diagram, followed by a parameter list. The parameter list describes the parameter, and shows (where appropriate) its maximum value and default value.

If you specify parameters that are not relevant to the function, FM/IMS ignores them. If you specify the same parameter more than once, FM/IMS uses the first value that you specify.

For information on how to read a syntax diagram, see ″How to read the syntax diagrams″ on page ix.

Batch example
A listing of a sample batch job.

Reports
A sample listing of any report or reports produced by the batch JCL.

IMS Template Update (ITU):

Purpose
You can use this utility to:

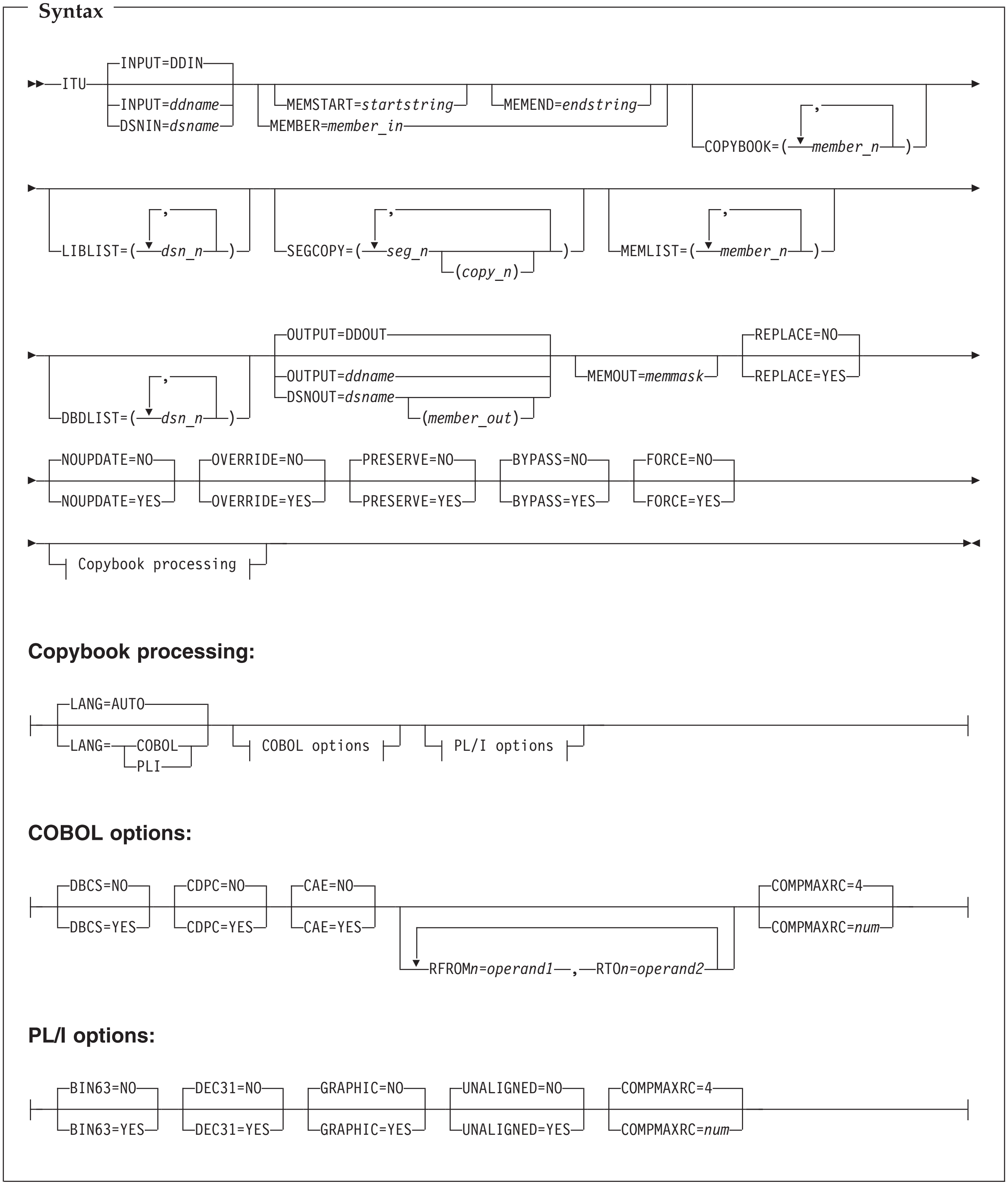
- Update templates with the latest copybook or DBD definitions.
- Create new templates based upon existing templates.

Usage notes

- Template members selected for processing can be filtered by specifying copybook names or masks so that only templates referencing those copybooks are included.

Related functions

ICU	IMS Criteria sets update
IVU	IMS Views update



INPUT=*ddname*
This points to the input DD name which can have one or more associated data sets that must be valid template data sets. If you do not specify INPUT or DSN parameters, then the DD DDIN is used as a default DD

name for the input data sets. You can specify concatenated data sets. If an output data set is not provided, the template is updated in the data set from which it originated.

DSNIN=*dsname*
Data set name where templates reside.

MEMBER=*member_in*
The name of a single member in a PDS, or a member name pattern representing one or more members in a PDS to be processed.

A member name pattern can consist of any characters that are valid in a member name and two special pattern characters: the asterisk (*) and the percent symbol (%).

- * represents any number of characters. As many asterisks as required can appear anywhere in a member name pattern. For example, if you enter a member name pattern of `*d*`, all members in the PDS whose name contains “d” are processed.
- % is a place holding character that means a single character. As many percent symbols as necessary can appear anywhere in a member name pattern. For example, if you enter a member name pattern of `%%%%`, all members in the PDS whose name is four characters in length are processed.

member_in is ignored if the data set is not a PDS.

MEMSTART=*startstring*
Is used to specify the start of a range of member names to be included. If MEMSTART is specified but MEMEND is omitted, all members of the PDS(E) from the *startstring* value onwards are included. *startstring* can have the same values, including wild cards, as for the *member-in* parameter of the MEMBER keyword.

MEMEND=*endstring*
Is used to specify the end of a range of member names to be included. If MEMEND is specified but MEMSTART is omitted, all members of the PDS(E) up to the *endstring* value onwards are included. *endstring* can have the same values, including wild cards, as for the *member_in* parameter of the MEMBER keyword.

COPYBOOK
Allows you to specify a list of member names or patterns used to filter so that only templates referencing those copybooks, or copybooks that match the patterns, are selected for processing.

member_n
The name of the member to be processed. Generic name masks are not allowed.

LIBLIST
Allows you to specify a list of up to twelve data set names used to replace the copybook library lists in the selected templates.

dsn_n The name of the data set name to be processed. Generic name masks are not allowed.

SEGCOPY
Allows you to specify a list of segment names with associated copybook names to be added to the template. If you do not specify the associated

copybook name, File Manager excludes the segment from the process. If the new segments have not yet been added to the DBD the process will not succeed.

MEMLIST
Allows you to specify a list of member names with optional associated output template names. If you do not specify the associated template name, File Manager uses the copybook name or the name as identified by the MEMOUT mask, *memmask*.

member_n
The name of the member to be processed. Generic name masks are not allowed.

template_n
The name of the template after it has been copied to the output data set. If unspecified, the output template is not renamed.

DBDLIST
The list of DBD libraries that are used to determine the first library containing the DBD pertaining to the template being updated. The DBD library name is updated to the first data set name containing the same member name. If a corresponding DBD member is not found then the template update will fail.

dbd_n The name of the DBD library to be processed. Generic name masks are not allowed.

OUTPUT=ddname
Identifies the DD card which points to the template data set in which the templates are stored or replaced. It must refer to a PDS, PDSE, or sequential data set. Concatenated data sets are not supported.

DSNOUT=*dsname*
Defines the data set name where templates are created. It must be a PDS, PDSE, or sequential data set. You can further describe this data set, as follows:

(*member-out***)**
Where DSNOUT=*dsname* specifies a PDS and you want to send the output to a specific member within this data set, this defines the output member name.

MEMOUT=*memmask*
Where a number of input members have been specified, you can specify a member name pattern for the output templates, allowing you to rename your templates as they are created. The member name pattern can consist of any characters that are valid in a member name and two special pattern characters: the asterisk (*) and percent sign (%).

Asterisk (*)
The asterisk is a place-holding character that means multiple characters with no change. Only one asterisk should appear in the mask. Any subsequent asterisk characters are treated as percent signs. For example, if you enter:

```
ABC*
```

The renamed members all begin with ABC followed by the remainder of the old member name.

Percent sign (%)

The percent sign is a place-holding character that means a single character with no change. As many percent symbols as necessary may appear anywhere in a member name. For example, if you enter:

```
%%%A*
```

The 1st 3 characters of the renamed members remain unchanged, the 4th character is replaced with the letter "A" and the remainder of the old member name remains unchanged.

REPLACE

Specifies whether or not File Manager replaces like-named templates in an output partitioned data set.

NO Like-named templates in the output partitioned data set are not replaced.

YES Like-named templates in the output partitioned data set are replaced.

NOUPDATE

Specifies whether or not File Manager writes back updates to the data set.

NO Updates are written back to the data set.

YES Updates are not written back to the data set.

OVERRIDE

Specifies whether or not File Manager overrides any compile options found in the template with the compiler options found in the parameter list.

NO Compile options found in the template are not overwritten with the compiler options found in the parameter list.

YES Compile options found in the template are overwritten with the compiler options found in the parameter list.

PRESERVE

Specifies whether or not File Manager uses the current version of the copybook.

NO File Manager searches for the first version of the copybook.

YES File Manager uses the current version of the cookbook, provided the copybook still exists in the library it was previously found in, and the library is in the list the update process is using.

File Manager searches for the first version of the copybook in the order the libraries are listed if one of these conditions applies:

- NO has been specified for this option.
- The copybook no longer exists in the library it was previously found in.
- The library is not in the list the update process is using.

FORCE

Specifies whether or not File Manager updates the template when no layout change has been detected.

NO If no layout change is detected during the update process, the update will not take place.

YES File Manager updates the template, even if no change has been made to the layout.

BYPASS

Specifies whether or not File Manager eliminates DBD checking and key processing.

Note: You should select this option when the copybook changes do not affect key position fields.

NO DBD checking and key processing performed.

YES DBD checking and key processing not performed.

Copybook processing

If you specify a copybook (instead of an existing template), then File Manager uses these processing options to compile the copybook into a template:

LANG

Determines whether File Manager automatically detects the copybook language or interprets the language as COBOL or PL/I.

AUTO

Automatically detect whether the copybook language is COBOL or PL/I, and invoke the appropriate compiler. If the compilation results in a return code greater than 4, then invoke the compiler for the other language. If the second compilation also results in a return code greater than 4, then retry the first compiler and report the compilation errors. If File Manager successfully creates a template (despite the compilation errors), then continue processing with the template.

COBOL

Invoke the COBOL compiler to create a template from the copybook. (Do not invoke the PL/I compiler, even if the COBOL compilation results in errors.)

PLI Invoke the PL/I compiler to create a template from the copybook. (Do not invoke the COBOL compiler, even if the PL/I compilation results in errors.)

COBOL options

The following options are used to compile a COBOL copybook into a template:

DBCS=YES

Use the DBCS compiler option.

DBCS=NO

Use the NODBCS compiler option.

For details on the effect of the DBCS and NODBCS compiler options, see the *IBM COBOL Programming Guide for OS/390 & VM*.

CDPC=NO

Do not use the COBOL SPECIAL-NAMES paragraph ″Decimal-point is comma″.

CDPC = YES

Use the COBOL SPECIAL-NAMES paragraph ″Decimal-point is comma″.

CAE=NO
Do not use the COBOL compile option ARITH(EXTEND).

CAE = YES
Use the COBOL compile option ARITH(EXTEND).

RFROM1 RTO1 ... RFROM5 RTO5
Up to five pairs of “From” and “To” pseudo-text character strings for the COBOL REPLACE compiler-directing statement.

If your COBOL copybooks contain characters that you want to remove or replace with other characters before compiling the copybooks into templates, then use these replacing options.

For example, if your copybooks contain colon characters (:) that you want to remove before compiling, then specify `'==:=='` as *operand1* and `'====='` as *operand2*.

For details on specifying “From” and “To” strings for COBOL REPLACE, see the *IBM COBOL Language Reference*.

COMPMAXRC
Sets the maximum acceptable return code for a copybook compile. A return code higher than the specified level causes the function to stop. Default is 4.

PL/I options
The following options are used to compile a PL/I copybook into a template:

BIN63=YES Use the LIMITS(FIXEDBIN(63)) compiler option.

BIN63=NO Use the LIMITS(FIXEDBIN(31)) compiler option.

DEC31=YES Use the LIMITS(FIXEDDEC(31)) compiler option.

DEC31=NO Use the LIMITS(FIXEDDEC(15)) compiler option.

GRAPHIC=YES
Use the GRAPHIC compiler option.

GRAPHIC=NO
Use the NOGRAPHIC compiler option.

UNALIGNED=YES
Use the DEFAULT RANGE (*) UNALIGNED, language statement to change the default alignment.

UNALIGNED=NO
Use the PL/I default.

COMPMAXRC
Sets the maximum acceptable return code for a copybook compile. A return code higher than the specified level causes the function to stop. Default is 4.

For details on the effect of these compiler options, see the *IBM VisualAge PL/I for OS/390 Programming Guide*.

Batch example

```
//FMBAT EXEC PGM=FMNIMS
//STEPLIB DD DSN=FMN.SFMNMOD1,DISP=SHR
//SYSPRINT DD SYSOUT=*
//IDIOPTS DD DSN=FMN.IDIOPTS,DISP=SHR
//FMIMSIN DD *
$$FILEM ITU DSNIN=FMN.IMS.IVP.TEMPLATE,
$$FILEM MEMBER=DJ%E,
$$FILEM MEMLIST=(DJ1E,
$$FILEM          DJ2E),
$$FILEM DBDLIST=(FMN.IMS.IVP.DBDLIB),
$$FILEM DSNOUT=FMN.IMS.IVP.TEMPLATE.MORE,
$$FILEM REPLACE=YES
```

Update report

```
IBM File Manager for z/OS IMS Component

          Template Update Report

Template      New name   Type       Status
-------------------------------------------------------------------------------
DJ1E                     Template   Replaced
DJ2E                     Template   Old template requires LIBLIST

FMN1495I 2 members read 0 Updated 0 Not changed 1 Replaced  1 Errors
```

Report fields

Template
This is the template name.

Lib This column shows the library number the template came if there is more than one input library. This column will only be present if there is more than one template input library.

New name
This is the new output template name if the template was renamed during the update process.

Type Identifies whether the object is a template view or criteria.

Status A value from the status table. See Table 4.

Table 4. Batch update status and action

Status	Explanation	Action
Updated	The template was successfully updated.	None
Not Replaced	The template exists in the output data set and the replace option is NO.	Specify replace and rerun if required.
Replaced	The template exists in the output data set and has been successfully updated with replace option YES.	None.
Compile Error	Unable to compile the copybooks associated with the template.	Rerun using option 4.4 in foreground for the failing template and look at the compile listing produced.

Table 4. Batch update status and action (continued)

Status	Explanation	Action
Corrupt Template	The internal format of the template is corrupted. This could have occurred because the template has been modified outside of File Manager.	This is an internal error. If the template has not been modified then keep a copy of it and contact you IBM representative.
SYSLIB not found *syslib*	The syslib referenced in the template could not be found.	Rerun the update in foreground using option 4.1 which will list the SYSLIBs which you can then modify. Alternatively provide LIBLIST=(dsn1,dsn2...) parameter to identify the current location for the copybooks.
SYSLIB invalid attrs *syslib*	The syslib referenced in the template has invalid attributes for the language type for this template.	Change the syslib reference using option 7.1 or LIBLIST parameter.
Copybook not found name	The copybook name could not be located in the current libraries.	Provide the data set containing the referenced copybook using either 4.1 or the LIBLIST parameter.
Storage exhausted	File Manager ran out of storage during processing	Increase the region size.
No copybooks in libraries	The library list provided either from the template or override has no copybooks.	Rerun the update in foreground using option 4.1 which will list the SYSLIBs which you can then modify. Alternatively provide LIBLIST=(dsn1,dsn2...) parameter to identify the current location for the copybooks.
Not a valid template	The type of template is not valid for update processing.	This is an internal error that should never occur—contact your IBM support center.
Duplicate name	The output template name has already been referenced by another template during update.	Correct parameters so that there are no duplicate names being saved.
Not found	The template member referenced could be not be found on the input data set.	Correct the input parameters to point to the right data set or member name.
Save error	The updated template could not be saved	Normally a space problem—check the output data set (or input data set if an output data set was not provided) and increase the size.

Table 4. Batch update status and action (continued)

Status	Explanation	Action
Update check successful	The update would have been a success, except that you ran with `NOUPDATE=YES` specified. This means that no updates were done.	None.
Updating of criteria failed	The update was not performed because fields that were previously referenced in criteria could not be found in the current versions of the copybooks.	Rerun the update in foreground using option 7.1 and the U command. This should allow you to correct the expressions that have been invalidated as a result of your copybook changes.
Updating of criteria failed	The update was not performed because fields that were previously referenced in criteria could not be found in the current versions of the copybooks.	Rerun the update in foreground using option 4.1. This should allow you to correct the expressions that have been invalidated as a result of your copybook changes.
Segment without layouts	The corresponding copy books for a given segment did not produce a layout.	Rerun the update in foreground using option 4.1. This should allow you to view the copybooks for each segment and determine which copybooks do not contain data layouts.
Old template requires liblist	The template you are trying to update is an old template that does not contain the copybook libraries required for the update process.	You need to provide a LIBLIST parameter for batch to provide the copybook data sets. Using option 4.4 you can provide the copybook data sets via the entry panel.
DBD initialization failed	The DBD referenced in the template could not be loaded properly. The DBD data set and member name will be listed with the status.	Investigate why the DBD data set and member listed in the message are incorrect. Use DBDLIST or option 4.1 online to change the DBD library referenced by the template.
Template allocate error	The corresponding template data set for a view or criteria could not be allocated for the update process.	Check the data set name. Use the TPLIST parameter or option 4.2/4.3/4.4 to update the originating template data set.
DBD not found *member_name*	The DBD for this template could not be found in the specified DBD library.	Ensure that a DBD *member_name* can be found in the DBDLIST libraries if specified or the DBD library associated with this template.

Table 4. Batch update status and action (continued)

Status	Explanation	Action
Template not found *member_name*	The template for this view or criteria set could not be found either in the specified template library.	Ensure that a template *member_name* can be found the TPLIST libraries if specified or the original template data set used to create the view or criteria set.

IMS View Update (IVU):

Purpose

You can use this utility to:

- Update one or more views.
- Create views based upon existing views.

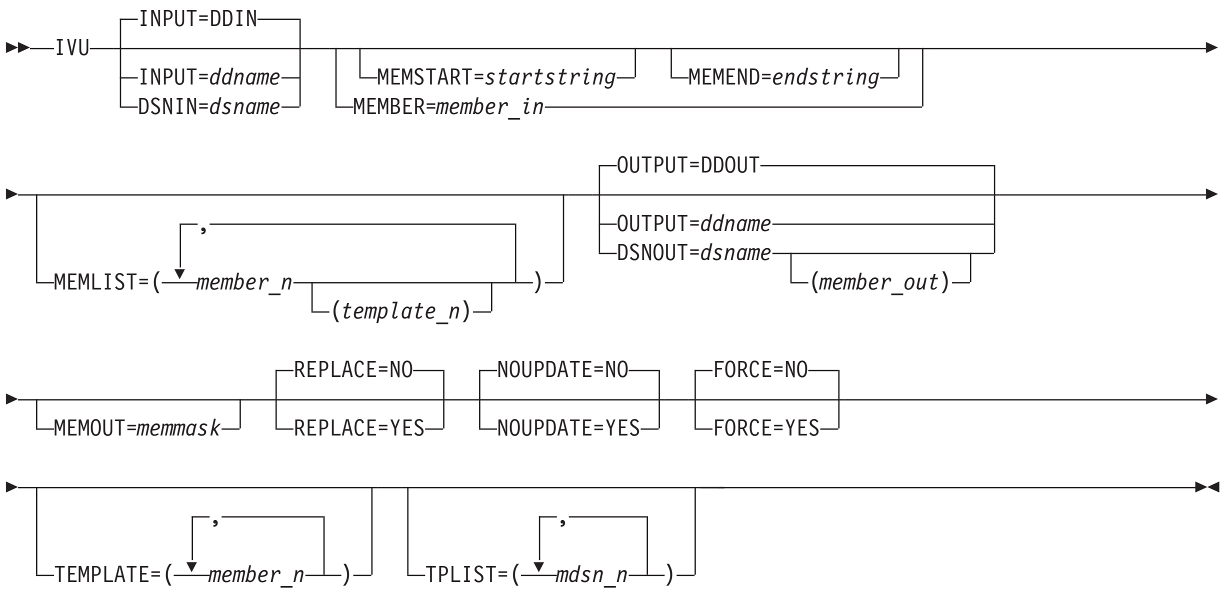
Usage notes

- You can filter the views selected for processing by providing the originating template member names or masks.

Related functions

ITU IMS Template Update
ICU IMS Criteria sets Update

Syntax



INPUT=*ddname*

This points to the input DD name which can have one or more associated data sets that must be valid template data sets. If you do not specify INPUT or DSN parameters, then the DD DDIN is used as a default DD name for the input data sets. You can specify concatenated data sets. If an output data set is not provided, the template is updated in the data set from which it originated.

DSNIN=*dsname*
: Data set name where templates reside.

MEMBER=*member_in*
: The name of a single member in a PDS, or a member name pattern representing one or more members in a PDS to be processed.

  A member name pattern can consist of any characters that are valid in a member name and two special pattern characters: the asterisk (*) and the percent symbol (%).

  * represents any number of characters. As many asterisks as required can appear anywhere in a member name pattern. For example, if you enter a member name pattern of `*d*`, all members in the PDS whose name contains "d" are processed.

  % is a place holding character that means a single character. As many percent symbols as necessary can appear anywhere in a member name pattern. For example, if you enter a member name pattern of `%%%%`, all members in the PDS whose name is four characters in length are processed.

  member_in is ignored if the data set is not a PDS.

MEMSTART=*startstring*
: Is used to specify the start of a range of member names to be included. If MEMSTART is specified but MEMEND is omitted, all members of the PDS(E) from the *startstring* value onwards are included. *startstring* can have the same values, including wild cards, as for the *member-in* parameter of the MEMBER keyword.

MEMEND=*endstring*
: Is used to specify the end of a range of member names to be included. If MEMEND is specified but MEMSTART is omitted, all members of the PDS(E) up to the *endstring* value onwards are included. *endstring* can have the same values, including wild cards, as for the *member_in* parameter of the MEMBER keyword.

MEMLIST
: Allows you to specify a list of member names with optional associated output template names. If you do not specify the associated template name, File Manager uses the copybook name or the name as identified by the MEMOUT mask, *memmask*.

  member_n
  : The name of the member to be processed. Generic name masks are not allowed.

  template_n
  : The name of the template after it has been copied to the output data set. If unspecified, the output template is not renamed.

OUTPUT=ddname
: Identifies the DD card which points to the template data set in which the templates are stored or replaced. It must refer to a PDS, PDSE, or sequential data set. Concatenated data sets are not supported.

DSNOUT=*dsname*
: Defines the data set name where templates are created. It must be a PDS, PDSE, or sequential data set. You can further describe this data set, as follows:

(*member-out***)**
Where DSNOUT=*dsname* specifies a PDS and you want to send the output to a specific member within this data set, this defines the output member name.

MEMOUT=*memmask*
Where a number of input members have been specified, you can specify a member name pattern for the output templates, allowing you to rename your templates as they are created. The member name pattern can consist of any characters that are valid in a member name and two special pattern characters: the asterisk (*) and percent sign (%).

Asterisk (*)
The asterisk is a place-holding character that means multiple characters with no change. Only one asterisk should appear in the mask. Any subsequent asterisk characters are treated as percent signs. For example, if you enter:

```
ABC*
```

The renamed members all begin with ABC followed by the remainder of the old member name.

Percent sign (%)
The percent sign is a place-holding character that means a single character with no change. As many percent symbols as necessary may appear anywhere in a member name. For example, if you enter:

```
%%%A*
```

The 1st 3 characters of the renamed members remain unchanged, the 4th character is replaced with the letter “A” and the remainder of the old member name remains unchanged.

REPLACE
Specifies whether or not File Manager replaces like-named templates in an output partitioned data set.

NO Like-named templates in the output partitioned data set are not replaced.

YES Like-named templates in the output partitioned data set are replaced.

NOUPDATE
Specifies whether or not File Manager writes back updates to the data set.

NO Updates are written back to the data set.

YES Updates are not written back to the data set.

FORCE
Specifies whether or not File Manager updates the template when no layout change has been detected.

NO If no layout change is detected during the update process, the update will not take place.

YES File Manager updates the template, even if no change has been made to the layout.

TEMPLATE
The list of member names or patterns that is used to filter, so that only views that were built from templates matching the names listed are selected for processing.

member_n
The name of the member to be processed. Generic name masks are not allowed.

TPLIST
The list of template data sets that is used to update the template data set name for views. The template data set is updated to the first data set in the list that contains the corresponding template member name that was used to create the current view.

dsn_n The name of the data set to be processed. Generic name masks are not allowed.

Batch example

```
//FMBAT EXEC PGM=FMNIMS
//STEPLIB DD DSN=FMN.SFMNMOD1,DISP=SHR
//SYSPRINT DD SYSOUT=*
//IDIOPTS DD DSN=FMN.IDIOPTS,DISP=SHR
//FMIMSIN DD *
$$FILEM IVU DSNIN=FMN.IMS.IVP.VIEW,
$$FILEM MEMBER=D*,
$$FILEM MEMLIST=(DJ2EVJU),
$$FILEM TPLIST=(FMN.IMS.IVP.TEMPLATE.MORE),
$$FILEM DSNOUT=FMN.IMS.IVP.VIEW.MORE,
$$FILEM REPLACE=YES
```

Update report

```
IBM File Manager for z/OS IMS Component

          Template Update Report

Template      New name   Type       Status
--------------------------------------------------------------------------------
DJ2EVJU                  View       Updated

FMN1494I 1 members read 1 Updated 0 Not changed 0 Not replaced  0 Errors
```

Report fields

Template
This is the template name.

Lib This column shows the library number the template came if there is more than one input library. This column will only be present if there is more than one template input library.

New name
This is the new output template name if the template was renamed during the update process.

Type Identifies whether the object is a template view or criteria.

Status A value from the status table - see Table 4 on page 61.

Criteria set update (ICU): The ICU (IMS Criteria set Update) function provides you with the ability to update criteria sets using a batch process.

Criteria set update (ICU)

Purpose

You can use this utility to:

- Update one or more criteria sets.
- Create criteria sets based upon existing existing criteria sets.

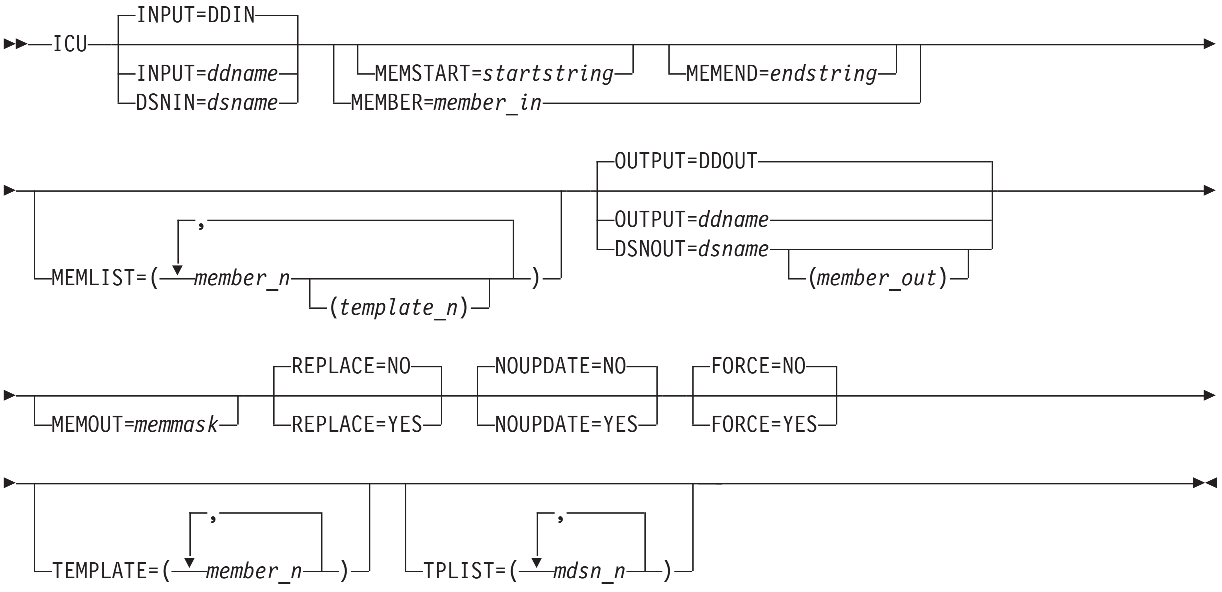
Usage notes

- You can filter the criteria sets selected for processing by providing the originating template member names or masks.

Related functions

ITU IMS Template Update
IVU IMS Views Update

Syntax



INPUT=*ddname*

This points to the input DD name which can have one or more associated data sets that must be valid template data sets. If you do not specify INPUT or DSN parameters, then the DD DDIN is used as a default DD name for the input data sets. You can specify concatenated data sets. If an output data set is not provided, the template is updated in the data set from which it originated.

DSNIN=*dsname*

Data set name where templates reside.

MEMBER=*member_in*

The name of a single member in a PDS, or a member name pattern representing one or more members in a PDS to be processed.

A member name pattern can consist of any characters that are valid in a member name and two special pattern characters: the asterisk (*) and the percent symbol (%).

* represents any number of characters. As many asterisks as required

can appear anywhere in a member name pattern. For example, if you enter a member name pattern of `*d*`, all members in the PDS whose name contains “d” are processed.

% is a place holding character that means a single character. As many percent symbols as necessary can appear anywhere in a member name pattern. For example, if you enter a member name pattern of `%%%%`, all members in the PDS whose name is four characters in length are processed.

member_in is ignored if the data set is not a PDS.

MEMSTART=*startstring*
Is used to specify the start of a range of member names to be included. If MEMSTART is specified but MEMEND is omitted, all members of the PDS(E) from the *startstring* value onwards are included. *startstring* can have the same values, including wild cards, as for the *member-in* parameter of the MEMBER keyword.

MEMEND=*endstring*
Is used to specify the end of a range of member names to be included. If MEMEND is specified but MEMSTART is omitted, all members of the PDS(E) up to the *endstring* value onwards are included. *endstring* can have the same values, including wild cards, as for the *member_in* parameter of the MEMBER keyword.

MEMLIST
Allows you to specify a list of member names with optional associated output template names. If you do not specify the associated template name, File Manager uses the copybook name or the name as identified by the MEMOUT mask, *memmask*.

member_n
The name of the member to be processed. Generic name masks are not allowed.

template_n
The name of the template after it has been copied to the output data set. If unspecified, the output template is not renamed.

OUTPUT=ddname
Identifies the DD card which points to the template data set in which the templates are stored or replaced. It must refer to a PDS, PDSE, or sequential data set. Concatenated data sets are not supported.

DSNOUT=*dsname*
Defines the data set name where templates are created. It must be a PDS, PDSE, or sequential data set. You can further describe this data set, as follows:

(*member-out***)**
Where DSNOUT=*dsname* specifies a PDS and you want to send the output to a specific member within this data set, this defines the output member name.

MEMOUT=*memmask*
Where a number of input members have been specified, you can specify a member name pattern for the output templates, allowing you to rename your templates as they are created. The member name pattern can consist of any characters that are valid in a member name and two special pattern characters: the asterisk (*) and percent sign (%).

Asterisk (*)
The asterisk is a place-holding character that means multiple characters with no change. Only one asterisk should appear in the mask. Any subsequent asterisk characters are treated as percent signs. For example, if you enter:

```
ABC*
```

The renamed members all begin with ABC followed by the remainder of the old member name.

Percent sign (%)
The percent sign is a place-holding character that means a single character with no change. As many percent symbols as necessary may appear anywhere in a member name. For example, if you enter:

```
%%%A*
```

The 1st 3 characters of the renamed members remain unchanged, the 4th character is replaced with the letter "A" and the remainder of the old member name remains unchanged.

REPLACE
Specifies whether or not File Manager replaces like-named templates in an output partitioned data set.

NO Like-named templates in the output partitioned data set are not replaced.

YES Like-named templates in the output partitioned data set are replaced.

NOUPDATE
Specifies whether or not File Manager writes back updates to the data set.

NO Updates are written back to the data set.

YES Updates are not written back to the data set.

FORCE
Specifies whether or not File Manager updates the template when no layout change has been detected.

NO If no layout change is detected during the update process, the update will not take place.

YES File Manager updates the template, even if no change has been made to the layout.

TEMPLATE
The list of member names or patterns that is used to filter, so that only criteria sets that were built from templates matching the names listed are selected for processing.

member_n
The name of the member to be processed. Generic name masks are not allowed.

TPLIST
The list of template data sets that is used to update the template data set name for criteria sets. The template data set is updated to the first data set in the list that contains the corresponding template member name that was used to create the current criteria set.

dsn_n The name of the data set to be processed. Generic name masks are not allowed.

Batch example

In this example, a criteria set is updated, removing a segment from the DBD.

```
//FMBAT EXEC PGM=FMNIMS
//STEPLIB DD DSN=FMN.SFMNMOD1,DISP=SHR
//SYSPRINT DD SYSOUT=*
//IDIOPTS DD DSN=FMN.IDIOPTS,DISP=SHR
//FMIMSIN DD *
$$FILEM ICU DSNIN=FMN.IMS.IVP.CRITERIA,
$$FILEM MEMBER=DJ1E*,
$$FILEM MEMLIST=(DJ1ESS),
$$FILEM TPLIST=(FMN.IMS.IVP.TEMPLATE.LESS),
$$FILEM DSNOUT=FMN.IMS.IVP.CRITERIA.LESS,
$$FILEM REPLACE=YES
```

Update report

```
Update report

IBM File Manager for z/OS IMS Component

          Template Update Report

Template      New name   Type       Status
-------------------------------------------------------------------------------
DJ1ESS                   Criteria   Replaced

FMN1494I 1 members read 0 Updated 1 Not changed 1 Replaced 0 Errors
```

Report fields

Template
This is the template name.

Lib This column shows the library number the template came if there is more than one input library. This column will only be present if there is more than one template input library.

New name
This is the new output template name if the template was renamed during the update process.

Type Identifies whether the object is a template view or criteria.

Status A value from the status table - see Table 4 on page 61.

Extract (IXB):

Purpose
The Extract dialog generates JCL that runs the Extract (IXB) function. This function extracts all segments or selected segments from a database, and saves them to a sequential data set. You can edit the JCL and IXB input parameters before you submit the job, or you can create your own batch job to run the Extract function.

Usage notes
The Extract JCL generated by FM/IMS has the following flows:

Extract (IXB)

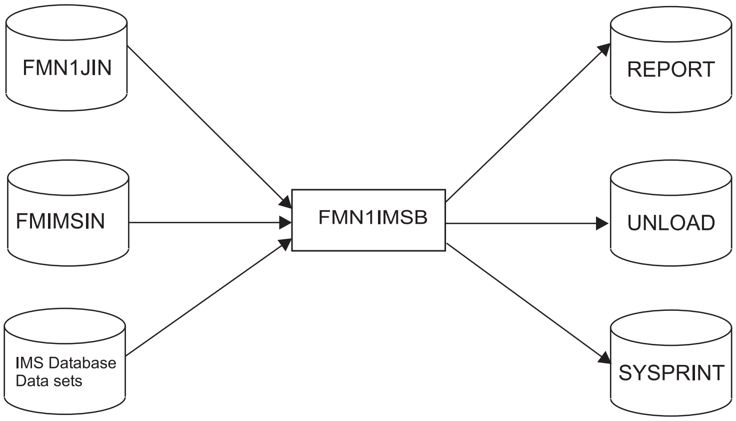


Figure 3. Flows for Extract (IXB) step

FMN1JIN
Points to the FM/IMS skeleton library. This DD statement is only used when you extract data from HALDB databases.

FMIMSIN
Contains the control statements for the Extract. These control statements are discussed below.

REPORT
Defines the output data set for the Database Extract Summary report.

IMS database data sets
In DLI mode, FM/IMS includes DDs for each of the IMS database data sets associated with the IMS database being extracted. If the database has logical relationships, the data sets for the logically related databases also need to be included.

UNLOAD
Defines the output data set for the IMS segments that the function extracts.

SYSPRINT
Defines the output data set for the messages that the Extract issues.

FMIMSIN control statements: You use the FMIMSIN control statements to specify:

- If the Extract is to run in a BMP or DLI region.
- Whether to use a static PSB, or if FM/IMS is to generate a dynamic PSB to extract the IMS data.
- Whether a Keys file is used to perform a partial Extract.
- Whether selection criteria have been specified to perform a partial Extract.

  If the database contains logical relationships, the criteria can be used to determine the logical relationships that are to be chased.

Each line in FMIMSIN starts with a $$FILEM and the first line contains the parameter IXB to specify that FM/IMS is performing an Extract.

```
$$FILEM IXB REGNTYPE=DLI,
$$FILEM     PSBTYPE=DYNAMIC,
$$FILEM     DBDDSN=FMN.IMS.IVP.DBDLIB,
. . .
. . .
```

The syntax of the control statements for an Extract is shown below.

Syntax - part 1

```
                     (1)
►►──IXB──IMSID=imsid───────┬─REGNTYPE=BMP─┤ BMP parameters ├─┬─────────────────────────────────►
                           └─REGNTYPE=DLI─┤ DLI parameters ├─┘

                                      (2)
   ┌─PSBTYPE=installation default──────────┐
►──┼───────────────────────────────────────┼─DBDDSN=dbddsn──┬──────────────────┬───────────────►
   ├─PSBTYPE=DYNAMIC─┤ DYNAMIC parameters ├─┤                └─DBDDSN2=dbddsn2──┘
   └─PSBTYPE=STATIC─┤ STATIC parameters ├───┘

                 ┌─CRITERIA=N─────────────────────────────────────────┐   ┌─KEYEXT=N─┐
►─DBDMEM=dbdname─┼─────────────────────────────────────────────────────┼───┼──────────┼─────────►
                 │            ┌─CRITDD=CRITDD──┐                       │   └─KEYEXT=Y─┘
                 └─CRITERIA=Y─┼────────────────┼─┬─────────────────┬───┘
                              ├─CRITDD=ddname──┤ └─CRITMEM=critmem─┘
                              └─CRITDSN=critdsn┘

   ┌─SDEPTSO=N─┐
►──┼───────────┼──┤ logical keys dataset parameters ├──┤ root keys dataset parameters ├───────►◄
   └─SDEPTSO=Y─┘
```

BMP parameters:

```
                                            (3)                          (3)
                  ┌─PARDLI=subsystem default───┐  ┌─IMSNBA=subsystem default──┐
├──┬──────────┬───┼────────────────────────────┼──┼───────────────────────────┼────────────────►
   └─AGN=agn──┘   ├─PARDLI=0───────────────────┤  └─IMSNBA=number─────────────┘
                  └─PARDLI=1───────────────────┘

                              (3)                            (3)
   ┌─IMSOBA=subsystem default──┐  ┌─LOCKMAX=subsystem default──┐
►──┼───────────────────────────┼──┼────────────────────────────┼──────────────────────────────┤
   └─IMSOBA=number─────────────┘  └─LOCKMAX=number─────────────┘
```

Notes:

1 If REGNTYPE=DLI is specified and the FM/IMS installation options module specifies COMPAT=Y, IMSID is an optional keyword. Otherwise, IMSID is a required keyword (as shown).

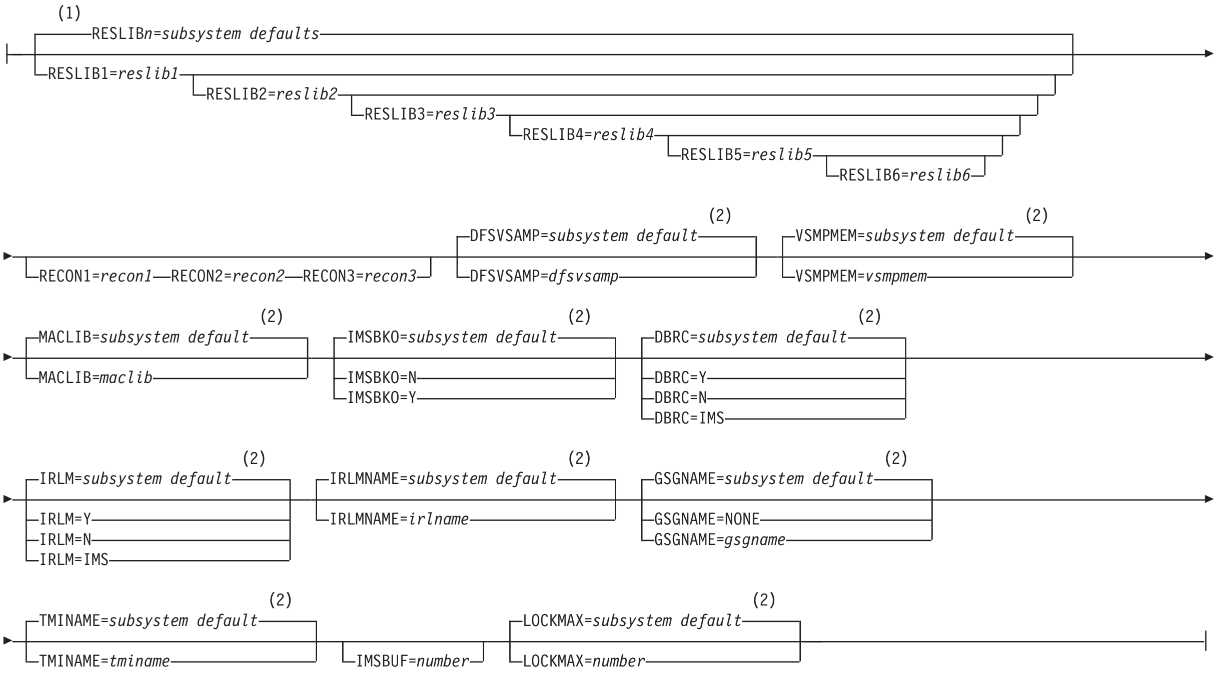
2 If you do not specify PSBTYPE, FM/IMS uses the PSBTYPE specified in the FM/IMS installation options module.

3 If you do not specify this parameter, FM/IMS uses the *subsystem* default for the parameter (if you have specified the IMSID parameter), or the *installation* default for the parameter (if you have not specified the IMSID parameter).

The FM/IMS Administrator specifies the subsystem and installation defaults at installation time.

Syntax - part 2

DLI parameters:



Notes:

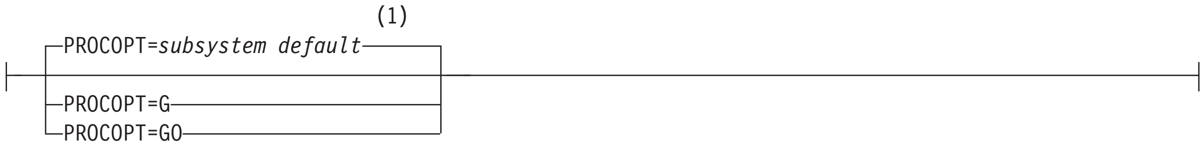
1 If you do not specify the RESLIB parameters, FM/IMS uses the *subsystem* defaults for the parameters (if you have specified the IMSID parameter), or the *installation* defaults for the parameters (if you have not specified the IMSID parameter).

2 If you do not specify this parameter, FM/IMS uses the *subsystem* default for the parameter (if you have specified the IMSID parameter), or the *installation* default for the parameter (if you have not specified the IMSID parameter).

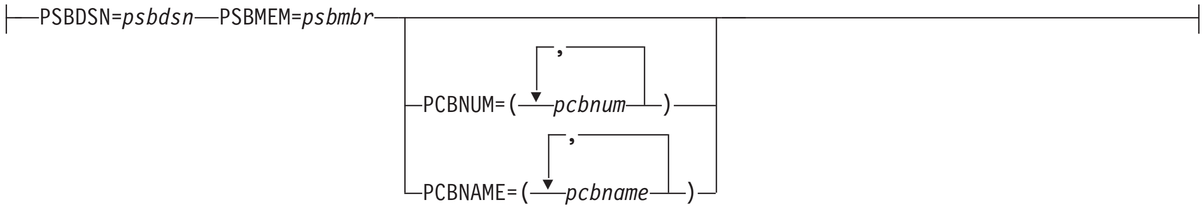
The FM/IMS Administrator specifies the subsystem and installation defaults at installation time.

Syntax - part 3

DYNAMIC parameters:



STATIC parameters:



Notes:

1 If you do not specify this parameter, FM/IMS uses the *subsystem* default for the parameter (if you have specified the IMSID parameter), or the *installation* default for the parameter (if you have not specified the IMSID parameter).

The FM/IMS Administrator specifies the subsystem and installation defaults at installation time.

Syntax - part 4

logical keys dataset parameters:

```
                                                                          (1)
 ┌─LKUNIT=TRK─┐  ┌─LKPQTY=20─────┐  ┌─LKSQTY=20─────┐  ┌─LKMGMTCL=installation default──────┐
├─┼────────────┼──┼───────────────┼──┼───────────────┼──┼────────────────────────────────────┼──────►
  ├─LKUNIT=BLK─┤  └─LKPQTY=number─┘  └─LKSQTY=number─┘  └─LKMGMTCL=management-class-name─────┘
  ├─LKUNIT=CYL─┤
  ├─LKUNIT=KB──┤
  └─LKUNIT=MB──┘

                                (1)                             (1)
 ┌─LKSTORCL=installation default──┐  ┌─LKDATACL=installation default──┐
►─┼────────────────────────────────┼──┼────────────────────────────────┼──────────────────────────►
  └─LKSTORCL=storage-class-name────┘  └─LKDATACL=data-class-name───────┘

►─┬─────────────────┬──┬─────────────────┬──┬─────────────────┬──────────────────────────────────┤
  └─LKVOL1=volser───┘  └─LKVOL2=volser───┘  └─LKVOL3=volser───┘
```

root keys dataset parameters:

```
                                                                          (1)
 ┌─RKUNIT=CYL─┐  ┌─RKPQTY=10─────┐  ┌─RKSQTY=10─────┐  ┌─RKMGMTCL=installation default──────┐
├─┼────────────┼──┼───────────────┼──┼───────────────┼──┼────────────────────────────────────┼──────►
  ├─RKUNIT=BLK─┤  └─RKPQTY=number─┘  └─RKSQTY=number─┘  └─RKMGMTCL=management-class-name─────┘
  ├─RKUNIT=TRK─┤
  ├─RKUNIT=KB──┤
  └─RKUNIT=MB──┘

                                (1)                             (1)
 ┌─RKSTORCL=installation default──┐  ┌─RKDATACL=installation default──┐
►─┼────────────────────────────────┼──┼────────────────────────────────┼──────────────────────────►
  └─RKSTORCL=storage-class-name────┘  └─RKDATACL=data-class-name───────┘

►─┬─────────────────┬──┬─────────────────┬──┬─────────────────┬──────────────────────────────────┤
  └─RKVOL1=volser───┘  └─RKVOL2=volser───┘  └─RKVOL3=volser───┘
```

Notes:

1 If you do not specify this parameter, FM/IMS uses the *installation* default specified in the FM/IMS installation option module.

The FM/IMS Administrator specifies the installation defaults at installation time.

IMSid=*imsid*

The 4-character ID of the IMS subsystem in which the databases you want to access are defined.

Note: If REGNTYPE=DLI is specified and the FM/IMS installation options module specifies COMPAT=Y, then this is an optional parameter. Otherwise, it is a required parameter.

REGNTYPE
Required parameter. Specifies the type of region that you want the function to run in.

BMP For a BMP region. You can use a BMP region when the database is online.

DLI For a DL/I batch processing region. You can use a DL/I batch processing region when the database is offline, or there is data sharing support.

PSBTYPE
Optional parameter. Specifies the type of PSB that the function uses to access the databases.

DYNAMIC
The function uses a temporary PSB that is generated when it is started and deleted when it finishes.

STATIC
The function uses an existing PSB generated by the database administrator. Details of which PSB to use must be specified in the STATIC parameters.

DBDDSN=*dbddsn*
Required parameter. The name of the load library that contains the database definitions (DBDs) that you want FM/IMS and IMS to use.

DBDDSN2=*dbddsn2*
Optional parameter. A second DBD library that is searched when FM/IMS or IMS cannot find the DBD in the first DBD library.

When you are using a dynamic PSB, IMS requires the primary database of the function, and the DBDs of all the databases that are logically related to the primary database, to be in at least one of the specified DBD libraries.

When you are using a static PSB, IMS requires the following DBDs to be in at least one of the specified DBD libraries:

- The DBD for each database that is specified in the PSB
- The DBD for each database that is logically related to a database specified in the PSB

DBDMEM=*dbdname*
Required parameter. The name of the primary database of the function. When you are using a static PSB, you must specify a database that the PSB has access to.

CRITERIA
Optional parameter. Specifies whether or not the Extract uses selection criteria.

N The Extract does not use selection criteria.

Y The Extract uses selection criteria. If Y is specified, you must specify the criteria set that the extract is to use as outlined in the CRITDD, CRITDSN and CRITMEM entries below.

CRITDD=*ddname*
Identifies the DD statement that specifies the sequential data set or PDS(E) and member name that contains the criteria you want your Extract to use. The default is CRITDD.

CRITDSN=*critdsn*

The name of the data set that contains the criteria that you want your Extract to use. You may specify a sequential data set or a PDS(E). If you specify a PDS(E), the parameter may also include a member name in parentheses.

CRITMEM=*critmem*

Required when the CRITDSN or CRITDD parameter specifies a PDS(E) and does not specify the member. Specifies the member of the PDS(E) that contains the criteria you want your Extract to use.

Note: The parameters DSNCRIT and MEMCRIT will continue to be supported for backward compatibility.

KEYEXT

Optional parameter. Specifies whether or not the Extract is restricted to those records that are specified in a key values data set.

N The Extract does not use a key values data set.

Y The Extract uses a key values data set. The key values data set contains the key values of the root segments of the records that you want extracted.

You specify the key values data set in the KEYSFILE DD statement. You may specify a sequential data set or a PDS(E) member.

SDEPTSO

This option is only applicable for DEDBs with Sequential Dependent (SDEP) segments. Specifies whether or not FM/IMS is to write the SDEP timestamp to the unload file.

N Default value. All of the sequential dependents of a root segment are physically loaded together in the reverse sequence they were read during the Extract process. This restores the SDEP segments so that, when reading them using DLI calls, the database looks the same. However when the DEDB database is processed using the Sequential Dependent Scan Utility, the order of the segments could be different.

Y FM/IMS is to write the SDEP timestamp to the unload file. When loading the DEDB database, the sequential dependent segments are restored in their original entry sequence order. This causes an increase in elapse time and I/O for the Extract job.

BMP parameters

The BMP parameters are used when the function is run in BMP mode (REGNTYPE=BMP).

AGN=*agn* If the subsystem specified in the IMSID parameter uses AGNs, then this is a required parameter. Otherwise, this parameter should not be specified. Specify an AGN that provides you with the access your function requires and that you have authority to use. (The BMP region that FM/IMS starts is only authorized to use the IMS resources that are associated with the specified AGN.)

PARDLI Optional parameter. Specifies the parallel DL/I option that the function is to use.

0 DL/I processing is performed in the BMP region.

1 All DL/I processing for the BMP is performed in the IMS control region.

PARDLI=1 prevents control region system 113 abends resulting from system X22 abends in the BMP region. However, if PARDLI=1, then parallel DL/I is disabled. This can degrade performance.

IMSNBA=*number*

Optional parameter. Specifies the number of Fast Path database buffers to be made available in the Common Service Area when a Fast Path region is activated. The number specified is used by the function when it accesses Fast Path databases. The number you specify must be in the range 1 to 9999.

IMSOBA=*number*

Optional parameter. Specifies the number of additional page-fixed buffers to be made available to a Fast Path region if the normal allotment (specified in the IMSNBA parameter) is used. The number specified is used by the function when it accesses Fast Path databases. The number you specify must be in the range 1 to 9999.

LOCKMAX=number

Optional parameter. Specifies the maximum number of locks (in units of 1000) that the function is allowed to hold at one time. The number you specify must be in the range 0 to 32767. If 0 is specified, there will be no limit on the number of locks the function can hold at one time.

DLI parameters

The DLI parameters are used when the function is run in DLI mode (REGNTYPE=DLI).

RESLIB*n***=***reslibn*

(Where *n* is a number in the range 1 to 6.) Specifies the names of the data sets containing the IMS SVC modules, the DFSMDA dynamic allocation modules and the IMS exit routines. The specified data sets are allocated to the TASKLIB DD and those data sets that are APF-authorized are allocated to the DFSRESLB DD. Optional parameter.

RECON1=*recon1*

Optional parameter. If you want IMS to dynamically allocate the DBRC RECON data sets, do not specify this parameter. IMS allocates the RECON data sets specified in the DFSMDA dynamic allocation modules when they are required. If you want FM/IMS to allocate the DBRC RECON data sets, specify the name of the primary DBRC RECON data set that you want IMS to use. FM/IMS allocates the specified data set to the RECON1 DD when it is required. If you specify the RECON1 parameter, you must also specify the RECON2 and RECON3 parameters.

RECON2=recon2

Optional parameter. If you want IMS to dynamically allocate the DBRC RECON data sets, do not specify this parameter. IMS allocates the RECON data sets specified in the DFSMDA dynamic allocation modules when they are required. If you want FM/IMS to allocate the DBRC RECON data sets, specify the name of the secondary DBRC RECON data set that you want IMS to use. FM/IMS allocates the specified data set to the RECON2 DD when it is required. If you specify the RECON2 parameter, you must also specify the RECON1 and RECON3 parameters.

RECON3=recon3
Optional parameter. If you want IMS to dynamically allocate the DBRC RECON data sets, do not specify this parameter. IMS allocates the RECON data sets specified in the DFSMDA dynamic allocation modules when they are required. If you want FM/IMS to allocate the DBRC RECON data sets, specify the name of the spare DBRC RECON data set that you want IMS to use. FM/IMS allocates the specified data set to the RECON3 DD when it is required. If you specify the RECON3 parameter, you must also specify the RECON1 and RECON2 parameters.

DFSVSAMP=*dfsvsamp*
Optional parameter. Specifies the name of the data set containing the parameters that define the IMS buffer subpools for OSAM and VSAM data sets. The specified data set is allocated to the DFSVSAMP DD. If the specified data set is partitioned, specify the required member in the VSMPMEM parameter.

VSMPMEM=*vsmpmem*
Optional parameter. When the DFSVSAMP data set that the function is to use is a PDS or PDSE, VSMPMEM specifies the member of that data set containing the parameters that define the IMS buffer subpools for OSAM and VSAM data sets. The specified member is allocated to the DFSVSAMP DD. If the DFSVSAMP data set is sequential, do not specify this parameter.

MACLIB=*maclib*
Optional parameter. Specifies the name of the IMS macro library. FM/IMS uses the IMS macros in the specified library when generating a dynamic PSB.

IMSBKO
Optional parameter. Specifies whether or not a dynamic backout is to be performed when an IMS pseudoabend occurs.

N A dynamic backout is not performed when an IMS pseudoabend occurs.

Y A dynamic backout is performed when an IMS pseudoabend occurs.

This parameter only takes effect when the function is running with an IMS log data set.

DBRC
Optional parameter. Specifies whether or not the function is to use Database Recovery Control:

Y DBRC is used.

N DBRC is not used unless the IMSCTRL macro statement specifies DBRC=FORCE.

IMS DBRC is used if the IMSCTRL macro statement specifies DBRC=YES or DBRC=FORCE. Otherwise, DBRC is not used.

IRLM Optional parameter. Specifies whether or not the function uses an Internal Resource Lock Manager (IRLM).

N IRLM is not used unless the IMSCTRL macro statement specifies IRLM=Y and an IRLMNM.

Y IRLM is used.

IMS IRLM is used if either of these are true:
- The IMSCTRL macro statement specifies IRLM=Y.
- The IMSCTRL macro statement specifies an IRLMNM and does not specify IRLM=N.

IRLM is not used if either of these are true:
- The IMSCTRL macro statement specifies IRLM=N.
- The IMSCTRL macro statement does not specify a value for IRLM and IRLMNM.

IRLMNAME=*irlm_name*
Optional parameter. Specifies the 4-byte z/OS subsystem name assigned to the Internal Resource Lock Manager (IRLM). The function passes the name to the IMS region controller when IRLM=Y or IRLM=IMS is specified.

RSR No longer used by FM/IMS but is retained for compatibility. FM/IMS accepts RSR=Y or RSR=N.

GSGNAME If the IMS subsystem imsid is part of a Remote Site Recovery (RSR) complex and the GSGNAME has not been specified in the IMSCTRL macro statement or the FM/IMS installation options module, and you want any activity performed by the function to be tracked by RSR, specify the Global Service Group (GSG) name for the RSR complex.

If the IMS subsystem is part of an RSR complex and the GSGNAME has been specified in the IMSCTRL macro statement or the FM/IMS installation options module and you do not want the activity of the function to be tracked, specify NONE.

Otherwise, do not specify this parameter.

TMINAME=*tminame*
Optional parameter. If the IMS subsystem is part of a Remote Site Recovery (RSR) complex and you want the activity performed by the function tracked by RSR, specify the Transport Manager Instance (TMI) name that the function is to use.

IMSBUF=*number*
Optional parameter. Specifies the number of 1K blocks to be used in calculating the size of the OSAM pool. The value specified in this field is only used if the DFSVSAMP data set does not include any IOBF control statements.

LOCKMAX=*number*
Optional parameter. Specifies the maximum number of locks (in units of 1000) that the function is allowed to hold at one time. The number you specify must be in the range 0 to 32767. If 0 is specified, there will be no limit on the number of locks the function can hold at one time.

DYNAMIC parameters
The DYNAMIC parameters are used when the function uses a dynamic PSB (PSBTYPE=DYNAMIC).

PROCOPT
Optional parameter. Specifies the PSB processing option (PROCOPT) that the dynamic PSB uses to read the database.

G The function reads with integrity.

GO The function reads without integrity.

STATIC parameters
The STATIC parameters are used when the function uses a static PSB (PSBTYPE=STATIC).

PSBDSN=*psbdsn*
Required parameter. The name of the PSB load library that contains the PSB that you want the function to use. The parameter may also include a member name in parentheses. If a member is specified here, the PSBMEM parameter must not be specified.

PSBMEM=*psbmbr*
Required when a member name has not been specified in the PSBDSN parameter. The name of the PSB that you want the function to use.

PCBNUM=*pcbnum*
Optional parameter. Specifies the relative numbers (not including the I/O PCB) of the PCBs in the PSB used by the function to read the primary database and (when required) the logically related databases.

PCBNAME=*pcbname*
Optional parameter. Specifies the names of the PCBs in the PSB used by the function to read the primary database and (when required) the logically related databases.

logical keys data set parameters
The logical keys data set is a VSAM work data set for the keys of the logically related segments that are to be processed by the Extract. These optional parameters allow you to specify allocation parameters for this data set:

LKUNIT
Defines the unit of the primary and secondary space allocation:
TRK Specifies that the space be allocated in tracks.
BLK Specifies that the space be allocated in blocks.
CYL Specifies that the space be allocated in cylinders.
KB Specifies that the space be allocated in kilobytes.
MB Specifies that the space be allocated in megabytes.

LKPQTY
Specifies the primary space allocation for the data set.

LKSQTY
Specifies the secondary space allocation for the data set.

LKMGMTCL
Specifies a management class for the data set. Do not specify this parameter on a system without SMS.

LKSTORCL
Specifies a storage class for the data set. Do not specify this parameter on a system without SMS.

LKDATACL
Specifies a data class for the data set. Do not specify this parameter on a system without SMS.

LKVOL*n*
(Where *n* is a number in the range 1 to 3.) Specifies the serial number of the volume that the data set is to be placed on. Use on systems without SMS.

Root keys data set parameters
The root keys data set is a VSAM work data set for the root keys of the primary and logically related databases that have been processed by the Extract. These optional parameters allow you to specify allocation parameters for this data set:

RKUNIT
Defines the unit of the primary and secondary space allocation:

TRK	Specifies that the space be allocated in tracks.
BLK	Specifies that the space be allocated in blocks.
CYL	Specifies that the space be allocated in cylinders.
KB	Specifies that the space be allocated in kilobytes.
MB	Specifies that the space be allocated in megabytes.

RKPQTY
Specifies the primary space allocation for the data set.

RKSQTY
Specifies the secondary space allocation for the data set.

RKMGMTCL
Specifies a management class for the data set. Do not specify this parameter on a system without SMS.

RKSTORCL
Specifies a storage class for the data set. Do not specify this parameter on a system without SMS.

RKDATACL
Specifies a data class for the data set. Do not specify this parameter on a system without SMS.

RKVOL*n*
(Where *n* is a number in the range 1 to 3.) Specifies the serial number of the volume that the data set is to be placed on. Use on systems without SMS.

Database Extract Statistics report:

```
IBM File Manager for z/OS IMS Component                                                                       Page 2
                                             Database Extract Statistics Report                               Date: 2006-09-01
                                                                                                              Time: 11.07.24

                                     -Segm Length- ----- Extract Statistics ------  Logically
               Database Segment Level    Min   Max      Read   Rejected  Extracted  Related DB
               --------------------------------------------------------------------------------
               DJ2E     SUBURB    01      62    62        87          0         87
                        LINKSTR   02      28    28      4357          0       4357  DJ3E
                        NSTREET   02      30    30        32          0         32
                        INSHIRE   02      20    20        87          0         87  DJ1E
                        LETRBOX   02      34    34        30          0         30
   Subtotal    DJ2E                                     4593          0       4593

               DJ3E     STREET    01      32    32      3293          0       3293
                        INSUB     02      18    18      4357          0       4357  DJ2E
   Subtotal    DJ3E                                     7650          0       7650

               DJ1E     SHIRE     01      53    53        13          0         13
                        SHIRENP   02      11    11        11          0         11
                        LINKSUB   02      18    18        87          0         87  DJ2E
   Subtotal    DJ1E                                      111          0        111

   Grand total                                         12354          0      12354

                                             *** End of IMS Extract Report ***
```

Figure 4. Database Extract Statistics

Database
The name of the DBD used for the Database Extract Statistics.

Segment
The name of the segment being extracted.

Level The level of the segment within the IMS Hierarchical structure.

Segment Length
Min - the minimum length of the segment as defined in the DBD.

Max - the maximum length of the segment as defined in the DBD.

For fixed length segments, the minimum length and maximum length have the same value.

Extract Statistics
Read - the number of segment occurrences read from the primary database for this segment.

For the primary database root segment, if Use Key value is "**Y**", this field represents the number of records read from the key value file.

Rejected - the number of segment occurrences read but rejected in relation to the field selection criteria within the primary database for this segment. Key value extractions might also have records rejected if there are erroneous keys specified in the key values data set.

Extracted - the number of segment occurrences read and extracted from the primary database for this segment.

Logically Related DB
The name of the DBD which is the logical parent of this segment.

Subtotal
The total number of segments read, rejected and extracted for this Database.

Grand Total
The total number of segments read, rejected and extracted for all Databases.

Example: Here is an example of a full Extract of an IMS database in DLI mode using a static PSB.

```
//FMNIXTRT 'Static PSB extract'
//*
//FILEMAN  EXEC PGM=FMN1IMSB
//STEPLIB  DD DSN=FMN.SFMNMOD1,DISP=SHR
//SYSPRINT DD SYSOUT=*,RECFM=FBA,LRECL=133
//FMNTSPRT DD SYSOUT=*
//FMN1JIN  DD DISP=SHR,DSN=FMN.SFMNSLIB
//FMIMSIN  DD *
$$FILEM IXB REGNTYPE=DLI,
$$FILEM     PSBTYPE=STATIC,
$$FILEM     PSBDSN=FMN.IMS.IVP.PSBLIB,
$$FILEM     PSBMEM=PUPA,
$$FILEM     PCBNUM=(3,2,1),
$$FILEM     DBDDSN=FMN.IMS.IVP.DBDLIB,
$$FILEM     DBDDSN2=IMSV910.FMI.DBDLIB,
$$FILEM     DBDMEM=DJ2E,
$$FILEM     CRITERIA=N,
$$FILEM     IMSID=IF42,
$$FILEM     LKUNIT=CYL,
$$FILEM     LKPQTY=10,
$$FILEM     LKSQTY=10,
$$FILEM     RKUNIT=CYL,
$$FILEM     RKPQTY=10,
$$FILEM     RKSQTY=10,
$$FILEM     KEYEXT=N
 /*
//REPORT   DD SYSOUT=*
//UNLOAD   DD DSN=FMN.IMS.EXTRACT.DJ2E,
//         DISP=(,CATLG,DELETE),
//         SPACE=(CYL,(1,1,0),RLSE),
//         DCB=(RECFM=VB,LRECL=00150,BLKSIZE=0)
//* -------------------------------------------------------------
//* -- Database data sets                                      --
//* -------------------------------------------------------------
//DJ2E DD DSN=FMN.IMS.IVP.DB.DJ2E,DISP=SHR
//DJ2F DD DSN=FMN.IMS.IVP.DB.DJ2F,DISP=SHR
//DJ3E DD DSN=FMN.IMS.IVP.DB.DJ3E,DISP=SHR
//DJ3F DD DSN=FMN.IMS.IVP.DB.DJ3F,DISP=SHR
//DJ1E DD DSN=FMN.IMS.IVP.DB.DJ1E,DISP=SHR
//DJ1F DD DSN=FMN.IMS.IVP.DB.DJ1F,DISP=SHR
```

Load (ILB):

Purpose
The Load dialog generates JCL that runs the Load (ILB) function. This function loads previously extracted data from a sequential data set into a database. You can edit the JCL and ILB input parameters before you submit the job, or you can create your own batch job to run the ILB function.

Usage notes
The Load JCL generated by FM/IMS consists of several steps depending on the parameters you have selected. The steps are:

- Delete/Define of the Database data sets. See Note 1.
- DBRC step to initialize the HALDB databases. Only included if the database is a HALDB. See Note 1 and Note 2.

- IMS PREREORG step to obtain the DBD logical and secondary index information. See Note 1 and Note 2.
- Step to sort the Load data into the correct order.
- FM/IMS step to load the IMS data. This is described in more detail below.
- IMS Prefix Resolution step to resolve the IMS pointers for logical relationships and secondary indexes. See Note 1 and Note 2.
- IMS secondary index unload step. See Note 1 and Note 2.
- IMS secondary index reload step. See Note 1 and Note 2.
- IMS prefix update step. See Note 1 and Note 2.

Notes:

1. This step is only included if you are loading using a dynamic PSB in DLI mode, have a load type of ″Insert-Load PSB (If DLI)″ and on the Load options panel specified ″Delete/Define IMS data sets″.
2. For more information on this step, see the *IMS Utilities Reference: Database and Transaction Manager*.

FM/IMS Load step description: The FM/IMS Load step has the following flows:

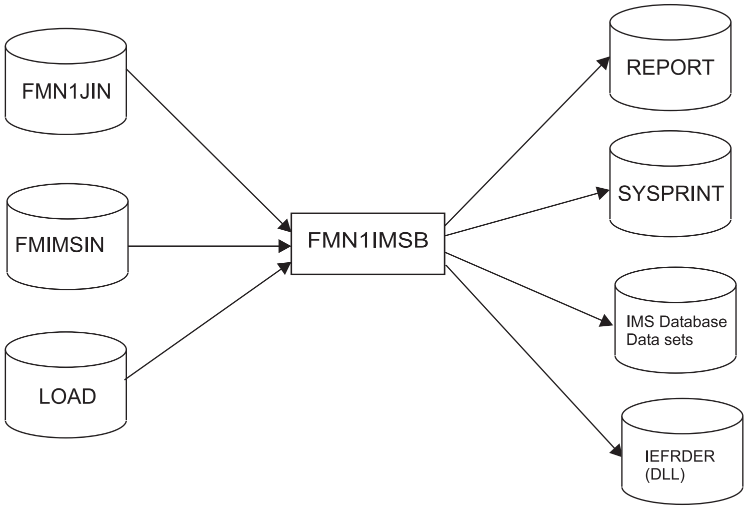


Figure 5. Flows for Load (ILB) step

FMN1JIN
Points to the FM/IMS skeleton library. This DD statement is only used when you are loading data into HALDB databases.

FMIMSIN
Contains the control statements for the Load. These control statements are discussed below.

REPORT
Defines the output data set for the Database Load Summary report.

IMS database data sets
In DLI mode, FM/IMS includes DDs for each of the IMS database data sets associated with the IMS database being loaded. If the database has logical relationships, the data sets for the logically related databases need to be included.

LOAD
Contains the IMS segments that are to be loaded by the function. In the job generated by FM/IMS, this is a temporary data set created by the sort step.

SYSPRINT
Defines the output data set for:

- The messages that the Load issues.
- The Load Diagnostic report that lists the IMS segments that could not be loaded.

IEFRDER
Defines the IMS log data sets that the Load is to use. Only required when running in DLI mode.

FMIMSIN control statements: You use the FMIMSIN control statements to specify:

- If the Load is to run in a BMP or DLI region.
- Whether to use a static PSB, or if FM/IMS is to generate a dynamic PSB to load the IMS data.
- How the data is to be loaded.
- Load checkpoint frequency.

Each line in FMIMSIN starts with a $$FILEM and the first line contains the parameter ILB to specify that FM/IMS is performing a Load.

```
$$FILEM ILB REGNTYPE=DLI,
$$FILEM     PSBTYPE=DYNAMIC,
$$FILEM     DBDDSN=FMN.IMS.IVP.DBDLIB,
. . .
. . .
```

The syntax of the control statements for a Load is shown here.

Syntax - part 1

```
                          (1)
►►──ILB──IMSID=imsid──────────┬─REGNTYPE=BMP─┤ BMP parameters ├─┬──────────────────────────────►
                              └─REGNTYPE=DLI─┤ DLI parameters ├─┘

                                      (2)
   ┌─PSBTYPE=installation default──────────┐
►──┼───────────────────────────────────────┼──DBDDSN=dbddsn──┬──────────────────────┬──DBDMEM=dbdname──►
   ├─PSBTYPE=DYNAMIC───────────────────────┤                 └─DBDDSN2=dbddsn2──────┘
   └─PSBTYPE=STATIC─┤ STATIC parameters ├──┘

                                                          (3)
                          ┌─CHKPFREQ=subsystem default──────┐
►─LOADTYPE=──┬─1─┬──LDDSN=lddsn──┼─────────────────────────────────┼─────────────────────────►◄
             ├─2─┤               └─CHKPFREQ=chkpfreq───────────────┘
             └─3─┘
```

BMP parameters:

```
                                          (3)                                 (3)
                  ┌─PARDLI=subsystem default──┐  ┌─IMSNBA=subsystem default──┐
├──┬───────────┬──┼───────────────────────────┼──┼───────────────────────────┼──────────────►
   └─AGN=agn───┘  ├─PARDLI=0──────────────────┤  └─IMSNBA=number────────────┘
                  └─PARDLI=1──────────────────┘

                                    (3)                              (3)
   ┌─IMSOBA=subsystem default──────┐  ┌─LOCKMAX=subsystem default──────┐
►──┼───────────────────────────────┼──┼────────────────────────────────┼──────────────────────┤
   └─IMSOBA=number─────────────────┘  └─LOCKMAX=number─────────────────┘
```

Notes:

1 If REGNTYPE=DLI is specified and the FM/IMS installation options module specifies COMPAT=Y, IMSID is an optional keyword. Otherwise, IMSID is a required keyword (as shown).

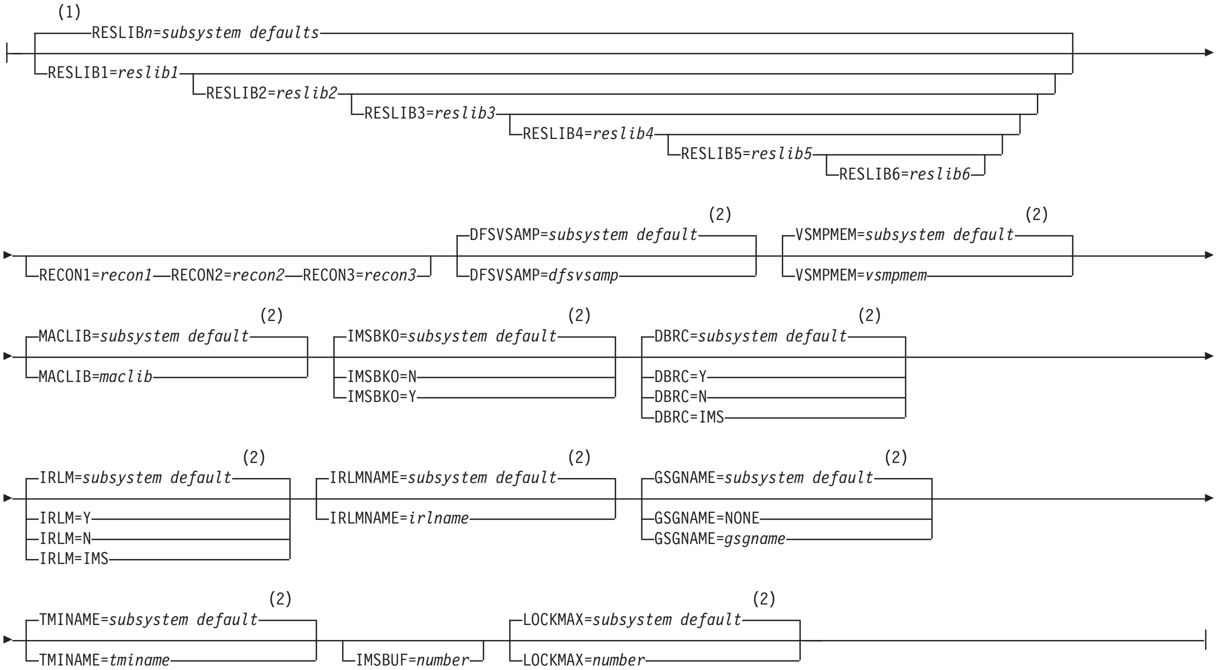
2 If you do not specify PSBTYPE, FM/IMS uses the PSBTYPE specified in the FM/IMS installation options module.

3 If you do not specify this parameter, FM/IMS uses the *subsystem* default for the parameter (if you have specified the IMSID parameter), or the *installation* default for the parameter (if you have not specified the IMSID parameter).

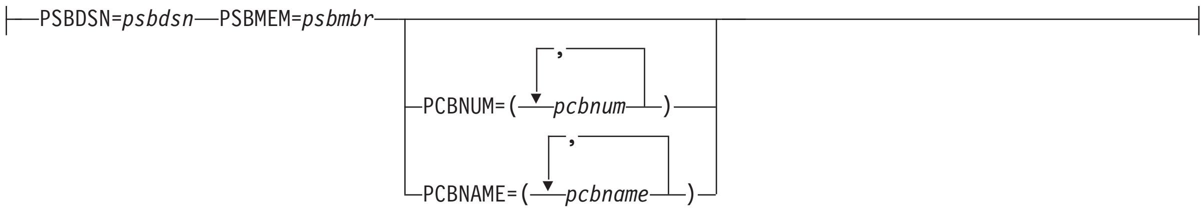
The FM/IMS Administrator specifies the subsystem and installation defaults at installation time.

Syntax - part 2

DLI parameters:



STATIC parameters:



Notes:

1 If you do not specify the RESLIB parameters, FM/IMS uses the *subsystem* defaults for the parameters (if you have specified the IMSID parameter), or the *installation* defaults for the parameters (if you have not specified the IMSID parameter).

2 If you do not specify this parameter, FM/IMS uses the *subsystem* default for the parameter (if you have specified the IMSID parameter), or the *installation* default for the parameter (if you have not specified the IMSID parameter).

The FM/IMS Administrator specifies the subsystem and installation defaults at installation time.

The LOADTYPE, LDDSN and CHKPFREQ parameters are described here. All the other parameters in this syntax diagram can also be specified for the Extract (IXB) function. For a description of these common parameters, see “Extract (IXB)” on page 71.

LOADTYPE

Required parameter. Specifies:

- Whether or not the Load replaces existing segments in the database.
- Whether or not the load uses a load PCB.

1 Specifies that the Load is not to replace existing segments in the database. If you select this option, the load only attempts to insert the segment. If the insert call receives an II status code, the Load does not replace the existing segment in the database with the load segment.

2 Specifies that the Load is to replace existing segments in the database. If you select this option, the Load attempts to insert the segment, as it does when option 1 is selected. However, if the insert call receives an II status code, the Load replaces the existing segment in the database with the load segment. Note that if a segment has a non-unique key or no key, existing segments in the database are not replaced. The load segments are inserted even when there is an existing segment in the database with the same key.

3 Specifies that the Load uses a load PCB (PROCOPT=L/ LS) to load the segments into the database. Use this option for an initial load of a database.

Note: This option is only available when you specify PSBTYPE=DYNAMIC and REGNTYPE=DLI.

LDDSN=*lddsn*

Required parameter. The name of the data set that contains the data that you want loaded.

CHKPFREQ

Optional parameter. Specifies the frequency of the automatic save function during the Load; that is, the Load checkpoint frequency. FM/IMS:

- Increments a count by 1 each time the Load inserts or replaces a segment.
- Issues a checkpoint when the count is equal to the Load checkpoint frequency.

  The valid range is 1 to 99999.

Database Load Statistics report:

```
IBM File Manager for z/OS IMS Component                                                              Page 2
                                                  Database Load Statistics Report                    Date: 2004-03-24
                                                                                                     Time: 13.13.38
                                                 4         5         6
                 1   2     3                          -------------Segments-----------
               DBD Name  Level  Segment        Read     Loaded   Rejected
               DJ2E          1  SUBURB            1          1          0
                             2  LINKSTR         198        198          0
                             2  NSTREET           3          3          0
                             2  INSHIRE           1          1          0
      7                      2  LETRBOX           1          1          0
   Subtotal    DJ2E                             204        204          0

               DJ3E          1  STREET          198        198          0
      7                      2  INSUB           454        454          0
   Subtotal    DJ3E                             652        652          0

               DJ1E          1  SHIRE             1          1          0
                             2  SHIRENP           1          1          0
      7                      2  LINKSUB           9          9          0
   Subtotal    DJ1E                              11         11          0
      8
   Grand total                                  867        867          0
```

Figure 6. Database Load Statistics Report

Here is a description of the columns of this report:

1 The DBD name of each database that was loaded.

The primary database is listed first.

2 The level of the segment type in the database hierarchy.

3 The name of the segment.

4 The number of segment occurrences of this type read from the input (Load from) file.

5 The number of segment occurrences of this type that were loaded.

6 The number of segment occurrences of this type that were not loaded.

If a segment occurrence was not loaded, a line in the Diagnostic Report explains why the segment occurrence was rejected.

7 The total number of segments read/loaded/rejected for this database.

8 The total number of segments read/loaded/rejected (all databases).

Database Load Summary report:

```
IBM File Manager for z/OS IMS Component                                             Page 1
                                                  Database Load Summary             Date: 2004-03-24
                                                                                    Time: 13.13.38
Primary database  DJ2E    1
Load
   From           FMNUSER.EXTRACT.DJ2E    2
   To             FMN.IMS.IVP.DB.DJ2E   3
DBD
   Data set name  FMN.IMS.IVP.DBDLIB     4
   Member         DJ2E                   5
Region Type       DLI                    6
Processing Options:
   Database Load  Insert or Replace     7
   IMS log usage  Use but delete if job end successfully   8
Databases loaded: 9       10        11
               DBD Name  ddname    Data set name
               --------  --------  ------------------------------------
               DJ2E      DJ2E      FMN.IMS.IVP.DB.DJ2E
               DJ2F      DJ2F      FMN.IMS.IVP.DB.DJ2F
               DJ3E      DJ3E      FMN.IMS.IVP.DB.DJ3E
               DJ3F      DJ3F      FMN.IMS.IVP.DB.DJ3F
               DJ1E      DJ1E      FMN.IMS.IVP.DB.DJ1E
               DJ1F      DJ1F      FMN.IMS.IVP.DB.DJ1F
```

Figure 7. Database Load Summary

The first part of the Database Load summary provide the following details about the load:

1 The DBD name of the primary database that was loaded.

2 The name of data set that contains the data that was loaded.

3 The primary database data set name.

4 The name of the DBD load library that contains the DBD of the primary database that was loaded.

5 The DBD name of the primary database that was loaded.

6 The region type that the load job ran in (either DLI or BMP). If the region type is BMP the IMS region is also shown.

7 The database load processing option. One of:

Insert only
The load job did not replace any existing segment occurrences in the databases being loaded.

Insert or replace
If a segment occurrence cannot be inserted, because a segment occurrence with this key already existed in the database, the existing segment occurrence was replaced.

8 IMS log usage processing option. One of:

Use and catalog at EOJ
The load job used an IMS log tape and it was cataloged at the end of the job.

Use but delete if job ends successfully
The load job used an IMS log tape but it was deleted if the job ran successfully.

Do not use
An IMS log tape was not used for the Load.

9 The DBD name of each of the databases loaded.

The primary database is listed first.

10 The ddname(s) of the database.

11 The database data set name.

Database Load Diagnostic report:

```
IBM File Manager for z/OS IMS Component                                                               Page  1
                                                  Database Load Diagnostic Report                     Date: 2004-03-24
                                                                                                      Time: 12.56.51

                                                                                               1          2        3
Error Message/Concatenated Key                                                                  DBD     Segment  Level
-------------------------------------------------------------------------------------------  --------   -------- -----
Error when inserting segment - status code II                                                 DJ2E      SUBURB       1   4
FREMANTLE
CDCDCDEDC444444444      5
695415335000000000
Error when inserting segment - status code II                                                 DJ2E      NSTREET      2   4
FREMANTLE         FREMANTLE BRIDGE
CDCDCDEDC444444444CDCDCDEDC4CDCCCC44444444444444 5
695415335000000000695415335029947500000000000000
```

Figure 8. Database Load Diagnostic Report

1 The DBD name of each database that was loaded.

The primary database is listed first.

2 The level of the segment type in the database hierarchy.

3 The name of the segment.

4 The diagnostic message.

Diagnostic messages consist of a message code, a short descriptive text and the IMS status code.

5 The concatenated keys of the segment occurrences for which this diagnostic message was issued.

Example: Here is an example of a load of an IMS database in DLI mode using a static PSB.

```
//FMNILOAD 'Static PSB load'
//*
//FILEMAN  EXEC PGM=FMN1IMSB
//STEPLIB  DD DSN=FMN.SFMNMOD1,DISP=SHR
//SYSPRINT DD SYSOUT=*,RECFM=FBA,LRECL=133
//FMNTSPRT DD SYSOUT=*
//FMN1JIN  DD DISP=SHR,DSN=FMN.SFMNSLIB
//FMIMSIN  DD *
$$FILEM ILB REGNTYPE=DLI,
$$FILEM     PSBTYPE=STATIC,
$$FILEM     PSBDSN=IMSV910.FMI.PSBLIB,
$$FILEM     PSBMEM=PUPA,
$$FILEM     PCBNUM=(3,2,1),
$$FILEM     LOADTYPE=2,
$$FILEM     LDDSN=FMN.IMS.EXTRACT.DJ2E,
$$FILEM     DBDDSN=FMN.IMS.IVP.DBDLIB,
$$FILEM     DBDDSN2=IMSV910.FMI.DBDLIB,
$$FILEM     DBDMEM=DJ2E,
$$FILEM     IMSID=IF42
 /*
//LOAD     DD DSN=&&FMN1XTSD,DISP=SHR
```

```
//IEFRDER  DD DSN=FMN.IMSLOG.D060713.T120159.DJ2E,
//       UNIT=SYSALLDA,
//       SPACE=(TRK,(20,150),RLSE),
//       DCB=(RECFM=VB,LRECL=6140,BLKSIZE=6144),
//       DISP=(NEW,DELETE,CATLG)
//* -------------------------------------------------------------
//* -- Database data sets                                      --
//* -------------------------------------------------------------
//DJ2E DD DSN=FMN.IMS.IVP.DB.DJ2E,DISP=OLD
//DJ2F DD DSN=FMN.IMS.IVP.DB.DJ2F,DISP=OLD
//DJ3E DD DSN=FMN.IMS.IVP.DB.DJ3E,DISP=OLD
//DJ3F DD DSN=FMN.IMS.IVP.DB.DJ3F,DISP=OLD
//DJ1E DD DSN=FMN.IMS.IVP.DB.DJ1E,DISP=OLD
//DJ1F DD DSN=FMN.IMS.IVP.DB.DJ1F,DISP=OLD
//ENDRC     ENDIF
//*
```

Print (IPR):

Purpose

Prints either all the segments in an IMS database, or only those segments selected by the view you specify. You can choose to print the segments in CHAR, HEX, SNGL or TABL format (similar to the display formats in the FM/IMS editor under ISPF).

Usage notes

The Print JCL generated by FM/IMS has the following flows.

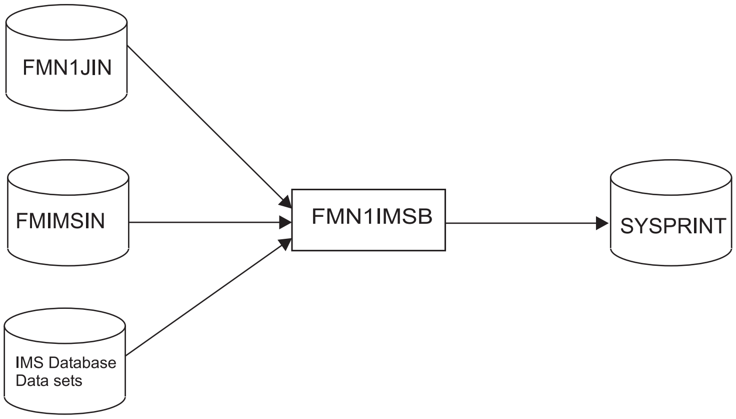


Figure 9. Flows for Print (IPR) step

FMN1JIN

Points to the FM/IMS skeleton library. This DD statement is only used when you are printing data from HALDB databases.

FMIMSIN

Contains the control statements for the Print. These control statements are discussed below.

IMS database data sets

In DLI mode, FM/IMS includes DDs for each of the IMS database data sets associated with the IMS database being printed. If the database has logical relationships, the data sets for the logically related databases need to be included.

SYSPRINT

Defines the output data set for:
- The messages that the Print issues.

- The printed IMS segments.

FMIMSIN control statements: You use the FMIMSIN control statements to specify:

- If the Print is to run in a BMP or DLI region.
- Whether to use a static PSB or if FM/IMS is to generate a dynamic PSB to print the IMS data.
- The format the IMS segments are printed in (CHAR, HEX, SNGL, or TABL).
- The FM/IMS view to be used when printing the IMS segments. Selection criteria (if specified) are used to determine the IMS segment to be printed. When printing in SNGL or TABL format, FM/IMS uses the fields to format the IMS segment.

Each line in FMIMSIN starts with a $$FILEM and the first line contains the parameter IPR to specify that FM/IMS is performing a Print.

```
$$FILEM IPR REGNTYPE=DLI,
$$FILEM     PSBTYPE=DYNAMIC,
$$FILEM     DBDDSN=FMN.IMS.IVP.DBDLIB,
. . .
```

The syntax of the control statements for a Print is shown here.

Syntax - part 1

```
                      (1)
►►──IPR──IMSID=imsid──────┬─REGNTYPE=BMP─┤ BMP parameters ├─┬──────────────────────────────►
                          └─REGNTYPE=DLI─┤ DLI parameters ├─┘


                                   (2)
   ┌─PSBTYPE=installation default──────────┐
►──┼───────────────────────────────────────┼──DBDDSN=dbddsn──┬──────────────────────┬─────►
   ├─PSBTYPE=DYNAMIC─┤ DYNAMIC parameters ├─┤                 └─DBDDSN2=dbddsn2──────┘
   └─PSBTYPE=STATIC─┤ STATIC parameters ├───┘

                 ┌─VIEW=N────────────────────────────────────────────┐   ┌─PRTFORM=CHAR─┐
►──DBDMEM=dbdname──┼───────────────────────────────────────────────────┼───┼──────────────┼─────►◄
                 │         ┌─VIEWDD=VIEWDD────┐                       │   ├─PRTFORM=HEX──┤
                 └─VIEW=Y──┼──────────────────┼──┬──────────────────┬─┘   ├─PRTFORM=SNGL─┤
                           ├─VIEWDD=ddname────┤  └─VIEWMEM=VIEWmem──┘     └─PRTFORM=TABL─┘
                           └─VIEWDSN=VIEWdsn──┘
```

BMP parameters:

```
                                               (3)                                   (3)
                   ┌─PARDLI=subsystem default──────┐  ┌─IMSNBA=subsystem default──────┐
├──┬────────────┬──┼───────────────────────────────┼──┼───────────────────────────────┼──────►
   └─AGN=agn────┘  ├─PARDLI=0──────────────────────┤  └─IMSNBA=number─────────────────┘
                   └─PARDLI=1──────────────────────┘

                                  (3)                                (3)
   ┌─IMSOBA=subsystem default──────────┐  ┌─LOCKMAX=subsystem default──────┐
►──┼───────────────────────────────────┼──┼────────────────────────────────┼───────────────────┤
   └─IMSOBA=number─────────────────────┘  └─LOCKMAX=number─────────────────┘
```

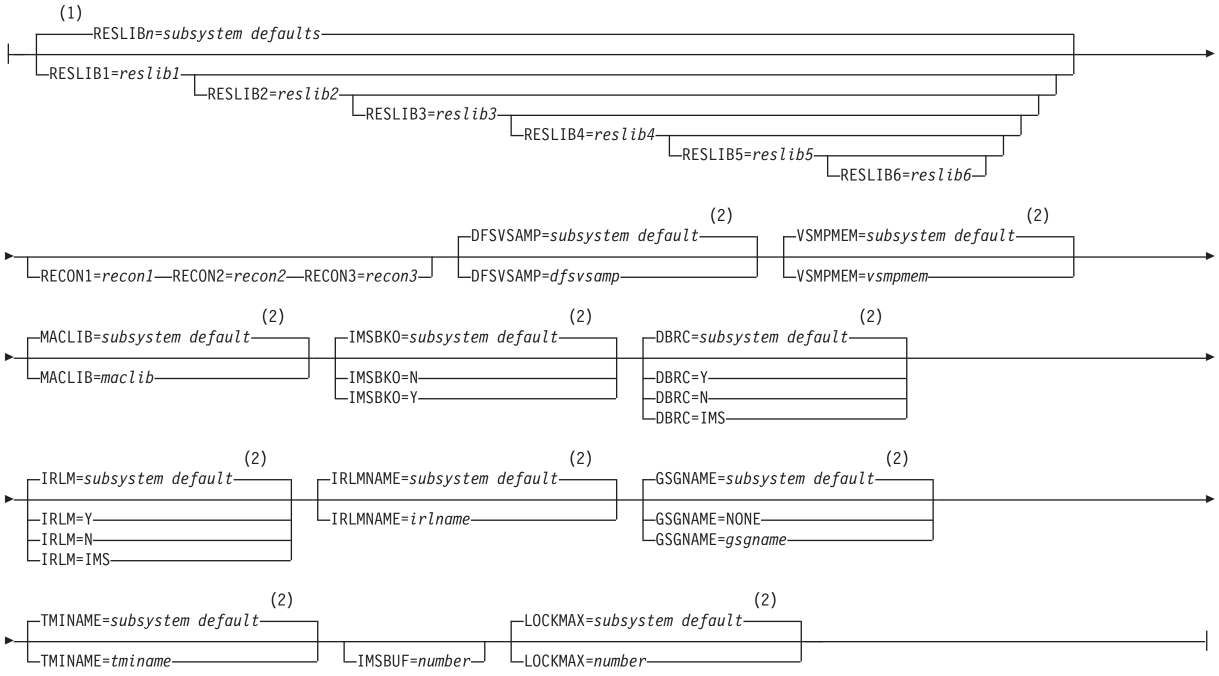
Notes:

1. If REGNTYPE=DLI is specified and the FM/IMS installation options module specifies COMPAT=Y, IMSID is an optional keyword. Otherwise, IMSID is a required keyword (as shown).
2. If you do not specify PSBTYPE, FM/IMS uses the PSBTYPE specified in the FM/IMS installation options module.
3. If you do not specify this parameter, FM/IMS uses the *subsystem* default for the parameter (if you have specified the IMSID parameter), or the *installation* default for the parameter (if you have not specified the IMSID parameter).

   The FM/IMS Administrator specifies the subsystem and installation defaults at installation time.

Syntax - part 2

DLI parameters:



Notes:

1 If you do not specify the RESLIB parameters, FM/IMS uses the *subsystem* defaults for the parameters (if you have specified the IMSID parameter), or the *installation* defaults for the parameters (if you have not specified the IMSID parameter).

2 If you do not specify this parameter, FM/IMS uses the *subsystem* default for the parameter (if you have specified the IMSID parameter), or the *installation* default for the parameter (if you have not specified the IMSID parameter).

The FM/IMS Administrator specifies the subsystem and installation defaults at installation time.

Syntax - part 3

DYNAMIC parameters:

```
                                         (1)
   ┌─PROCOPT=subsystem default───────┐
├──┼────────────────────────────────┼──────────────────────────────────────────────────┤
   ├─PROCOPT=G──────────────────────┤
   └─PROCOPT=GO─────────────────────┘
```

STATIC parameters:

```
├──PSBDSN=psbdsn──PSBMEM=psbmbr──┬─────────────────────────┬──────────────────────────┤
                                 │          ┌─,───────┐    │
                                 │          ▼         │    │
                                 ├─PCBNUM=(───pcbnum──┴─)──┤
                                 │           ┌─,────────┐  │
                                 │           ▼          │  │
                                 └─PCBNAME=(───pcbname──┴─)┘
```

Notes:

1 If you do not specify this parameter, FM/IMS uses the *subsystem* default for the parameter (if you have specified the IMSID parameter), or the *installation* default for the parameter (if you have not specified the IMSID parameter).

The FM/IMS Administrator specifies the subsystem and installation defaults at installation time.

The VIEW, VIEWDD, VIEWDSN, VIEWMEM and PRTFORM parameters are described here. All the other parameters in this syntax diagram can also be specified for the Extract (IXB) function. For a description of these common parameters, see “Extract (IXB)” on page 71 for details.

VIEW Optional parameter. Specifies whether or not the Print uses a view.

N The Print does not use a view.

Y The Print uses a view. If Y is specified, you must specify the view that is to be used as outlined in the VIEWDD, VIEWDSN and VIEWMEM entries:

VIEWDD=ddname
Identifies the DD statement that specifies the sequential data set or PDS(E) and member name that contains the view you want your Print to use. The default is VIEWDD.

VIEWDSN=viewdsn
The name of the data set that contains the view that you want your Print to use. You may specify a sequential data set or a PDS(E). If you specify a PDS(E), the parameter may also include a member name in parentheses.

VIEWMEM=viewmem
Required when the VIEWDSN or VIEWDD parameter specifies a PDS(E) and does not specify the member. Specifies the member of the PDS(E) that contains the view you want your Print to use.

PRTFORM

Required parameter. Specifies the format in which the data is to be printed:

CHAR

Prints the data in character format.

HEX Prints the data in hexadecimal ″updown″ format. This format uses three lines; one for the character, one for the zone digit, and one for the numeric digit of each byte.

SNGL Prints the data in SNGL format. This format splits the data into fields and uses a separate line for each field. Only available when you are using a view.

TABL Prints the data in a tabular format. Splits the data into fields and arranges the fields in columns. Only available when you are using a view.

Example: Here is an example of a printing an IMS database in TABL format in DLI mode using a dynamic PSB.

```
//FMNIPRNT 'Dynamic PSB print'
//*
//FILEMAN  EXEC PGM=FMN1IMSB
//STEPLIB  DD DSN=FMN.SFMNMOD1,DISP=SHR
//SYSPRINT DD SYSOUT=*,RECFM=FBA,LRECL=133
//FMNTSPRT DD SYSOUT=*
//FMN1JIN  DD DISP=SHR,DSN=FMN.SFMNSLIB
//FMIMSIN  DD *
$$FILEM     IMSID=IF52,
$$FILEM     VIEW=Y,
$$FILEM     VIEWDD=VIEWDD,
$$FILEM     VIEWMEM=DJ2EVJU,
$$FILEM     DBDDSN=FMN.IMS.IVP.DBDLIB,
$$FILEM     DBDDSN2=IMSV910.FMI.DBDLIB,
$$FILEM     DBDMEM=DJ2E,
$$FILEM     IMSID=IF52,
$$FILEM     VIEW=Y,
$$FILEM     VIEWDD=VIEWDD,
$$FILEM     VIEWMEM=DJ2EVJU,
$$FILEM     PRTFORM=TABL
 /*
//VIEWDD    DD DSN=FMN.IMS.IVP.VIEW,DISP=SHR
//* ---------------------------------------------------------------
//* -- Database data sets                                        --
//* ---------------------------------------------------------------
//DJ2E DD DSN=FMN.IMS.IVP.DB.DJ2E,DISP=SHR
//DJ2F DD DSN=FMN.IMS.IVP.DB.DJ2F,DISP=SHR
//DJ3E DD DSN=FMN.IMS.IVP.DB.DJ3E,DISP=SHR
//DJ3F DD DSN=FMN.IMS.IVP.DB.DJ3F,DISP=SHR
//DJ1E DD DSN=FMN.IMS.IVP.DB.DJ1E,DISP=SHR
//DJ1F DD DSN=FMN.IMS.IVP.DB.DJ1F,DISP=SHR
```

Batch Edit (IEB):

Purpose

Runs a REXX procedure that, in addition to containing any other valid REXX, can contain the following functions which are all described in separate sections later in this chapter.

DELIMS	Delete an IMS segment
GETIMS	Retrieve an IMS segment
ISRTIMS	Insert an IMS segment

PRINT
Print an IMS segment

REPLIMS
Update an IMS segment

SAVEIMS
Commits the data to the IMS database and frees IMS resource. After SAVEIMS is issued, the position in the database is lost

SETRC
Sets the job step return code

VIEWIMS
Create a view for later use, or load an existing view (that you can then use with the GETIMS function to retrieve a segment).

VIEWPOSN
Positions on an IMS segment so that the next GETIMS (with USING_VIEW) retrieves segments starting at this point.

Usage notes

The REXX functions are available only in REXX procedures that you use with IEB.

Some of these functions pass values to, or read from, the following REXX variables:

INREC, OUTREC
The GETIMS function retrieves a segment, and passes the contents of the segment to both INREC and OUTREC. Use INREC as a reference variable only, and do not update it. You can update OUTREC, and use it as the contents of a new segment that you insert with the ISRTIMS function, or you can use it with the REPLIMS function to replace the contents of an existing segment.

FMSTATUS
Contains the DL/I status code from the previous DELIMS, GETIMS, ISRTIMS or REPLIMS function call. For a description of DL/I status code values, see *IMS Messages and Codes*.

FMCONKEY, FMSEGNM
Contains the concatenated key (FMCONKEY) or name (FMSEGNM) of the segment deleted, retrieved, inserted or replaced by the previous DELIMS, GETIMS, ISRTIMS or REPLIMS function call.

FMNDBDNM
Contains the DBD name. (However, this variable is null until you call a DELIMS, GETIMS, ISRTIMS or REPLIMS function.)

When using the Rexx function EXECIO, ensure that you code the command ″ADDRESS MVS″ before the EXECIO, otherwise you get an RC-3 executing the REXX.

These REXX variable names must be specified in all uppercase.

For all IEB functions:
The string of keywords and keyword values for each IEB function statement MUST be enclosed in single quotes. For statements requiring more than one line, each line must be enclosed in single quotes with the end quote followed by a comma to indicate the continuation to the next line.

For IEB functions DELIMS, GETIMS, ISRTIMS and REPLIMS :
Where a field_value variable is specified for a *#field_name* or *#field_ref* keyword, the field_value variable may be enclosed in single or double quotes or may be specified without quotes.

For IEB function VIEWIMS:
Field *selection_criteria* are specified as REXX expressions. The *selection_criteria* expression should NOT be enclosed within single or double quotes.

The Batch Edit JCL generated by FM/IMS has the following flows:

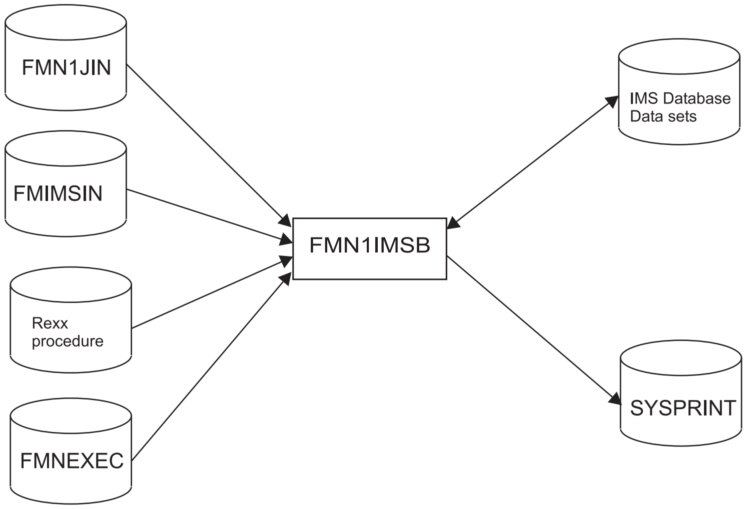


Figure 10. Flows for Batch Edit (IEB) step

FMN1JIN
Points to the FM/IMS skeleton library. This DD statement is only used when you are editing HALDB databases.

FMIMSIN
Contains the control statements for the batch edit. These control statements are discussed below.

IMS database data sets
In DLI mode, FM/IMS includes DDs for each of the IMS database data sets associated with the IMS database being maintained. If the database has logical relationships, the data sets for the logically related databases need to be included.

SYSPRINT
This contains the FM/IMS messages that were issued by FM/IMS while running the IEB REXX procedure.

REXX Procedure
The REXX procedure which is to be run to maintain the IMS database in batch. This REXX procedure consists of standard REXX statements combined which FM/IMS IEB calls to allow a user to read and update IMS segments.

In FMIMSIN, if PROC=* is specified in the REXX procedure, then the REXX procedure is specified instream after the PROC=* parameter.

In FMIMSIN, if PROC=<*member*> is specified, then REXX procedure <*member*> found in the FMNEXEC library is run.

FMNEXEC
This contains the member that is to be run if PROC=<*member*> is specified in the FMIMSIN parameters.

FMIMSIN control statements: You use the FMIMSIN control statements to specify:

- If the Batch Edit is to run in a BMP or DLI region.
- Whether to use a static PSB, or if FM/IMS is to generate a dynamic PSB to maintain the IMS data.
- Whether the IEB Rexx procedure to be run is instream (PROC=*), or can be found in the library specified on the FMNEXEC DD.

Each line in FMIMSIN starts with a $$FILEM and the first line contains the parameter IEB to specify that FM/IMS is performing Batch Edit.

```
$$FILEM IEB REGNTYPE=DLI,
$$FILEM     PSBTYPE=DYNAMIC,
$$FILEM     DBDDSN=FMN.IMS.IVP.DBDLIB,
. . .
```

The syntax of the control statements for a Batch Edit is shown here.

Syntax - part 1

```
                  (1)
►►──IEB──IMSID=imsid──────┬─REGNTYPE=BMP─┤ BMP parameters ├─┬──────────────────────────────────►
                          └─REGNTYPE=DLI─┤ DLI parameters ├─┘


                                        (2)
  ┌─PSBTYPE=installation default──────────┐
►─┼───────────────────────────────────────┼─DBDDSN=dbddsn──┬──────────────────┬─DBDMEM=dbdname──►
  ├─PSBTYPE=DYNAMIC───────────────────────┤                └─DBDDSN2=dbddsn2──┘
  └─PSBTYPE=STATIC─┤ STATIC parameters ├──┘

  ┌─VIEW=N───────────────────────────────────────────┐
►─┼──────────────────────────────────────────────────┼─┬─PROC=proc─────┬───────────────────────►
  │                                                  │ └─PROC=asterisk─┘
  └─VIEW=Y─┬─VIEWDD=VIEWDD──────┬─┬──────────────────┬┘
           ├─VIEWDD=ddname──────┤ └─VIEWMEM=VIEWmem──┘
           └─VIEWDSN=VIEWdsn────┘


                                     (3)
  ┌─CHKPFREQ=subsystem default────────┐
►─┼───────────────────────────────────┼───────────────────────────────────────────────────────►◄
  └─CHKPFREQ=chkpfreq─────────────────┘
```

BMP parameters:

```
                                            (3)                              (3)
              ┌─PARDLI=subsystem default────────┐ ┌─IMSNBA=subsystem default────────┐
├──┬──────────┬┼────────────────────────────────┼─┼─────────────────────────────────┼──────────►
   └─AGN=agn──┘├─PARDLI=0───────────────────────┤ └─IMSNBA=number───────────────────┘
               └─PARDLI=1───────────────────────┘

                                   (3)                              (3)
  ┌─IMSOBA=subsystem default──────────┐ ┌─LOCKMAX=subsystem default───┐
►─┼───────────────────────────────────┼─┼─────────────────────────────┼──────────────────────────┤
  └─IMSOBA=number─────────────────────┘ └─LOCKMAX=number──────────────┘
```

Notes:

1 If REGNTYPE=DLI is specified and the FM/IMS installation options module specifies COMPAT=Y, IMSID is an optional keyword. Otherwise, IMSID is a required keyword (as shown).

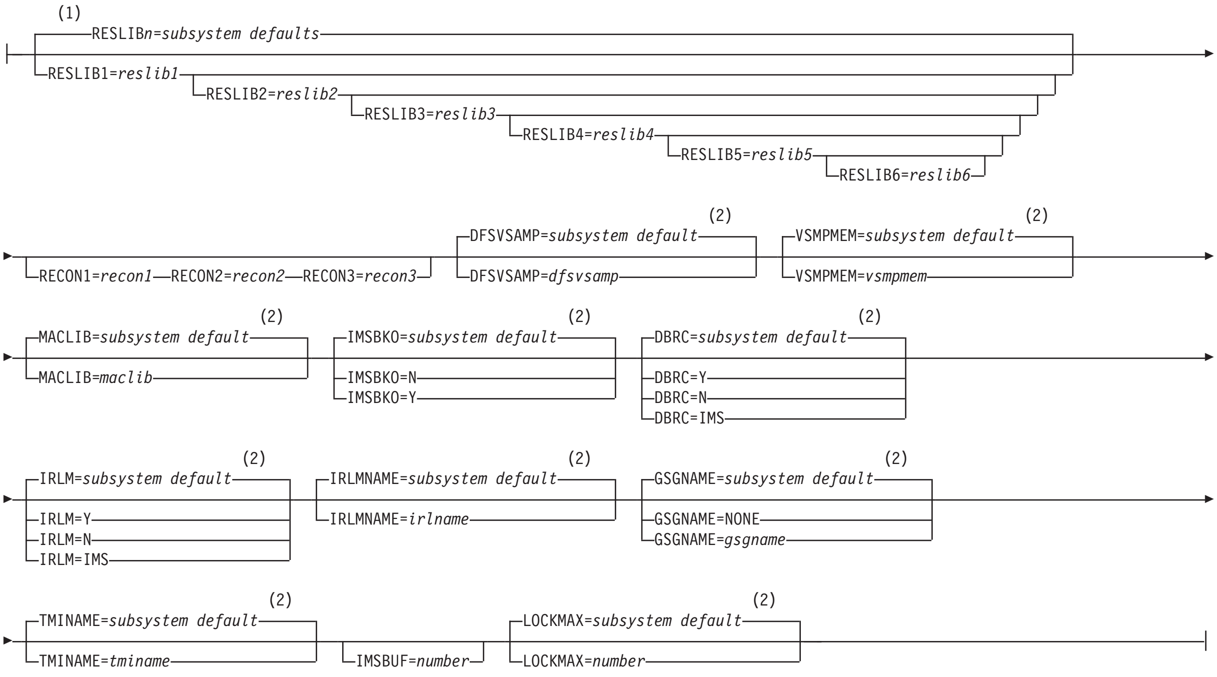
2 If you do not specify PSBTYPE, FM/IMS uses the PSBTYPE specified in the FM/IMS installation options module.

3 If you do not specify this parameter, FM/IMS uses the *subsystem* default for the parameter (if you have specified the IMSID parameter), or the *installation* default for the parameter (if you have not specified the IMSID parameter).

The FM/IMS Administrator specifies the subsystem and installation defaults at installation time.

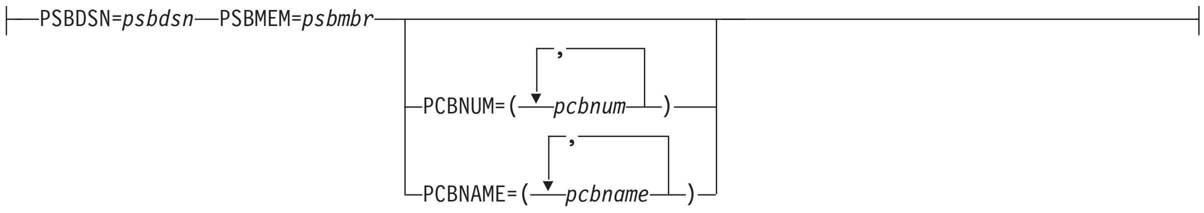
Syntax - part 2

DLI parameters:



```
(1)
RESLIBn=subsystem defaults
RESLIB1=reslib1
RESLIB2=reslib2
RESLIB3=reslib3
RESLIB4=reslib4
RESLIB5=reslib5
RESLIB6=reslib6

RECON1=recon1 RECON2=recon2 RECON3=recon3
DFSVSAMP=subsystem default (2)
DFSVSAMP=dfsvsamp
VSMPMEM=subsystem default (2)
VSMPMEM=vsmpmem

MACLIB=subsystem default (2)
MACLIB=maclib
IMSBKO=subsystem default (2)
IMSBKO=N
IMSBKO=Y
DBRC=subsystem default (2)
DBRC=Y
DBRC=N
DBRC=IMS

IRLM=subsystem default (2)
IRLM=Y
IRLM=N
IRLM=IMS
IRLMNAME=subsystem default (2)
IRLMNAME=irlname
GSGNAME=subsystem default (2)
GSGNAME=NONE
GSGNAME=gsgname

TMINAME=subsystem default (2)
TMINAME=tminame
IMSBUF=number
LOCKMAX=subsystem default (2)
LOCKMAX=number
```

STATIC parameters:



```
PSBDSN=psbdsn PSBMEM=psbmbr
PCBNUM=( pcbnum )   (repeat separator: ,)
PCBNAME=( pcbname )   (repeat separator: ,)
```

Notes:

1 If you do not specify the RESLIB parameters, FM/IMS uses the *subsystem* defaults for the parameters (if you have specified the IMSID parameter), or the *installation* defaults for the parameters (if you have not specified the IMSID parameter).

2 If you do not specify this parameter, FM/IMS uses the *subsystem* default for the parameter (if you have specified the IMSID parameter), or the *installation* default for the parameter (if you have not specified the IMSID parameter).

The FM/IMS Administrator specifies the subsystem and installation defaults at installation time.

The VIEW, VIEWDD, VIEWDSN, VIEWMEM, CHKPFREQ and PROC parameters are described here. All the other parameters in this syntax diagram can also be specified for the Extract (IXB) function. For a description of these common parameters, see "Extract (IXB)" on page 71.

CHKPFREQ=*chkpfreq*

Optional parameter. Specifies the frequency of the automatic save function during the Batch Edit; that is, the Batch Edit checkpoint frequency.
FM/IMS:

- Increments a count by 1 each time the Batch Edit inserts, deletes, or replaces a segment.
- Issues a checkpoint when the count is equal to the Batch Edit checkpoint frequency.

The valid range is 1 to 99999.

VIEW Optional parameter. Specifies whether or not the Batch Edit uses a view.

N The Batch Edit does not use a view.

Y The Batch Edit uses a view. If Y is specified, you must specify the view that is to be used as outlined in the VIEWDD, VIEWDSN, and VIEWMEM entries.

VIEWDD=*ddname*

Identifies the DD statement that specifies the sequential data set or PDS(E) and member name that contains the view you want the Batch Edit to use. The default is VIEWDD.

VIEWDSN=*viewdsn*

The name of the data set that contains the view that you want your Batch Edit to use. You may specify a sequential data set or a PDS(E). If you specify a PDS(E), the parameter may also include a member name in parentheses.

VIEWMEM=*viewmem*

Required when the VIEWDSN or VIEWDD parameter specifies a PDS(E) and does not specify the member. Specifies the member of the PDS(E) that contains the view you want the Batch Edit to use.

PROC=*proc*

Required parameter. Valid values are:

The name of a PDS(E) member

The name of the PDS(E) member containing the REXX procedure that you want the Batch Edit to use.

You specify the PDS(E) containing this member in the FMNEXEC DD statement. The JCL for the Batch Edit must include this DD statement.

An asterisk (*)

Indicates that the REXX procedure is inline.

The procedure is read from SYSIN immediately after the FMIMSIN control statements. The inline procedure is terminated by a record containing a slash and a plus sign (/+) in columns 1–2.

REXX IEB calls: The REXX procedure that FM/IMS invokes uses standard TSO/E REXX. However, you are able to call IEB functions to be able to retrieve, update, and print IMS segments. You are also able to create an FM/IMS view for use in the IEB procedure.

In the REXX procedure, you use GETIMS calls to retrieve segments from the IMS database.

You have the option of using:

- GETIMS calls with the USING _VIEW clause (Retrieval using a view).

  FM/IMS uses the selection criteria specified in the view to retrieve the relevant segment.
- GETIMS calls without the USING_VIEW clause (Retrieval without a view).

  You have to navigate the database by issuing GETIMS call to retrieve the relevant segments. This method is similar to issuing IMS calls in a program; you need to determine how the segments are to be retrieved.

But FM/IMS does not allow you to use GETIMS calls with the USING_VIEW clause and GETIMS calls without the USING_VIEW clause in the same REXX procedure.

Retrieval using a view: The following IEB calls are available when processing the database using a view.

DELIMS calls (Delete an IMS segment): This IEB call deletes the current segment and must be preceded by a GETIMS function call with the USING_VIEW parameter.

Syntax

```
►►──DELIMS──(─SEGMENT=delsegm─)──────────────────────────────────────►◄
```

delsegm
The name of the segment you want to delete.

If FM/IMS encounters an unavailable HALDB partition or DEDB area, the DELIMS function receives a return code of 8 and the ISPF variable FMSTATUS indicates an IMS status code of:

FH Unavailable DEDB area
BA Unavailable HALDB partition

Returns zero if successful, otherwise nonzero. If DELIMS is unsuccessful, then you can check the DL/I status code in the FMSTATUS REXX variable to determine the cause of failure.

IEB example using DELIMS:

```
⋮
$$FILEM IEB
⋮
$$FILEM PROC=*
filerc = VIEWIMS('TPLDSN=FMN.IMS.IVP.TEMPLATE')
 /* Delete Segment at the current position */
filerc = GETIMS('SEGMENT=NSTREET NEXT')
filerc = DELIMS('SEGMENT=NSTREET')...
```

GETIMS CALL (Retrieve an IMS segment): This IEB call retrieves the next segment selected by the view, skipping any intervening segments that are not selected by the view.

Syntax (using a view)

```
          ┌─NEXT──────────────────────────┐
►►──GETIMS──(─┼───────────────────────────────┼─USING_VIEW)──────────────►◄
          │                ┌─NEXT─┐       │
          ├─SEGMENT=getsegm─┴──────┴───────┤
          ├─TOP────────────────────────────┤
          └─ROOT───────────────────────────┘
```

getsegm
: The name of the segment you want to retrieve. If you also specify CHILD, then GETIMS retrieves the next hierarchically-dependent segment with this name. Otherwise, the default value is NEXT, and GETIMS retrieves the next segment with this name, regardless of hierarchy.

NEXT Retrieves the next segment in the database that matches the criteria specified in the view.

ROOT
: Retrieves the root segment in the hierarchical path of the current segment. If you use specify ROOT with USING_VIEW, and the root segment is not selected by the view, then ROOT retrieves the first selected segment after that root segment.

TOP Without USING_VIEW, TOP retrieves the first root segment in the database. With USING_VIEW, TOP retrieves the first segment in the database that is selected by the view; this is not necessarily a root segment.

USING_VIEW
: Restricts GETIMS to retrieving segments that are selected by the view, which you must have already loaded by calling the VIEWIMS function before this call to GETIMS.

Returns zero if successful, otherwise nonzero. A return code of 4 indicates that the GETIMS('NEXT USING_VIEW') reached the end of the database.

If FM/IMS encounters an unavailable HALDB partition or DEDB area, the GETIMS function receives a return code of 8 and the ISPF variable FMSTATUS indicates an IMS status code of:

FH Unavailable DEDB area
BA Unavailable HALDB partition

For nonzero return codes other than 4, check the DL/I status code in the FMSTATUS REXX variable to determine the cause of failure.

If successful, GETIMS passes the contents of the retrieved segment to the INREC and OUTREC REXX variables.

You can use GETIMS either with or without a view. If you use a view, then GETIMS only retrieves segments selected by the view. For example, GETIMS('NEXT USING_VIEW') retrieves the next segment selected by the view, skipping any intervening segments that are not selected by the view.

IEB example using GETIMS USING_VIEW:

```
⋮
$$FILEM IEB
⋮
$$FILEM PROC=*
filerc = VIEWIMS('TPLDSN=FMN.IMS.IVP.TEMPLATE')
 /* Retrieve the next IMS segment from the view */
filerc = GETIMS('NEXT USING_VIEW')
```

ISRTIMS CALL (Insert an IMS segment): This IEB call inserts an IMS segment.

Syntax

```
►►──ISRTIMS──(─SEGMENT=inssegm────────────────────────────────────────►

►─┬──────────────────────────────────────────┬────────────────────────►◄
  │       ┌──────────────────────────────┐   │
  │       ▼                         (1)  │   │
  └─SET──┬─#field_ref=field_value──────┬─┴─)─┘
         └─#field_name=field_value─────┘
```

Notes:

1 To use field reference numbers, you must have already loaded a view, by preceding this function call with a call to VIEWIMS.

inssegm
: The name of the segment you want to insert. This must be a child of the current segment.

field_ref
: A field reference number.

field_name
: A field name.

field_value
: The value of the field specified by *field_ref* or *field_name*.

Returns zero if successful, otherwise nonzero.

If FM/IMS encounters an unavailable HALDB partition or DEDB area, the DELIMS function receives a return code of 8 and the ISPF variable FMSTATUS indicates an IMS status code of:

FH Unavailable DEDB area
BA Unavailable HALDB partition

To use OUTREC as the contents of the new segment, omit the SET clause. If you are inserting a variable-length segment, then the first two bytes of OUTREC must specify the length of the segment. Otherwise, use the SET clause to specify the field values for the new segment.

IEB example using ISRTIMS:

```
⋮
$$FILEM IEB
⋮
$$FILEM PROC=*
filerc = VIEWIMS('TPLDSN=FMN.IMS.IVP.TEMPLATE')
filerc = GETIMS('SEGMENT=SUBURB NEXT USING_VIEW')
```

```
 /* Insert using OUTREC */
OUTREC = "FLOREAT RESERVE"
filerc = ISRTIMS('SEGMENT=NSTREET')

 /* Insert using SET */

filerc = ISRTIMS('SEGMENT=NSTREET ',
                 ' SET #2=FLOREAT PARK ')
```

PRINT CALL (Print an IMS segment): This function prints the current segment in the format specified.

Syntax

```
                      ┌CHAR─┐
►►──PRINT──(─INREC,─┤      ├─)─────────────────────────────────────────►◄
                      ├HEX──┤
                      ├SNGL─┤
                      └TABL─┘
```

Returns zero if successful, otherwise nonzero.

Prints the segment contents stored in the INREC variable, in one of the following formats:

CHAR Prints the segment as it is stored on the IMS database.

HEX Prints three lines of data. The first line shows the character representation of each byte. The second and third lines show the hexadecimal value of each byte.

SNGL Prints the segment split into individual fields down the report.

TABL Prints the segment split into individual fields across the report.

The INREC variable contains the data of the current segment.

To print in SNGL or TABL format, you must have already loaded a view, by preceding this function call with a call to VIEWIMS.

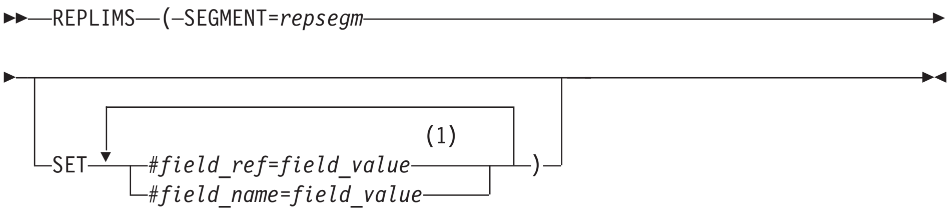
IEB example using PRINT

```
⋮
$$FILEM IEB
⋮
$$FILEM     PROC=*

filerc = VIEWIMS('TPLDSN=FMN.IMS.IVP.TEMPLATE')

filerc = GETIMS('SEGMENT=SUBURB  ',
               ' WHERE SEGMENT=SUBURB #SUBURB-NAME=CITY BEACH')
PRINT(INREC,CHAR)
PRINT(INREC,HEX)
PRINT(INREC,SNGL)
PRINT(INREC,TABL)
```

```
/+
⋮
```

REPLIMS CALL (Replace an IMS segment): This IEB call replaces the current segment and must be preceded by a GETIMS function call with the USING_VIEW parameter.

Syntax



Notes:

1 To use field reference numbers, you must have already loaded a view, by preceding this function call with a call to VIEWIMS.

repsegm
The name of the segment you want to replace.

field_ref
A field reference number.

field_name
A field name.

field_value
The value of the field specified by *field_ref* or *field_name*.

Returns zero if successful, otherwise nonzero.

If FM/IMS encounters an unavailable HALDB partition or DEDB area, the DELIMS function receives a return code of 8 and the ISPF variable FMSTATUS indicates an IMS status code of:

FH Unavailable DEDB area
BA Unavailable HALDB partition

To replace the current contents of the segment with the contents of the OUTREC variable, omit the SET clause. If you are replacing a variable- length segment, then the first two bytes of OUTREC must specify the length of the segment. Otherwise, use the SET clause to specify replacement field values for the segment.

When you use OUTREC, you replace the entire contents of the segment. When you use the SET clause, you can replace only the field values that you specify; other field values remain unchanged.

IEB example using REPLIMS:

```
⋮
$$FILEM IEB
⋮
$$FILEM PROC=*
filerc = VIEWIMS('TPLDSN=FMN.IMS.IVP.TEMPLATE')
filerc = GETIMS('SEGMENT=SUBURB NEXT USING_VIEW')

 /* Replace a segment using OUTREC */
OUTREC = OVERLAY('6168',outrec,19)
filerc = REPLIMS('SEGMENT=SUBURB')

 /* Replace a segment using SET */
rtrc=REPLIMS('SEGMENT=SUBURB ',
             'SET #POST-CODE=6168')
```

SETRC CALL (Set Job Step return code): The IEB call sets the Job Step return code to the value specified. It does not cause FM/IMS to exit the REXX procedure. The format of the call is:

Syntax

```
►►──SETRC─(rc)────────────────────────────────────────────────────────►◄
```

rc Return code.

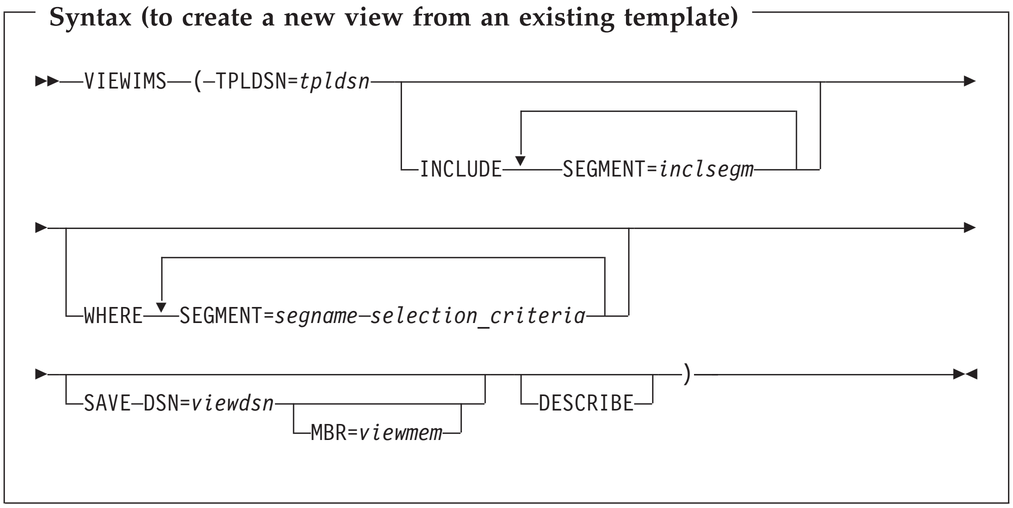
IEB example using SETRC:

```
RTRC=GETIMS(USING_VIEW)
IF RTRC<>0 THEN DO
   SETRC(21)
   EXIT
END
```

VIEWIMS CALL (Load or create a view):

Syntax (to load an existing view or template)

```
►►──VIEWIMS──(──┬──┬─VIEWDSN=viewdsn─┬──┬───────────────────┬─┬──────────►
                │  └─VIEWDD=viewdd───┘  └─VIEWMBR=viewmem───┘ │
                └──┬─TPLDSN=tpldsn─┬──────────────────────────┘
                   └─TPLDD=tpldd───┘

►──┬──────────┬──)──────────────────────────────────────────────────────►◄
   └─DESCRIBE─┘
```



Syntax (to create a new view from an existing template)

If you specify the DESCRIBE parameter, then VIEWIMS prints a report showing the structure of the view or template for each segment in the database.

viewdsn
One of:
- The name of the sequential data set containing the view.
- The name of the PDS containing the view. (You then specify the member name in *viewmem*.)

viewdd The DDNAME containing one of the following:
- The sequential data set containing the view.
- The PDS containing the view. You then specify the member name in *viewmem*.

viewmem
The member name of the view.

tpldsn The name of the data set containing the template that is used as a basis to create a view. The template for the DBD must have already been created on-line in FM/IMS.

tpldd The DDNAME containing the template data set that is used as a basis to create a view. The template for the DBD must have already been created online in FM/IMS.

inclsegm
The name of a segment to be included in the view.

If you call the GETIMS function with the USING_VIEW parameter, then only the included in the view is retrieved.

However, if you omit the INCLUDE clause entirely, then all segments are included in the view.

segname
The name of a segment for which you want to specify field selection criteria.

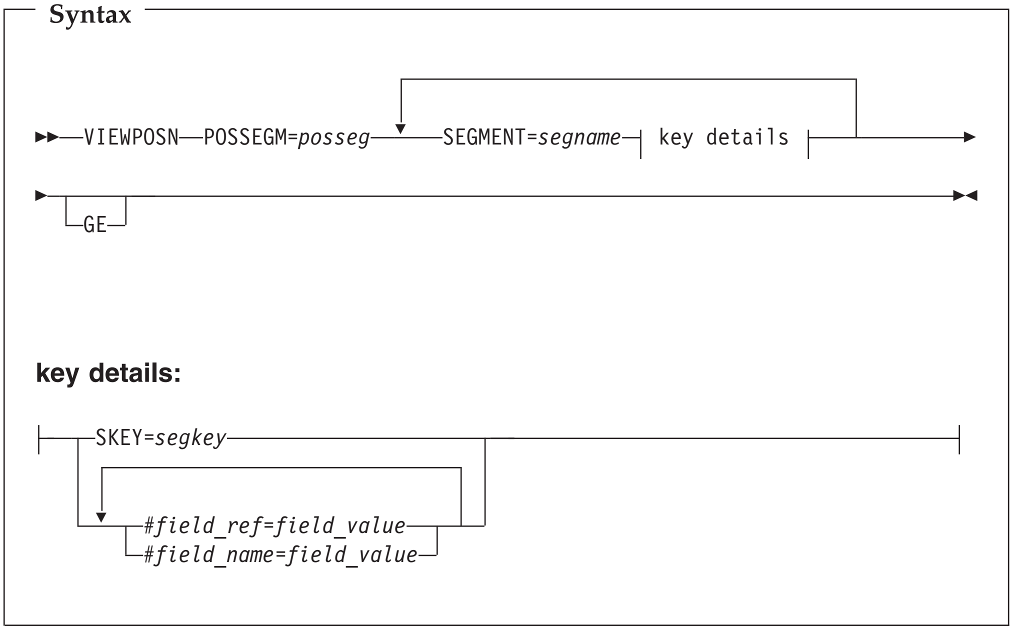
selection_criteria
Field selection criteria for the *segname* segment. Determines which segment occurrences are retrieved by calls to the GETIMS function with the USING_VIEW parameter.

For details on specifying field selection criteria, see page 86.

IEB example using VIEWIMS

```
⋮
$$FILEM IEB
⋮
$$FILEM     PROC=*
  rtrc=VIEWIMS('TPLDSN=FMN.IMS.IVP.TEMPLATE ',
               'SAVE     DSN=FMN.IMS.IVP.VIEW MBR=ZZ999 ',
               'INCLUDE  SEGMENT=SUBURB ',
               '         SEGMENT=LINKSTR ',
               'WHERE    SEGMENT=SUBURB substr(#2,1,1)=F ',
               '         SEGMENT=LINKSTR #4=ST ',
               'DESCRIBE')
/+
⋮
```

VIEWPOSN CALL (Position on an IMS segment): This IEB call positions on an IMS segment so that the next GETIMS (with USING_VIEW) retrieves segments starting at this point.



```
Syntax

                                      ┌────────────────────────────────────┐
                                      ▼                                    │
►►──VIEWPOSN──POSSEGM=posseg──────SEGMENT=segname─┤ key details ├──────────┴──────►

►──┬────┬──────────────────────────────────────────────────────────────────────►◄
   └─GE─┘

key details:

├──┬─SKEY=segkey─────────────────────┬──────────────────────────────────────────┤
   │  ┌───────────────────────────┐  │
   │  ▼                           │  │
   └────┬─#field_ref=field_value──┬─┴──┘
        └─#field_name=field_value─┘
```

posseg The name of the segment you want to position on.

segname
The name of a segment in the hierarchical path of the segment you want to position on.

field_ref
A field reference number.

field_name
A field name.

field_value
The value of the field specified by *field_ref* or *field_name*.

Returns zero if successful, otherwise nonzero.

If FM/IMS encounters an unavailable HALDB partition or DEDB area, the DELIMS function receives a return code of 8 and the ISPF variable FMSTATUS indicates an IMS status code of:

FH Unavailable DEDB area
BA Unavailable HALDB partition

To replace the current contents of the segment with the contents of the OUTREC variable, omit the SET clause. If you are inserting a variable- length segment, then the first two bytes of OUTREC must specify the length of the segment.

IEB example using VIEWPOSN:

```
⋮
$$FILEM IEB
⋮
$$FILEM PROC=*
filerc = VIEWIMS('TPLDSN=FMN.IMS.IVP.TEMPLATE')
filerc = VIEWPOSN('POSSEGM=NSTREET ',
                  'SEGMENT=SUBURB #SUBURB-NAME=CITY BEACH ',
                  'SEGMENT=NSTREET #2=FLOREAT BEACH ')

filerc = GETIMS(NEXT USING_VIEW)
```

Retrieval without a view: The following calls are available when processing the database without using a view.

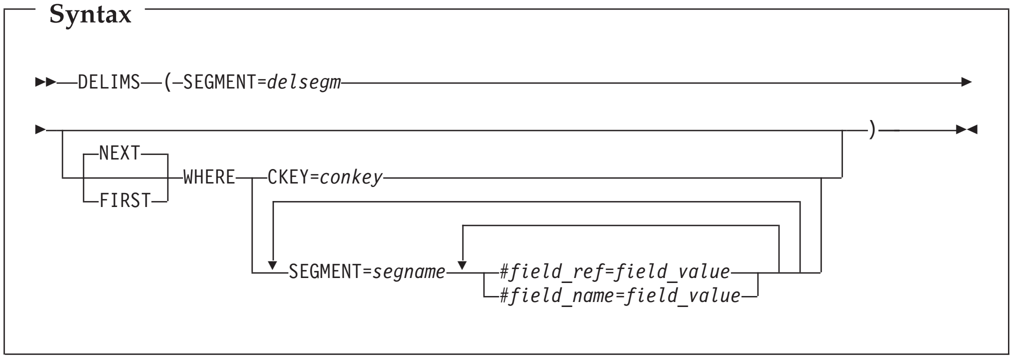
A view may be loaded to allow the WHERE clause to specify key fields or to print the database in SNGL or TABL format, but the GETIMS call is issued without the USING_VIEW parameter.

DELIMS CALL (Delete an IMS segment): This IEB call deletes the current segment.

Purpose
To delete an IMS segment

Usage notes
The DELIMS inbuilt FM/IMS REXX function can be used to:

- delete the current segment.
- delete a particular segment by specifying a where clause.



delsegm
The name of the segment you want to delete.

NEXT Deletes the next segment in the database after the current segment that satisfies the WHERE clause.

FIRST Deletes the first segment in the database that satisfies the WHERE clause.

conkey
The concatenated key of the segment you want to delete.

If you specify the concatenated key of a segment that you have already deleted, then DELIMS produces a nonzero return code, and does not delete any segment.

segname
The name of a segment in the hierarchical path of the segment that you want to delete.

field_ref
The reference number of a key field in the *segname* segment, as specified in the template or view.

field_name
The name of a key field in the *segname* segment, as specified in the template or view.

field_value
The value of the key field specified by *field_ref* or *field_name*.

If FM/IMS encounters an unavailable HALDB partition or DEDB area, the DELIMS function receives a return code of 8 and the ISPF variable FMSTATUS indicates an IMS status code of:

FH for an unavailable DEDB area.

BA an unavailable HALDB partition.

Returns zero if successful, otherwise nonzero. If DELIMS is unsuccessful, then you can check the DL/I status code in the FMSTATUS REXX variable to determine the cause of failure.

To delete the current segment, omit the WHERE clause. Otherwise, use the WHERE clause to identify the segment you want to delete.

IEB example using DELIMS

```
⋮
$$FILEM IEB
⋮
$$FILEM     PROC=*
filerc = VIEWIMS('TPLDSN=FMN.IMS.IVP.TEMPLATE')
 /*  Delete Segment at the current position    */
filerc = GETIMS('SEGMENT=NSTREET NEXT')
filerc = DELIMS('SEGMENT=NSTREET')
 /*  Delete Segment using a where Clause       */
filerc = DELIMS('SEGMENT=NSTREET ',
               ' WHERE SEGMENT=SUBURB #SUBURB-NAME=CITY BEACH',
                      'SEGMENT=NSTREET #2=FLOREAT BEACH ')
/+
⋮
```

GETIMS CALL (Retrieve an IMS segment):

Syntax (without a view)

```
►►──GETIMS──(──┬──────┬──────────────────────────────────────────────────►
               └─HOLD─┘


   ┌─NEXT──────────────────────────────────────────────┐
►──┼───────────────────────────────────────────────────┼──)──────────────►◄
   │                       ┌─NEXT─┐                    │
   ├─SEGMENT=getsegm───────┼──────┴─────────────────────┤
   │                       │ ┌─NEXT──┐                  │
   │                       └─┼───────┼─┤ WHERE clause ├─┘
   │                         └─FIRST─┘                  │
   ├─┬─────────────────┬─CHILD──────────────────────────┤
   │ └─SEGMENT=getsegm─┘                                │
   ├─PARENT─────────────────────────────────────────────┤
   ├─ROOT───────────────────────────────────────────────┤
   └─TOP────────────────────────────────────────────────┘
```

WHERE clause:

```
├──WHERE──┬─CKEY=conkey──────────────────────────────────────────┬──────┤
          │ ┌──────────────────────────────────────────────────┐ │
          │ │                ┌─────────────────────────────┐   │ │
          │ ▼                ▼               (1)           │   │ │
          └─SEGMENT=segname──┬─#field_ref──────=field_value─┬┴───┴─┘
                             └─#field_name=field_value──────┘
```

Notes:

1 To use field reference numbers, you must have already loaded a view, by preceding this function call with a call to VIEWIMS.

HOLD
Retrieves the segment with a hold. Code this parameter when the GETIMS call is followed by a REPLIMS call without a WHERE clause, or a DELIMS call without a WHERE clause.

If the segment or one of its parents has a nonunique key or no key, this parameter is mandatory. Without it, the REPLIMS call or DELIMS call fails. The REPLIMS call or DELIMS call will not fail if the segment and its parents have unique keys but, if you do code it, your EXEC executes less DL/I calls and runs more efficiently.

getsegm
The name of the segment you want to retrieve. If you also specify CHILD, then GETIMS retrieves the next hierarchically dependent segment with this name. Otherwise, the default value is NEXT, and GETIMS retrieves the next segment with this name, regardless of hierarchy.

CHILD
Retrieves the next segment in the hierarchical path of the current segment. If you also specify SEGMENT=*getsegm*, then GETIMS retrieves the next segment of that name in the hierarchical path of the current segment.

NEXT Retrieves the next segment in the database after the current segment.

FIRST Retrieves the first segment in the database that satisfies the WHERE clause.

ROOT
Retrieves the root segment in the hierarchical path of the current segment.

PARENT
Retrieves the parent segment of the current segment.

If the current segment is a root segment, then GETIMS retrieves that same segment. To avoid looping in this situation, you can use the FMSEGNM REXX variable to check the name of the current segment.

TOP Retrieves the first root segment in the database.

conkey
The concatenated key of the segment you want to retrieve.

segname
The name of a segment in the hierarchical path of the segment that you want to retrieve.

field_ref
The reference number of a key field in the *segname* segment, as specified in the template or view.

field_name
The name of a key field in the *segname* segment, as specified in the template or view.

field_value
The value of the key field specified by *field_ref* or *field_name*.

Returns zero if successful, otherwise nonzero. A return code of 4 indicates that a segment was not found for one of the following reasons:

- The segment specified by the WHERE clause does not exist.
- GETIMS('NEXT') reached the end of the database.
- GETIMS('CHILD') reached the end of the hierarchical path (current segment has no children).

If FM/IMS encounters an unavailable HALDB partition or DEDB area, the GETIMS function receives a return code of 8 and the ISPF variable FMSTATUS will indicate an IMS status code of:

FH an unavailable DEDB area

BA an unavailable HALDB partition

For nonzero return codes other than 4, check the DL/I status code in the FMSTATUS REXX variable to determine the cause of failure.

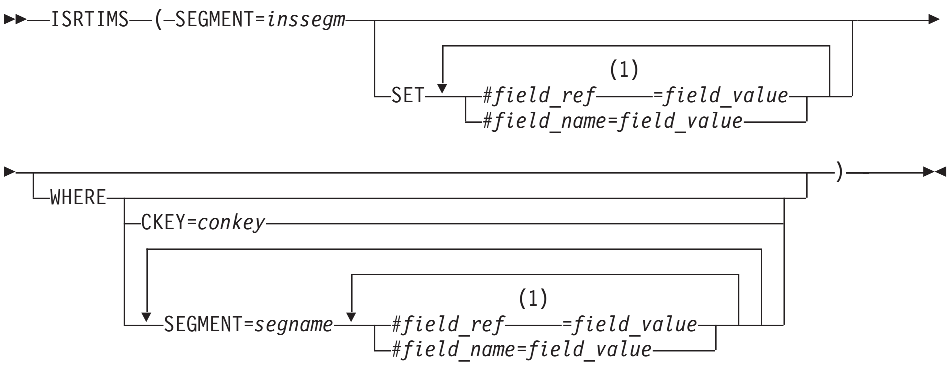
If successful, GETIMS passes the contents of the retrieved segment to the INREC and OUTREC REXX variables.

IEB example using GETIMS

```
⋮
$$FILEM IEB
⋮
$$FILEM     PROC=*

filerc = VIEWIMS('TPLDSN=FMN.IMS.IVP.TEMPLATE')
filerc = GETIMS('NEXT')
filerc = GETIMS()

filerc = GETIMS('SEGMENT=NSTREET ',
               ' WHERE SEGMENT=SUBURB #SUBURB-NAME=CITY BEACH ',
                      'SEGMENT=NSTREET #2=FLOREAT BEACH ')

filerc = GETIMS('SEGMENT=NSTREET ',
               ' WHERE CKEY=CITY BEACH        FLOREAT BEACH ')

/+
⋮
```

ISRTIMS CALL (Insert an IMS segment):

Syntax



Notes:

1 To use field reference numbers, you must have already loaded a view, by preceding this function call with a call to VIEWIMS.

inssegm
The name of the segment you want to insert. This must be a child of the segment that you specify in the WHERE clause, or, if you omit the WHERE clause, a child of the current segment.

conkey
The concatenated key of the parent of the segment you want to insert.

segname
The name of a segment in the hierarchical path of the parent of the segment that you want to insert.

field_ref
A field reference number.

field_name
A field name.

field_value
The value of the field specified by *field_ref* or *field_name*.

Returns zero if successful, otherwise nonzero.

If FM/IMS encounters an unavailable HALDB partition or DEDB area, the ISRTIMS function receives a return code of 8 and the ISPF variable FMSTATUS indicates an IMS status code of:

FH an unavailable DEDB area

BA an unavailable HALDB partition

To insert a new child segment under the current segment, omit the WHERE clause. Otherwise, use the WHERE clause to identify the parent of the new child segment that you want to insert.

To use OUTREC as the contents of the new segment, omit the SET clause. If you are inserting a variable-length segment, then the first two bytes of OUTREC must specify the length of the segment.

Otherwise, use the SET clause to specify the field values for the new segment.

IEB example using ISRTIMS

```
⋮
$$FILEM IEB
⋮
$$FILEM     PROC=*

filerc = VIEWIMS('TPLDSN=FMN.IMS.IVP.TEMPLATE')

filerc = GETIMS('NEXT')
filerc = GETIMS('SEGMENT=SUBURB  ',
              ' WHERE SEGMENT=SUBURB #SUBURB-NAME=CITY BEACH')

 /* Insert using OUTREC */

OUTREC = "FLOREAT RESERVE"
filerc = ISRTIMS('SEGMENT=NSTREET')

 /* Insert using SET    */

filerc = ISRTIMS('SEGMENT=NSTREET ',
              ' SET #2=FLOREAT RESERVE ',
              ' WHERE SEGMENT=SUBURB #SUBURB-NAME=CITY BEACH')

/+
⋮
```

PRINT CALL (Print an IMS segment):

Syntax

```
                        ┌─CHAR─┐
►►──PRINT──(─INREC,─────┼──────┼─)────────────────────────────────────►◄
                        ├─HEX──┤
                        ├─SNGL─┤
                        └─TABL─┘
```

Returns zero if successful, otherwise nonzero.

Prints the segment contents stored in the INREC variable, in one of the following formats:

CHAR Prints the segment as it is stored on the IMS database.

HEX Prints three lines of data. The first line shows the character representation of each byte. The second and third lines show the hexadecimal value of each byte.

SNGL Prints the segment split into individual fields down the report.

TABL Prints the segment split into individual fields across the report.

The INREC variable contains the data of the current segment.

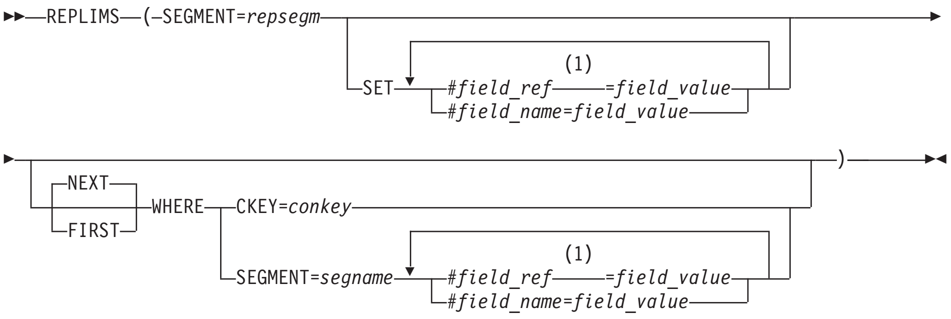
To print in SNGL or TABL format, you must have already loaded a view, by preceding this function call with a call to VIEWIMS.

IEB example using PRINT

```
⋮
$$FILEM IEB
⋮
$$FILEM     PROC=*

filerc = VIEWIMS('TPLDSN=FMN.IMS.IVP.TEMPLATE')

filerc = GETIMS('SEGMENT=SUBURB  ',
                ' WHERE SEGMENT=SUBURB #SUBURB-NAME=CITY BEACH')
PRINT(INREC,CHAR)
PRINT(INREC,HEX)
PRINT(INREC,SNGL)
PRINT(INREC,TABL)
/+
⋮
```

REPLIMS CALL (Replace an IMS segment):

Syntax



Notes:

1 To use field reference numbers, you must have already loaded a view, by preceding this function call with a call to VIEWIMS.

repsegm The name of the segment you want to update.

NEXT Replaces the next segment in the database after the current segment that satisfies the WHERE clause.

FIRST Replaces the first segment in the database that satisfies the WHERE clause.

conkey The concatenated key of the segment you want to update.

segname The name of a segment in the hierarchical path of the segment that you want to update.

field_ref A field reference number.

field_name A field name.

field_value The value of the field specified by *field_ref* or *field_name*.

Returns zero if successful, otherwise nonzero.

If FM/IMS encounters an unavailable HALDB partition or DEDB area, the REPLIMS function receives a return code of 8 and the ISPF variable FMSTATUS indicates an IMS status code of:

FH an unavailable DEDB area

BA an unavailable HALDB partition

To update the contents of the current segment, omit the WHERE clause. Otherwise, use the WHERE clause to identify the segment you want to update.

To replace the current contents of the segment with the contents of the OUTREC variable, omit the SET clause. If you are inserting a variable-length segment, then the first two bytes of OUTREC must specify the length of the segment.

Otherwise, use the SET clause to specify replacement field values for the segment.

When you use OUTREC, you replace the entire contents of the segment. When you use the SET clause, you can replace only the field values that you specify; other field values remain unchanged.

IEB example using REPLIMS

```
⋮
$$FILEM IEB
⋮
$$FILEM     PROC=*

filerc = VIEWIMS('TPLDSN=FMN.IMS.IVP.TEMPLATE')

filerc = GETIMS('SEGMENT=SUBURB  ',
               ' WHERE SEGMENT=SUBURB #SUBURB-NAME=CITY BEACH')

 /* Replace a segment using OUTREC */
         /* or */
OUTREC = OVERLAY('6168',outrec,19)
filerc = REPLIMS('SEGMENT=SUBURB')
         /* or */
 /* Replace a segment using SET */

rtrc=REPLIMS('SEGMENT=SUBURB ',
             'SET #POST-CODE=6168',
            ' WHERE SEGMENT=SUBURB #SUBURB-NAME=CITY BEACH')
/+
⋮
```

SAVEIMS CALL (Commit IMS changes):

Syntax

```
►►──SAVEIMS──────────────────────────────────────────────────►◄
```

Returns zero if successful, otherwise nonzero.

IEB example using SAVEIMS

```
⋮
$$FILEM IEB
⋮
$$FILEM     PROC=*

filerc = GETIMS('SEGMENT=SUBURB NEXT',

OUTREC = OVERLAY('1968',outrec,19)
filerc = REPLIMS('SEGMENT=SUBURB')
filerc = SAVEIMS
⋮
```

SETRC CALL (Set Job Step return code): The IEB call sets the Job Step return code to the value specified. It does not cause FM/IMS to exit the REXX procedure.

The format of the call is:

Syntax

```
►►──SETRC─(rc)───────────────────────────────────────────────►◄
```

rc Return code.

IEB example using SETRC:

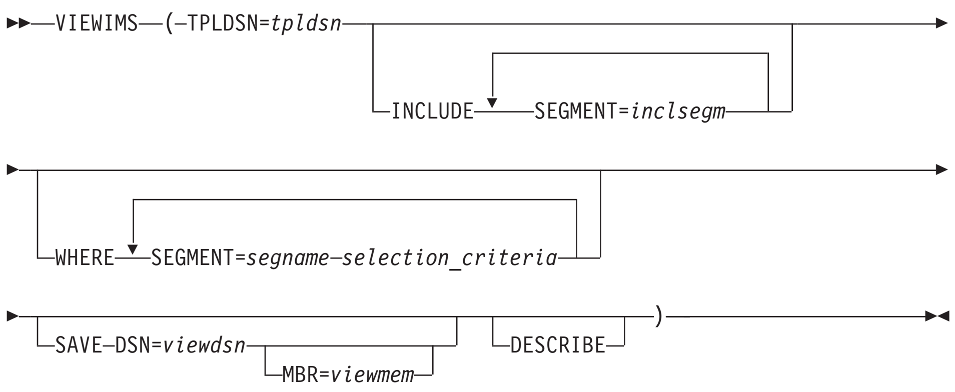
```
RTRC=GETIMS(USING_VIEW)
IF RTRC<>0 THEN DO
   SETRC(21)
   EXIT
END
```

VIEWIMS CALL (Load or create a view):

Syntax (to load an existing view or template)

```
►►──VIEWIMS──────────────────────────────────────────────────►

►─(──┬──┬─VIEWDSN=viewdsn─┬──┬─────────────────────┬─┬──┬──────────┬─)──────►◄
     │  └─VIEWDD=viewdd───┘  └─VIEWMBR=viewmem─────┘ │  └─DESCRIBE─┘
     └──┬─TPLDSN=tpldsn─┬────────────────────────────┘
        └─TPLDD=tpldd───┘
```

Syntax (to create a new view from an existing template)



If you specify the DESCRIBE parameter, then VIEWIMS prints a report showing the structure of the view or template for each segment in the database.

viewdsn
One of:
- The name of the sequential data set containing the view.
- The name of the PDS containing the view. (You then specify the member name in *viewmem*.)

viewdd The DDNAME containing one of the following:
- The sequential data set containing the view.
- The PDS containing the view. You then specify the member name in *viewmem*.

viewmem
The member name of the view.

tpldsn The name of the data set containing the template that is used as a basis to create a view. The template for the DBD must have already been created on-line in FM/IMS.

tpldd The DDNAME containing the template data set that is used as a basis to create a view. The template for the DBD must have already been created on-line in FM/IMS.

inclsegm
The name of a segment to be included in the view.

If you call the GETIMS function with the USING_VIEW parameter, then only the included in the view is retrieved.

However, if you omit the INCLUDE clause entirely, then all segments are included in the view.

segname
The name of a segment for which you want to specify field selection criteria.

selection_criteria
Field selection criteria for the *segname* segment. Determines which segment occurrences are retrieved by calls to the GETIMS function with the USING_VIEW parameter.

For details on specifying field selection criteria, see page 86.

IEB example using VIEWIMS

```
⋮
$$FILEM IEB
⋮
$$FILEM     PROC=*
  rtrc=VIEWIMS('TPLDSN=FMN.IMS.IVP.TEMPLATE ',
              'SAVE     DSN=FMN.IMS.IVP.VIEW MBR=ZZ999 ',
              'INCLUDE  SEGMENT=SUBURB ',
              '         SEGMENT=LINKSTR ',
              'WHERE    SEGMENT=SUBURB substr(#2,1,1)=F ',
              '         SEGMENT=LINKSTR #4=ST ',
              'DESCRIBE')
/+
⋮
```

Example: Here is an example of maintaining an IMS database in DLI mode using a static PSB.

```
//FMNIEDIT 'Static PSB edit'
//*
//FILEMAN  EXEC PGM=FMN1IMSB
//STEPLIB  DD DSN=FMN.SFMNMOD1,DISP=SHR
//SYSPRINT DD SYSOUT=*,RECFM=FBA,LRECL=133
//FMNTSPRT DD SYSOUT=*
//FMN1JIN  DD DISP=SHR,DSN=FMN.SFMNSLIB
//FMIMSIN  DD *
$$FILEM IEB REGNTYPE=DLI,
$$FILEM     PSBTYPE=STATIC,
$$FILEM     PSBDSN=IMSV910.FMI.PSBLIB,
$$FILEM     PSBMEM=PUPA,
$$FILEM     PCBNUM=(3,2,1),
$$FILEM     DBDDSN=FMN.IMS.IVP.DBDLIB,
$$FILEM     DBDDSN2=IMSV910.FMI.DBDLIB,
$$FILEM     DBDMEM=DJ2E,
$$FILEM     IMSID=IF42,
$$FILEM     PROC=*
filerc = VIEWIMS('TPLDSN=FMN.IMS.IVP.TEMPLATE')
filerc = VIEWPOSN('POSSEGM=NSTREET ',
                 'SEGMENT=SUBURB #SUBURB-NAME=CITY BEACH ',
                 'SEGMENT=NSTREET #2=FLOREAT BEACH ')

filerc = GETIMS(NEXT USING_VIEW)

 /* Insert an NSTREET segment using SET */
filerc = ISRTIMS('SEGMENT=NSTREET ',
                 ' SET #2=FLOREAT PARK ')
/+
 /*
//* -------------------------------------------------------------
//* -- Database data sets                                      --
//* -------------------------------------------------------------
//DJ2E DD DSN=FMN.IMS.IVP.DB.DJ2E,DISP=SHR
//DJ2F DD DSN=FMN.IMS.IVP.DB.DJ2F,DISP=SHR
//DJ3E DD DSN=FMN.IMS.IVP.DB.DJ3E,DISP=SHR
//DJ3F DD DSN=FMN.IMS.IVP.DB.DJ3F,DISP=SHR
//DJ1E DD DSN=FMN.IMS.IVP.DB.DJ1E,DISP=SHR
//DJ1F DD DSN=FMN.IMS.IVP.DB.DJ1F,DISP=SHR
```

General tips for using the REXX functions defined by IEB and IBB: Rather than repeating this information under each function description, here are some general tips:

What is the ″current″ segment?

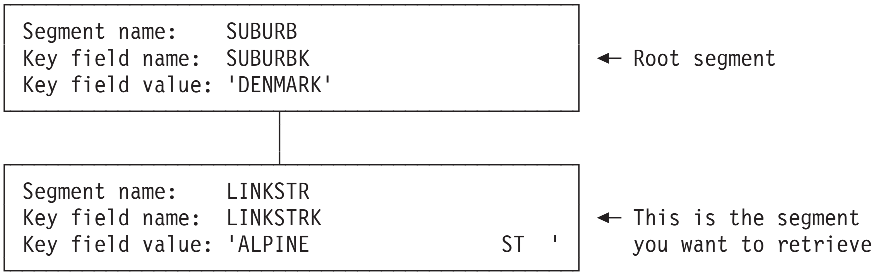
Several of the function descriptions refer to the "current" segment. This is the segment most recently retrieved, inserted or replaced by a GETIMS, ISRTIMS or REPLIMS function.

Locating a segment by specifying its key values

With the DELIMS, GETIMS, ISRTIMS and REPLIMS functions, you can use a WHERE clause to locate a segment according to its key values. On the WHERE clause, you can specify one of:

- A complete concatenated key as a single parameter, CKEY.
- The segment names and key fields for one or more levels in the hierarchical path of the segment you want, as separate parameters.

  For example, given the following segment hierarchy:



  then you would specify:

```
GETIMS('SEGMENT=LINKSTR ',
       'WHERE SEGMENT=SUBURB #SUBURBK=DENMARK ',
             'SEGMENT=LINKSTR #LINKSTRK=ALPINE                ST')
```

  You do not have to specify all the segments in the hierarchical path. If, for example, there is only one LINKSTR segment with the key ′ALPINE ST ′ in the entire database, you can specify:

```
GETIMS('SEGMENT=LINKSTR',
  'WHERE SEGMENT=LINKSTR #LINKSTRK=ALPINE                ST')
```

  If ′ALPINE ST ′ is the first or only LINKSTR segment under the SUBURB segment with a key of DENMARK, you can specify:

```
GETIMS('SEGMENT=LINKSTR',
  'WHERE SEGMENT=SUBURB #SUBURBK=DENMARK')
```

  You can specify the fields using either the field names (as in the previous examples) or their reference numbers (such as #1, #2). Field values can be placed directly after the equals (=) sign or inside quotes after the = sign. Quotes are required if the field value contains leading or trailing spaces which would otherwise be ignored. In this way:

```
GETIMS('SEGMENT=LINKSTR',
  'WHERE SEGMENT=LINKSTR #LINKSTRK=ALPINE                ST')
```

  and

```
GETIMS('SEGMENT=LINKSTR',
  'WHERE SEGMENT=LINKSTR #LINKSTRK=   ALPINE                ST   ')
```

  are evaluated as identical, but

```
GETIMS('SEGMENT=LINKSTR',
  'WHERE SEGMENT=LINKSTR #LINKSTRK="   ALPINE                ST   "')
```

  is evaluated differently.

  To list these reference numbers, you can use either the FM/IMS ISPF panels to browse the template for this DBD, or the VIEWIMS DESCRIBE REXX function with IEB.

You can mix field names and reference numbers in the same WHERE clause.

Before you can use this type of WHERE clause, you must call the VIEWIMS function, and load a template or view that matches the DBD you are using. This enables the WHERE clause to match the field names or reference numbers with the appropriate fields in the IMS segments.

When the WHERE clause is on a GETIMS, DELIMS or REPLIMS call, you may specify the FIRST parameter or NEXT parameter before it. When the FIRST parameter is specified, the call locates the first segment in the database that satisfies the WHERE clause. When the NEXT parameter is specified, the call locates the next segment in the database after the current segment that satisfies the WHERE clause. The default is NEXT.

When the WHERE clause is on an ISRTIMS call, the call locates the first segment in the database that satisfies the WHERE clause.

Referring to a previous current segment

The FMCONKEY REXX variable contains the concatenated key of the segment deleted, retrieved, inserted or replaced by the previous DELIMS, GETIMS, ISRTIMS or REPLIMS function call. Similarly, the FMSEGNM REXX variable contains the name of that segment.

If you need to use the FMCONKEY and FMSEGNM values later, you can save them in your own defined variables. These user defined variables can then be used with the SEGMENT and CKEY parameters.

Use of the filerc variable: A variable called ’filerc’ is used in several examples. This method is used to get hold of the REXX return code which can be used for testing. You can call the variable any valid REXX name. You can use different names for different statements or you do not need to use a variable at all if you are not going to use the return code.

This variable does not hold the DL/I status code. FMSTATUS is designed for this purpose.

Batch Browse (IBB):

Purpose

Runs a REXX procedure that can display or print segments from a data base. In addition to using any other valid REXX in this procedure, IBB defines some new REXX functions that you can use to retrieve or print segments, or to create a view.

Usage notes

For general tips for using IBB, see “General tips for using the REXX functions defined by IEB and IBB” on page 124. For note about the use of the filerc variable, see “Use of the filerc variable.”

The Batch Browse JCL generated by FM/IMS has the following flows:

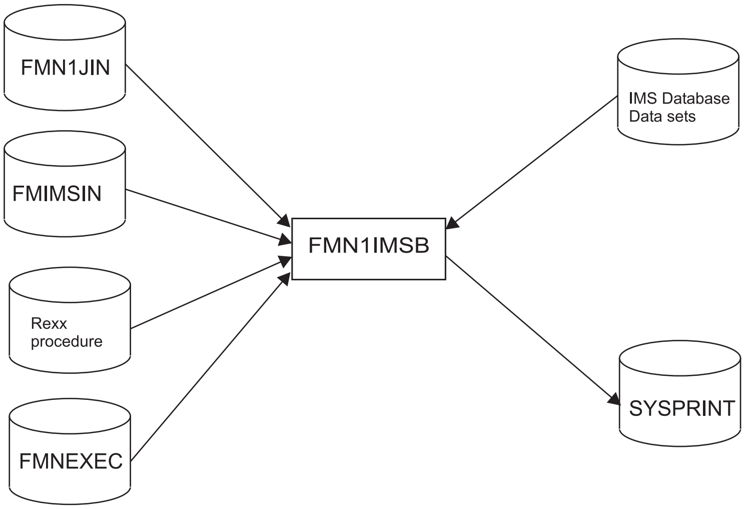


Figure 11. Flows for Batch Browse (IBB) step

FMN1JIN
Points to the FM/IMS skeleton library. This DD statement is only used when you are browsing HALDB databases.

FMIMSIN
Contains the control statements for the batch browse. These control statements are discussed below.

IMS database data sets
In DLI mode, FM/IMS includes DDs for each of the IMS database data sets associated with the IMS database being maintained. If the database has logical relationships, the data sets for the logically related databases need to be included.

SYSPRINT
This contains the FM/IMS messages that were issued by FM/IMS while running the IEB REXX procedure.

REXX Procedure
The REXX procedure which is to be run to view the IMS database in batch. This REXX procedure consists of standard REXX statements combined with FM/IMS IEB calls to allow you to read the IMS segments.

In FMIMSIN, if PROC=* is specified, then the REXX procedure is specified instream after the PROC=* parameter.

In FMIMSIN, if PROC=<*member*> is specified, then the REXX procedure <*member*> found in the FMNEXEC library is run.

FMNEXEC
This contains the member that is run if PROC=<*member*> is specified in the FMIMSIN parameters.

FMIMSIN control statements: You use the FMIMSIN control statements to specify:

- If the Batch Browse is to run in a BMP or DLI region.
- Whether to use a static PSB, or if FM/IMS is to generate a dynamic PSB to view the IMS data.
- Whether the IBB REXX procedure to be run is instream (PROC=*), or can be found in the library specified on the FMNEXEC DD.

Each line in FMIMSIN starts with a $$FILEM and the first line contains the parameter IBB to specify that FM/IMS is performing Batch Browse.

```
$$FILEM IBB REGNTYPE=DLI,
$$FILEM     PSBTYPE=DYNAMIC,
$$FILEM     DBDDSN=FMN.IMS.IVP.DBDLIB,
. . .
```

The syntax of the control statements for a Batch Browse is shown here.

Syntax - part 1

```
                        (1)
►►──IBB──IMSID=imsid─────────┬─REGNTYPE=BMP─┤ BMP parameters ├─┬──────────────────────────►
                             └─REGNTYPE=DLI─┤ DLI parameters ├─┘

                                       (2)
   ┌─PSBTYPE=installation default─────────────┐
►──┼──────────────────────────────────────────┼─DBDDSN=dbddsn──┬─────────────────────┬──►
   ├─PSBTYPE=DYNAMIC─┤ DYNAMIC parameters ├───┤                └─DBDDSN2=dbddsn2─────┘
   └─PSBTYPE=STATIC──┤ STATIC parameters ├────┘

                   ┌─VIEW=N──────────────────────────────────────────┐
►──DBDMEM=dbdname──┼─────────────────────────────────────────────────┼──┬─PROC=proc──────┬──►◄
                   │          ┌─VIEWDD=VIEWDD───────────────────┐    │  └─PROC=asterisk──┘
                   └─VIEW=Y───┼─────────────────────────────────┼────┘
                              ├─VIEWDD=ddname───────────────────┤
                              └─VIEWDSN=VIEWdsn──┬───────────────────┬┘
                                                 └─VIEWMEM=VIEWmem───┘
```

BMP parameters:

```
                                        (3)                                (3)
               ┌─PARDLI=subsystem default─┐    ┌─IMSNBA=subsystem default─┐
├──┬───────────┬──┼──────────────────────────┼────┼──────────────────────────┼──────────►
   └─AGN=agn───┘  ├─PARDLI=0─────────────────┤    └─IMSNBA=number────────────┘
                  └─PARDLI=1─────────────────┘

                                  (3)                               (3)
   ┌─IMSOBA=subsystem default─────┐   ┌─LOCKMAX=subsystem default─┐
►──┼──────────────────────────────┼───┼───────────────────────────┼────────────────────┤
   └─IMSOBA=number────────────────┘   └─LOCKMAX=number────────────┘
```

Notes:

1 If REGNTYPE=DLI is specified and the FM/IMS installation options module specifies COMPAT=Y, IMSID is an optional keyword. Otherwise, IMSID is a required keyword (as shown).

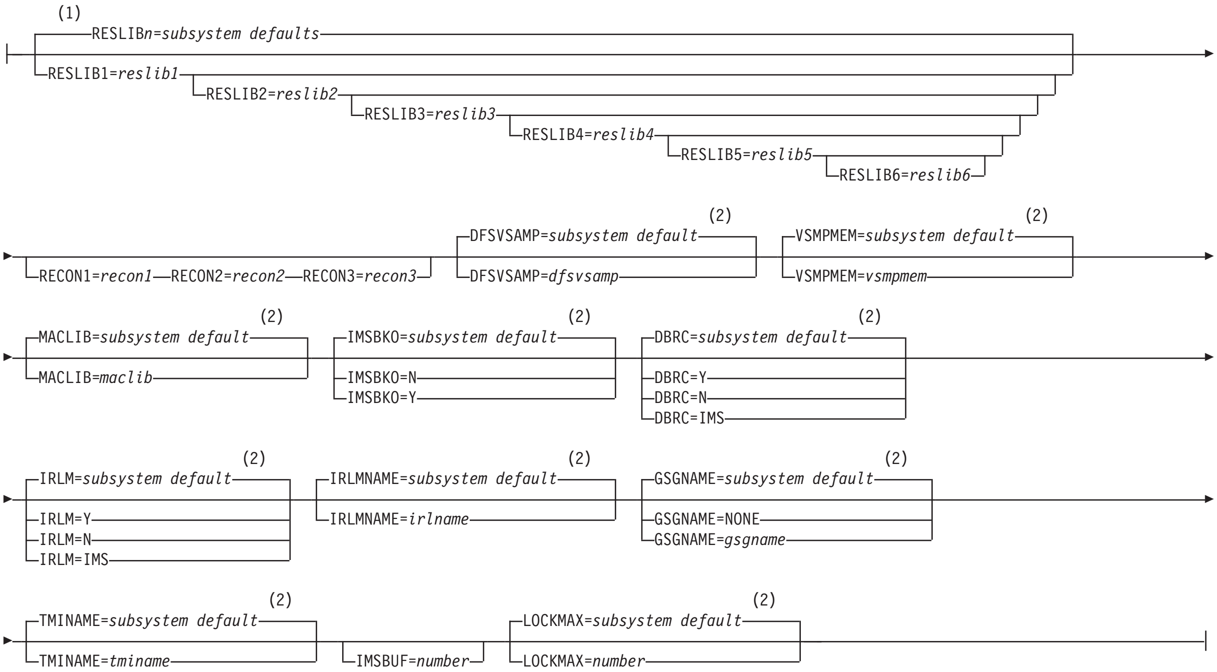
2 If you do not specify PSBTYPE, FM/IMS uses the PSBTYPE specified in the FM/IMS installation options module.

3 If you do not specify this parameter, FM/IMS uses the *subsystem* default for the parameter (if you have specified the IMSID parameter), or the *installation* default for the parameter (if you have not specified the IMSID parameter).

The FM/IMS Administrator specifies the subsystem and installation defaults at installation time.

Syntax - part 2

DLI parameters:



Notes:

1 If you do not specify the RESLIB parameters, FM/IMS uses the *subsystem* defaults for the parameters (if you have specified the IMSID parameter), or the *installation* defaults for the parameters (if you have not specified the IMSID parameter).

2 If you do not specify this parameter, FM/IMS uses the *subsystem* default for the parameter (if you have specified the IMSID parameter), or the *installation* default for the parameter (if you have not specified the IMSID parameter).

The FM/IMS Administrator specifies the subsystem and installation defaults at installation time.

Syntax - part 3

DYNAMIC parameters:

```
                                 (1)
    ┌PROCOPT=subsystem default──────┐
├───┤                               ├────────────────────────────────────────────────────────┤
    ├PROCOPT=G──────────────────────┤
    └PROCOPT=GO─────────────────────┘
```

STATIC parameters:

```
├──PSBDSN=psbdsn──PSBMEM=psbmbr──┬──────────────────────────┬──────────────────────────────┤
                                 │        ┌,───────┐        │
                                 ├PCBNUM=(─▼─pcbnum─┴─)─────┤
                                 │         ┌,────────┐      │
                                 └PCBNAME=(─▼─pcbname─┴─)───┘
```

Notes:

1 If you do not specify this parameter, FM/IMS uses the *subsystem* default for the parameter (if you have specified the IMSID parameter), or the *installation* default for the parameter (if you have not specified the IMSID parameter).

The FM/IMS Administrator specifies the subsystem and installation defaults at installation time.

The VIEW, VIEWDD, VIEWDSN, VIEWMEM, and PROC parameters are described here. All the other parameters in this syntax diagram can also be specified for the Extract (IXB) function. For a description of these common parameters, see "Extract (IXB)" on page 71.

VIEW Optional parameter. Specifies whether or not the Batch Browse uses a view.

N The Batch Browse does not use a view.

Y The Batch Browse uses a view. If Y is specified, you must specify the view that is to be used as outlined in the VIEWDD, VIEWDSN, and VIEWMEM entries.

VIEWDD=*ddname*
Identifies the DD statement that specifies the sequential data set or PDS(E) and member name that contains the view you want the Batch Browse to use. The default is VIEWDD.

VIEWDSN=*viewdsn*
The name of the data set that contains the view that you want your Batch Browse to use. You may specify a sequential data set or a PDS(E). If you specify a PDS(E), the parameter may also include a member name in parentheses.

VIEWMEM=*viewmem*
Required when the VIEWDSN or VIEWDD parameter specifies a PDS(E) and does not specify the member. Specifies the member of the PDS(E) that contains the view you want the Batch Browse to use.

PROC=*proc*
Required parameter. Valid values are:

The name of a PDS(E) member
The name of the PDS(E) member containing the REXX procedure that you want the Batch Browse to use.

You specify the PDS(E) containing this member in the FMNEXEC DD statement. The JCL for the batch edit must include this DD statement.

An asterisk (*)
Indicates that the REXX procedure is inline.

The procedure is read from SYSIN immediately after the FMIMSIN control statements. The inline procedure is terminated by a record containing a slash and a plus sign (/+) in columns 1–2.

REXX IBB calls: The REXX procedure FM/IMS invokes standard TSO/E REXX. However, you are able to call IBB functions to be able to retrieve and print the IMS segments. You are also able to create FM/IMS view for use in the IBB REXX procedure.

When running this REXX procedure and calling IBB functions, you process the database **either using a FM/IMS view or by issuing IMS calls**. FM/IMS does not allow you to swap between using a view and not using a view.

By referencing or creating a view using the IBB VIEWIMS function and issuing a GETIMS with the USING_VIEW parameter, FM/IMS uses the selection criteria specified in the view to retrieve the relevant segment.

When not using a view, you have to navigate the database by issuing a GETIMS call to retrieve the relevant segments. This method is similar to issuing IMS calls in a program. You need to determine how the segments are to be retrieved.

Retrieval using a view: The following IBB calls are available when processing the database using a view:

GETIMS CALL (Retrieve an IMS segment)
The same as for Batch Edit. See "GETIMS CALL (Retrieve an IMS segment)" on page 106.

PRINT CALL (Print an IMS segment)
The same as for Batch Edit. See "PRINT CALL (Print an IMS segment)" on page 109.

SETRC CALL (Set Job Step return code)
The same as for Batch Edit. See "SETRC CALL (Set Job Step return code)" on page 111.

VIEWIMS CALL (Load or create a view)
The same as for Batch Edit. See "VIEWIMS CALL (Load or create a view)" on page 111.

VIEWPOSN CALL (Position on an IMS segment)
The same as for Batch Edit. See “VIEWPOSN CALL (Position on an IMS segment)” on page 113.

Retrieval without a view: The following calls are available when processing the database without using a view.

A view may be loaded to allow the WHERE clause to specify key fields, or to print the database in SNGL or TABL format but the GETIMS call is issued without the USING_VIEW parameter.

GETIMS CALL (Retrieve an IMS segment)
The same as for Batch Edit, except that the HOLD parameter is not available. See “GETIMS CALL (Retrieve an IMS segment)” on page 115.

PRINT CALL (Print an IMS segment)
The same as for Batch Edit. See “PRINT CALL (Print an IMS segment)” on page 119.

SETRC CALL (Set Job Step return code)
The same as for Batch Edit. See “SETRC CALL (Set Job Step return code)” on page 122.

VIEWIMS CALL (Load or create a view)
The same as for Batch Edit. See “VIEWIMS CALL (Load or create a view)” on page 122.

Example: Here is an example of maintaining an IMS database in DLI mode using a static PSB.

```
FMNIBRWS 'Static PSB browse'
//*
//FILEMAN  EXEC PGM=FMN1IMSB
//STEPLIB  DD DSN=FMN.SFMNMOD1,DISP=SHR
//SYSPRINT DD SYSOUT=*,RECFM=FBA,LRECL=133
//FMNTSPRT DD SYSOUT=*
//FMN1JIN  DD DISP=SHR,DSN=FMN.SFMNSLIB
//FMIMSIN  DD *
$$FILEM IBB REGNTYPE=DLI,
$$FILEM     PSBTYPE=STATIC,
$$FILEM     PSBDSN=IMSV910.FMI.PSBLIB,
$$FILEM     PSBMEM=PUPA,
$$FILEM     PCBNUM=(3,2,1),
$$FILEM     DBDDSN=FMN.IMS.IVP.DBDLIB,
$$FILEM     DBDDSN2=IMSV910.FMI.DBDLIB,
$$FILEM     DBDMEM=DJ2E,
$$FILEM     IMSID=IF42,
$$FILEM     PROC=*
filerc = VIEWIMS('TPLDSN=FMN.IMS.IVP.TEMPLATE')
filerc = VIEWPOSN('POSSEGM=NSTREET ',
                 'SEGMENT=SUBURB #SUBURB-NAME=CITY BEACH ',
                 'SEGMENT=NSTREET #2=FLOREAT BEACH ')

Do while filerc=0
   filerc = GETIMS(NEXT USING_VIEW)
   PRINT(INREC,TABL)
END

/+
 /*
//* --------------------------------------------------------------
//* -- Database data sets                                       --
//* --------------------------------------------------------------
//DJ2E DD DSN=FMN.IMS.IVP.DB.DJ2E,DISP=SHR
//DJ2F DD DSN=FMN.IMS.IVP.DB.DJ2F,DISP=SHR
```

```
//DJ3E DD DSN=FMN.IMS.IVP.DB.DJ3E,DISP=SHR
//DJ3F DD DSN=FMN.IMS.IVP.DB.DJ3F,DISP=SHR
//DJ1E DD DSN=FMN.IMS.IVP.DB.DJ1E,DISP=SHR
//DJ1F DD DSN=FMN.IMS.IVP.DB.DJ1F,DISP=SHR
```

Initialize (DIB):

Purpose

The Initialize dialog generates JCL that runs the Initialize (DIB) function. The Initialize function initializes databases.

Usage notes

The Batch database initialization JCL created by FM/IMS has the following flows:

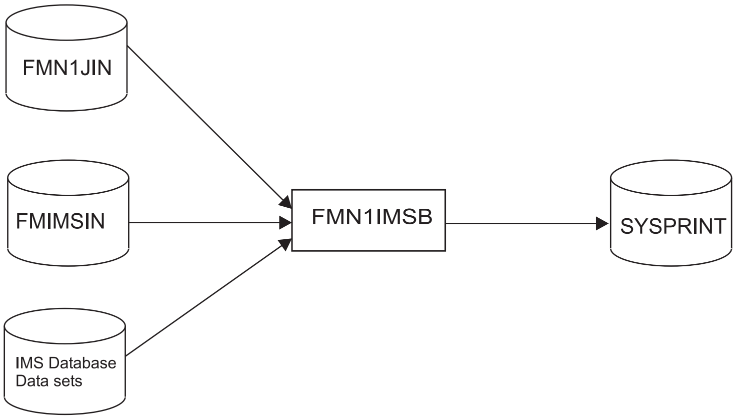


Figure 12. Flows for Initialize (DIB) step

FM/IMS can initialize the IMS database data sets associated with an IMS database.

If you select Delete/Define IMS data sets on the Initialize Entry panel FM/IMS retrieves the data set information of the data sets listed on the Database Data Set Specification panel and builds JCL to delete or define these data sets.

The Initialize utility can be executed for these types of IMS databases:

Full function

- HDAM, Hierarchical Direct Access Method
- HIDAM, Hierarchical Index Direct Access Method
- HISAM, Hierarchical Index Sequential Access Method
- SHISAM, Simple Hierarchical Index Sequential Access Method
- HSAM, Hierarchical Sequential Access Method
- SHSAM, Simple Hierarchical Sequential Access Method

The Initialize process for a full function database depends on whether the database has secondary indexes, logical relationships or neither.

The simplest steps to initialize a full function database are:

1. Database initialization, insert a dummy segment.
2. Delete the dummy segment.

If the database has logical relationships the steps are:

1. Delete work datasets.
2. IMS Prereorganisation Utility (DFSURPR0)
3. Database initialization, insert a dummy segment

4. IMS Database Prefix Resolution Utility (DFSURG10)
5. IMS Database Prefix Update Utility (DFSURGP0)
6. Delete the dummy segment.
7. Delete work datasets.

If the database has secondary indexes the steps are:

1. 1. Delete work datasets.
2. IMS Prereorganisation Utility (DFSURPR0)
3. Database initialization, insert a dummy segment.
4. IMS Database Prefix Resolution Utility (DFSURG10)
5. HISAM Reorganization Unload Utility (DFSURUL0)
6. HISAM Reorganisation Load Utility (DFSURRL0)
7. Delete the dummy segment.
8. Delete work datasets.

Fast Path

- DEDB, Data Entry Database (Fast Path)

For a **DEDB** the IMS Utility DBFUMIN0 is used to initialize the areas.

HALDBs

- PHDAM, Partitioned Hierarchical Direct Access Method (HALDB)
- PHIDAM, Partitioned Hierarchical Index Direct Access Method (HALDB)

For a **HALDB** database the steps to initialize the database are:

1. Delete Work data sets.
2. Set the master HALDB database to PINIT (DSPURX00).
3. Initialize the HALDB partitions: IMS Prereorganisation Utility (DFSURPR0)
4. Generate IMS image copy (DSPURX00) only if ″Submit image copy″ has been selected. Please note that the HALDB partitions are not available until an image copy has been run.
5. Delete Work data set.

FMIMSIN

Contains the control statements for the Initialize. These control statements are discussed below.

IMS database data sets

In DLI mode, FM/IMS includes DDs for each of the IMS database data sets associated with the IMS database being printed. If the database has logical relationships, the data sets for the logically related databases need to be included.

SYSPRINT

This contains the FM/IMS messages that were issued by FM/IMS while extracting the IMS data, as well as the printed IMS segments.

FMN1JIN

Points to the FM/IMS skeleton library. This DD statement is only used when you are initializing HALDB databases.

FMIMSIN control statements: You use the FMIMSIN control statements to specify the databases the function is to initialize. The function can only be run in DLI mode, and always uses a dynamic PSB.

Initialize (DIB)

Each line in FMIMSIN starts with a $$FILEM and the first line contains the parameter DIB to specify that FM/IMS will initialize a database.

```
$$FILEM DIB REGNTYPE=DLI,
$$FILEM     DBDDSN=FMN.IMS.IVP.DBDLIB,
. . .
```

The syntax of the control statements for the Initialize is shown here.

Syntax

```
                                      (1)
►►──DIB───┬─DIBTYPE=INSERT─┬──IMSID=imsid─────DBDDSN=dbddsn──┬──────────────────┬──────────────►
          └─DIBTYPE=DELETE─┘                                 └─DBDDSN2=dbddsn2──┘

►─DBDMEM=dbdname──┤ DLI parameters ├─────────────────────────────────────────────────────────►◄
```

DLI parameters:

```
    (2)
├──┬───RESLIBn=subsystem defaults────────────────────────────────────────────────────┬──────►
   └─RESLIB1=reslib1──┬──────────────────────────────────────────────────────────────┤
                      └─RESLIB2=reslib2──┬───────────────────────────────────────────┤
                                         └─RESLIB3=reslib3──┬────────────────────────┤
                                                            └─RESLIB4=reslib4──┬─────┤
                                                                               └─RESLIB5=reslib5──┬──────────────────┤
                                                                                                  └─RESLIB6=reslib6──┘

                                                              (3)                          (3)
►──┬─────────────────────────────────────────────┬──┬─DFSVSAMP=subsystem default─┬──┬─VSMPMEM=subsystem default─┬──►
   └─RECON1=recon1──RECON2=recon2──RECON3=recon3─┘  └─DFSVSAMP=dfsvsamp──────────┘  └─VSMPMEM=vsmpmem───────────┘

                           (3)                          (3)                          (3)
►──┬─MACLIB=subsystem default─┬──┬─IMSBKO=subsystem default─┬──┬─IRLM=subsystem default─┬──────►
   └─MACLIB=maclib────────────┘  ├─IMSBKO=N─────────────────┤  ├─IRLM=Y─────────────────┤
                                 └─IMSBKO=Y─────────────────┘  ├─IRLM=N─────────────────┤
                                                               └─IRLM=IMS───────────────┘

                             (3)                                            (3)
►──┬─IRLMNAME=subsystem default─┬──┬─────────────────┬──┬─LOCKMAX=subsystem default─┬──────────┤
   └─IRLMNAME=irlname───────────┘  └─IMSBUF=number───┘  └─LOCKMAX=number────────────┘
```

Notes:

1 If the FM/IMS installation options module specifies COMPAT=Y, IMSID is an optional keyword. Otherwise, IMSID is a required keyword (as shown).

2 If you do not specify the RESLIB parameters, FM/IMS uses the *subsystem* defaults for the parameters (if you have specified the IMSID parameter), or the *installation* defaults for the parameters (if you have not specified the IMSID parameter).

3 If you do not specify this parameter, FM/IMS uses the *subsystem* default for the parameter (if you have specified the IMSID parameter), or the *installation* default for the parameter (if you have not specified the IMSID parameter).

The FM/IMS Administrator specifies the subsystem and installation defaults at installation time.

The DIBTYPE parameter is described here. All the other parameters in this syntax diagram can also be specified for the Extract (IXB) function. For a description of these common parameters, see “Extract (IXB)” on page 71.

DIBTYPE
: The Initialize step to be run.

  INSERT
  : Specifies the step that inserts a dummy segment into the database.

  DELETE
  : Specifies the step that deletes the dummy segment from the database.

Example: Here is an example of the generated JCL for initializing a HDAM/OSAM database without logical relations or secondary indexes. You can also initialize such databases and all necessary JCL will be generated.

```
//* -------------------------------------------------------------
//*
//*       Delete/Define Database Data Sets
//*
//* -------------------------------------------------------------
//*
//PSDELETE EXEC PGM=IEFBR14
//HDOU DD DSN=FMN.IMS.HDOU,
//            DISP=(OLD,DELETE)
//*
//PSALLOC  EXEC PGM=IEFBR14
//HDOU DD DSN=FMN.IMS.HDOU,
//            DISP=(NEW,CATLG,DELETE),
//            SPACE=(TRK,(1,1)),
//            DCB=(RECFM=FBS,LRECL=2048,BLKSIZE=2048),
//            DSORG=PS,
//            MGMTCLAS=MCFM,
//            UNIT=3390,
//            STORCLAS=SCFM
 /*
//*
//* -------------------------------------------------------------
//*
//*       LOAD - INSERT DUMMY RECORDS
//*
//* -------------------------------------------------------------
//*
//TESTRC    IF (RC LE 4) THEN
//FILEMAN  EXEC PGM=FMN1IMSB
//STEPLIB  DD DSN=FMN.SFMNMOD1,DISP=SHR
//SYSPRINT DD SYSOUT=*,RECFM=FBA,LRECL=133
//FMNTSPRT DD SYSOUT=*
//FMN1JIN  DD DSN=FMN.SFMNSLIB,DISP=SHR
//SYSTERM  DD SYSOUT=*
//FMIMSIN  DD *
$$FILEM DIB DIBTYPE=INSERT,
$$FILEM     DBDDSN=FMN.IMS.IVP.DBDLIB,
$$FILEM     DBDMEM=HDOU,
$$FILEM     IMSID=IF52
 /*
//IEFRDER DD DUMMY
//* -------------------------------------------------------------
//* -- Database datasets                                       --
//* -------------------------------------------------------------
//HDOU DD DSN=FMN.IMS.HDOU,DISP=SHR
//ENDRC     ENDIF
//*
//* -------------------------------------------------------------
//*
//*       DELETE DUMMY RECORDS
```

```
//*
//* ---------------------------------------------------------------
//*
//TESTRC    IF (RC LE 4) THEN
//FILEMAN  EXEC PGM=FMN1IMSB
//STEPLIB  DD DSN=FMN.SFMNMOD1,DISP=SHR
//SYSPRINT DD SYSOUT=*,RECFM=FBA,LRECL=133
//FMNTSPRT DD SYSOUT=*
//FMN1JIN  DD DSN=FMN.SFMNSLIB,DISP=SHR
//SYSTERM  DD SYSOUT=*
//FMIMSIN  DD *
$$FILEM DIB DIBTYPE=DELETE,
$$FILEM     DBDDSN=FMN.IMS.IVP.DBDLIB,
$$FILEM     DBDMEM=HDOU,
$$FILEM     IMSID=IF52
 /*
//IEFRDER  DD DSN=&&LOGD,
//       UNIT=SYSALLDA,
//       SPACE=(TRK,(20,20),RLSE),
//       DCB=(RECFM=VB,LRECL=6140,BLKSIZE=6144),
//       DISP=(NEW,DELETE,DELETE)
//* ---------------------------------------------------------------
//* -- Database data sets                                        --
//* ---------------------------------------------------------------
//HDOU DD DSN=FMN.IMS.HDOU,DISP=SHR
//*
//ENDRC     ENDIF
```

Print Audit Report (AUD):

Purpose

Print Audit Report (AUD) is a batch function that allows you to print the audit reports to a data set, the terminal or DDNAME SYSPRINT, depending on the current settings of the PRINTOUT print option.

Usage notes

The Batch Print Audit Report JCL created by FM/IMS has the following flows

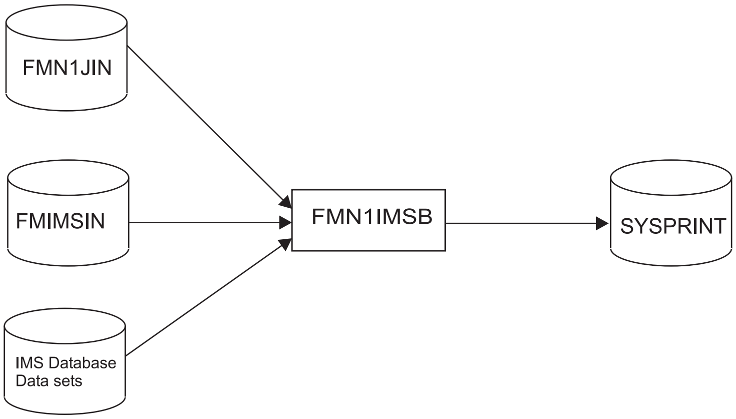


Figure 13. Flows for Batch Print Audit Report (AUD) step

The data in the Audit Trail report may be translated, depending on the current setting of the PRTTRANS print option. When this option is set to OFF, no translation occurs. When the PRTTRANS option is ON, unprintable characters are converted to displayable characters.

The PRTTRANS option can also be set to KN, meaning that unprintable characters are converted to displayable characters, and all lower case English is converted to uppercase.

When the Audit Trail data set is sent to the terminal, unexpected results such as terminal errors may occur if the data contains unprintable characters. It is recommended that PRTTRANS be set to ON or KN if an Audit Trail data set containing DBCS or other unprintable characters is to be viewed at the terminal.

SYSIN
This contains the FM/IMS parameters that are used by the AUD function. These parameters are discussed in further detail below.

SYSPRINT
This contains the FM/IMS messages and the printed Audit report.

SYSIN parameters: The SYSIN parameters are used by FM/IMS to determine how the Audit Trail report is to be printed. These parameters determine:

- If the report should be formatted or not.
- Whether or not only the changed fields should be printed.
- The format the IMS segments are printed in (CHAR, HEX, SNGL, or TABL).
- Whether or not the Audit Trail should be deleted after it has been printed.

Each line in SYSIN starts with a $$FILEM and the first line contains the parameter AUD to specify that FM/IMS is printing an Audit Trail report. TYPE=IMS is required.

```
$$FILEM AUD,
$$FILEM    TYPE=IMS,
$$FILEM    FORMAT=NO,
. . .
```

The format of the SYSIN for print audit report is:

Syntax

```
                                            ┌─CHANGED=YES─┐
►►──AUD──IMSID=imsid──┬─FORMAT=NO──┬──┼─────────────┼──DSNAME=audit trail log dataset──────────►◄
                      └─FORMAT=YES─┘  └─CHANGED=NO──┘
```

FORMAT
To display the report data in formatted character representation rather than hexadecimal format.

CHANGED
To report only fields that were changed, not whole segments.

DELETELOG
Select this option to delete the Audit Trail data set after the report has been generated. If this option is not selected, then the data set will be kept.

Example: Here is an example of the JCL for an Audit report where the print is not formatted and only fields that were changed are printed.

```
//FMBAT EXEC PGM=FILEMGR
//STEPLIB DD DSN=FMN.SFMNMOD1,DISP=SHR
//SYSPRINT DD SYSOUT=*
```

Print Audit Report (AUD)

```
//IDIOPTS DD DSN=FMN.IDIOPTS,DISP=SHR
//SYSIN DD *
$$FILEM AUD ,
$$FILEM TYPE=IMS,
$$FILEM FORMAT=NO,
$$FILEM CHANGED=YES,
$$FILEM DSNAME='FMNUSER.IMSAUDIT.D070926.T205717'
```

Formatted Audit Trail report:

```
IBM File Manager for z/OS IMS Component  Audit Trail Report                                                 Page 1
                                             Detailed Statistics Formatted                                  Date: 2007-09-26
DBD Name: DJ2E                                                                                              Time: 21.14.02
Database: FMN.IMS.IVP.DBDLIB

                       Checkpoint FM000001 Occurred
Data base Update Number: 1              Segment changed on 07/09/26 at 20.59.29
Segment Name: SUBURB Level: 1 Desc:
Concat Key: KORRELOCKING
                                              --------------------- Field Value -------------------------
 Level / Field name                  Format   -------- Before                        After --------------   Message

01 SUBURB                           AN    62
 02 SUBURB-NAME                     AN    18  KORRELOCKING                    KORRELOCKING
 02 POST-CODE                       ZD     4    6485                            6399
 02 SUBURB-TYPE                     AN     1
 02 PUBLIC-PHONE                    AN     4  K642                            K642
 02 DISTANCE-2-PERTH                ZD     4     174                             174
 02 CLOSEST-CENTRE                  AN    14  WYALKATCHEM                     WYALKATCHEM
 02 DISTANCE-2-CENTRE               ZD     3    10                              10
 02 MAP-REFERENCE                   AN     9
  03 LATITUDE-DEGREE                ZD     2    31                              31
  03 LATITUDE-MINUTE                ZD     2    12                              12
  03 LONGITUDE-DEGREE               ZD     3   117                             117
  03 LONGITUDE-MINUTE               ZD     2    28                              28
 02 ACRONYM                         AN     5  KORRE                           KORRE
** End of Segment Structure - Length = 62
Data base Update Number: 2              Segment inserted on 07/09/26 at 21.01.07
Segment Name: LETRBOX Level: 2 Desc:
Concat Key: KORRELOCKING      ,6966
                                              --------------------- Field Value -------------------------
 Level / Field name                  Format   -------- Before                        After --------------   Message

01 LETTER-BOX                       AN    34
 02 LETTER-BOX-KEY                  AN     4 K
  03 POST-CODE                      AN     4 K                              6966
 02 LOCATION                        AN    30                                WAY OUT IN THE BUSH

** End of Segment Structure - Length = 34
```

Figure 14. Audit Trail Report - Detail Statistics Formatted

Unformatted Audit Trail report:

```
IBM File Manager for z/OS IMS Component  Audit Trail Report                                                   Page 1
                                             Detailed Statistics Unformatted                                  Date: 2007-09-26
DBD Name: DJ2E                                                                                                Time: 21.03.04
Database: FMN.IMS.IVP.DBDLIB

                       Checkpoint FM000001 Occurred
Data base Update Number: 1              Segment changed on 07/09/26 at 20.59.29
Segment Name: SUBURB Level: 1 Desc:
Concat Key: KORRELOCKING
           Segment             1         2         3         4         5         6         7         8         9         10
Status   Start Length  1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
------------------------------------------------------------------------
BEFORE       1     62  KORRELOCKING      6485 K6420174WYALKATCHEM   010311211728KORRE
                       DDDDCDDCDCDC444444FFFF4DFFFFFFFEECDDCECCCD444FFFFFFFFFFFFDDDDC
                       26995363295700000064850264201746813213385400001031121172826995
CHANGED                *******************---******
AFTER                  KORRELOCKING      6399 K6420174WYALKATCHEM   010311211728KORRE
                       DDDDCDDCDCDC444444FFFF4DFFFFFFFEECDDCECCCD444FFFFFFFFFFFFDDDDC
                       26995363295700000063990264201746813213385400001031121172826995
CHANGED                *******************---******
Data base Update Number: 2              Segment inserted on 07/09/26 at 21.01.07
Segment Name: LETRBOX Level: 2 Desc:
Concat Key: KORRELOCKING      ,6966
           Segment             1         2         3         4         5         6         7         8         9         10
Status   Start Length  1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
------------------------------------------------------------------------
AFTER        1     34  6966WAY OUT IN THE BUSH
                       FFFFECE4DEE4CD4ECC4CEEC44444444444
                       6966618064309503850242800000000000
                       ****
```

Figure 15. Audit Trail Report - Detail Statistics Unformatted

Summary Audit Trail report:

```
IBM File Manager for z/OS IMS Component  Audit Trail Report                                                   Page 2
                                             Summary Statistics                                               Date: 2007-09-26
DBD Name: DJ2E                                                                                                Time: 21.03.04
Database: FMN.IMS.IVP.DB.DJ2E

                       Data base Opened on 2007-09-26 at 20.57.28 by userid: FMNUSER
                       Data base Closed on 2007-09-26 at 21.01.21
Audit Trail:
Description: EDIT THE SUBURB DATABASE
Data set Name: FMNUSER.IMSAUDIT.D070926.T205717
DBDlib:
Data set Name: FMN.IMS.IVP.DBDLIB
Member .  . : DJ2E
Template:
Data set Name: FMN.IMS.IVP.TEMPLATE
Member .  . : DJ2E
Total Segments Replace  : 1
              Inserted  : 1
               Deleted  : 0
TOTAL Data base Changes : 2
Records bypassed:       : 0
Failed deletes reported : 0
Audit Trail Report Options Used
            Print Format  .  .  .  .  .  .  .  .  .   : Formatted
            Maximum Print Lines per Page .  .  .   : 60
            Print Only Changed Fields on an Update : No
```

Figure 16. Audit Trail Report - Summary Statistics

UK32194, UK32195, UK32199, UK32200, UK32202, UK32203, UK32204, UK32205, UK32208, UK32209, UK32210, UK32211

Release Date: **21 December 2007**

This set of PTFs contains the following APAR fixes:

APAR #	APAR Abstract	Doc Impact
PK56026	V8 new function Q4 2007.	User's Guide and Reference (SC19-1239-00) User's Guide and Reference for CICS (SC19-1242-00)
PK56132	File Manager view load module information is incorrect.	None.
PK56252	FM/MVS abend S0C4-11 FMNMAIN FMNEDIT X'1950'.	None.
PK56796	S0C1 issuing VER command from 9672 processor.	None.
PK57215	ABEND0C4 PIC10 occurred if DEDB WAS updated with hex mode.	None.
PK57243	When using a template with multiple record types, the CEDIT function is not working as expected.	None.
PK57370	Empty, multi-volume input files causes DSC function to fail.	None.
PK57371	FMN0606I File Manager encountered an internal error - code 80000925.	None.
PK57457	DSCMP result inconsistent with different readahead limits.	None.
PK57824	FMNMAIN (audit) File Manager ABEND0C4 FMN0607I File Manager abended with code 840C4000 - REASON 00000011.	None.
PK57830	Abend in File Manager 3.13 when used in split screen.	None.
PK57889	ABENDS0C4 issuing profile command when being in browse mode of a VSAM file.	None.
PK57896	File Manager V8 new function Q4 2007 part 2.	None.
PK57897	File Manager V8 new function Q4 2007 part 3.	None.

APAR #	APAR Abstract	Doc Impact
PK57906	File Manager V8 new function Q4 2007 part 4.	None.
PK58025	FM/IMS error message ’REPLACE STATUS CODE DA’ issued when user attempts to edit data with CAPS ON.	None.

PK56026

Initial problem description

This APAR documents some enhancements and problems corrected by FM development.

Outline of solution

File Manager Base component enhancements:

1. Data Set List SCS 3.4 will allow now for simplified RENAME of entire VSAM clusters with automatic component names adjustment.

2. DSM Data Set Compare and DSP Data Set Print will add support for a new IGNLEN parameter in batch as well as panel options to allow for record length mismatch being ignored during processing and additional output formatting.

3. File Manager will now support initialization of never ″written to datasets″ referenced in WRITE statements with user procedures for all applicable functions.

4. File Manager AUD Audit Print batch function will now support specification of a DDNAME for input data set specification.

5. SET HEADERPG=NO processing for BATCH has been corrected to suppress default FM header page output.

6. DATAHDR processing has been enhanced to apply also to FCH batch processing (in addition to DSP).

7. DSC Data Set Copy utility will support member names specification of lowercase and unprintable characters (with certain limitations).

File Manager DB2 component enhancements:

1. Provide an interface to enable FM/DB2 functions to be invoked from an external application.

2. Allow FM/DB2 to be added to the applications shown on the DB2 Admin Tool workbench.

3. Provide a method for accessing the FM/DB2 editor from a DB2 Admin Tool list of tables.

File Manager CICS component and Rational Developer for System Z integration enhancements:

1. Provide Internet Protocol Version 6 (IPv6) support.

Documentation impact

This APAR requires changes to be made to:

- User’s Guide and Reference (SC19-1239-00)
- User’s Guide and Reference for CICS (SC19-1242-00)

Changes to the User’s Guide and Reference

Chapter 6, ″Managing data sets″: In the section, ″Printing from File Manager″, subsection ″Printing data sets″:

- After step 6, add a new step:

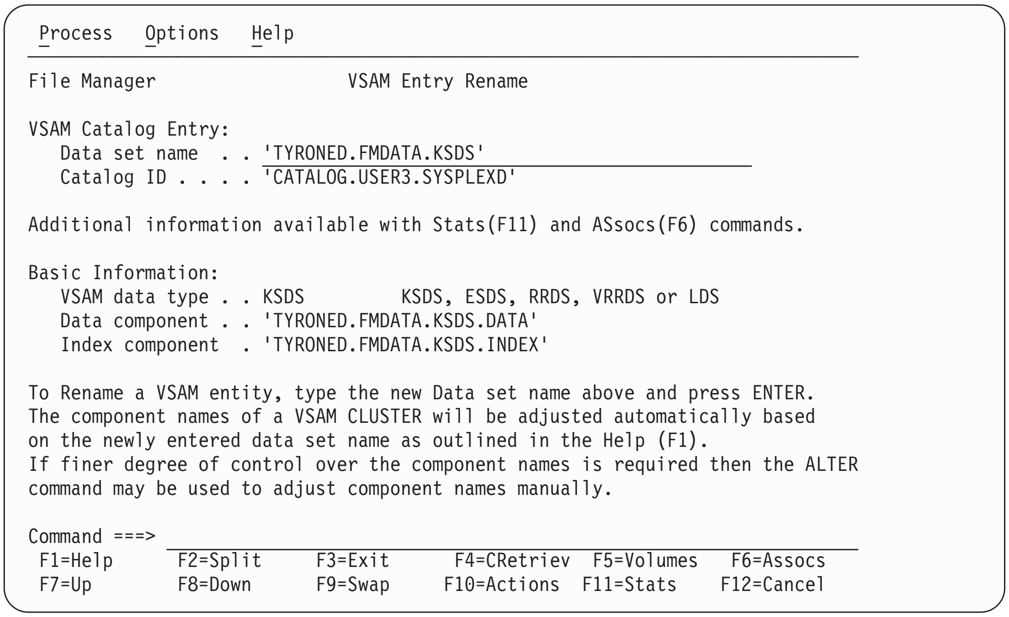
  7. Additional print formatting options are available after selecting the **Additional print options** option. These options allow for the printed output to contain additional information about record structure, as well as controlling the inclusion of records when their length does not match the layouts of the template records. For more details, see Figure 19 on page 146.
- Rename the existing step 7 to step 8.
- Under ″RELATED TOPICS″, add a reference to the (new) Print options panel.

Chapter 7, ″Using File Manager utilities″: In the section, ″Managing Catalog Entries″, subsection ″Altering an existing Catalog Entry″, add after step #2:

To rename a VSAM data set (CLUSTER and its corresponding components), enter REN(ame) in the line command field adjacent to the entry you want to change. File Manager displays the VSAM Entry Rename panel for the data type of your data set is displayed, showing the data set information that can be changed. The component names of a VSAM CLUSTER are adjusted automatically based on the newly entered data set name. If a finer degree of control over the component names is required, then the ALTER command may be used to adjust component names manually. To change any additional VSAM parameters, use the ALTER command.

Chapter 14, Panels and fields:

- Add the new **VSAM Entry Rename** panel as pictured here:



```
 Process   Options   Help
────────────────────────────────────────────────────────────────────────────
File Manager                  VSAM Entry Rename

VSAM Catalog Entry:
   Data set name  . . 'TYRONED.FMDATA.KSDS'
   Catalog ID . . . . 'CATALOG.USER3.SYSPLEXD'

Additional information available with Stats(F11) and ASsocs(F6) commands.

Basic Information:
   VSAM data type . . KSDS         KSDS, ESDS, RRDS, VRRDS or LDS
   Data component . . 'TYRONED.FMDATA.KSDS.DATA'
   Index component  . 'TYRONED.FMDATA.KSDS.INDEX'

To Rename a VSAM entity, type the new Data set name above and press ENTER.
The component names of a VSAM CLUSTER will be adjusted automatically based
on the newly entered data set name as outlined in the Help (F1).
If finer degree of control over the component names is required then the ALTER
command may be used to adjust component names manually.

Command ===> 
 F1=Help      F2=Split     F3=Exit      F4=CRetriev  F5=Volumes   F6=Assocs
 F7=Up        F8=Down      F9=Swap     F10=Actions  F11=Stats    F12=Cancel
```

Figure 17. VSAM Entry Rename panel

Parent panels:
– Catalog Services Data Set list panel
– Display VTOC Data Set List panel

Related tasks and examples:

–

- **Display VTOC Data Set List panel**:

  In the section, ″Available commands″:

  - In the description for ALTER(A), delete the last sentence ″Synonymous with RENAME.″.
  - Replace **RENAME (R)** and its description with:

    RENAME (REN)

    This is a synonym for Alter with non-VSAM data sets and it works identically. For VSAM data sets it invokes a separate VSAM Entry Rename panel.

- **Catalog Services: Data Set List panel**:

  In the section, ″Available commands″:

  - Add:

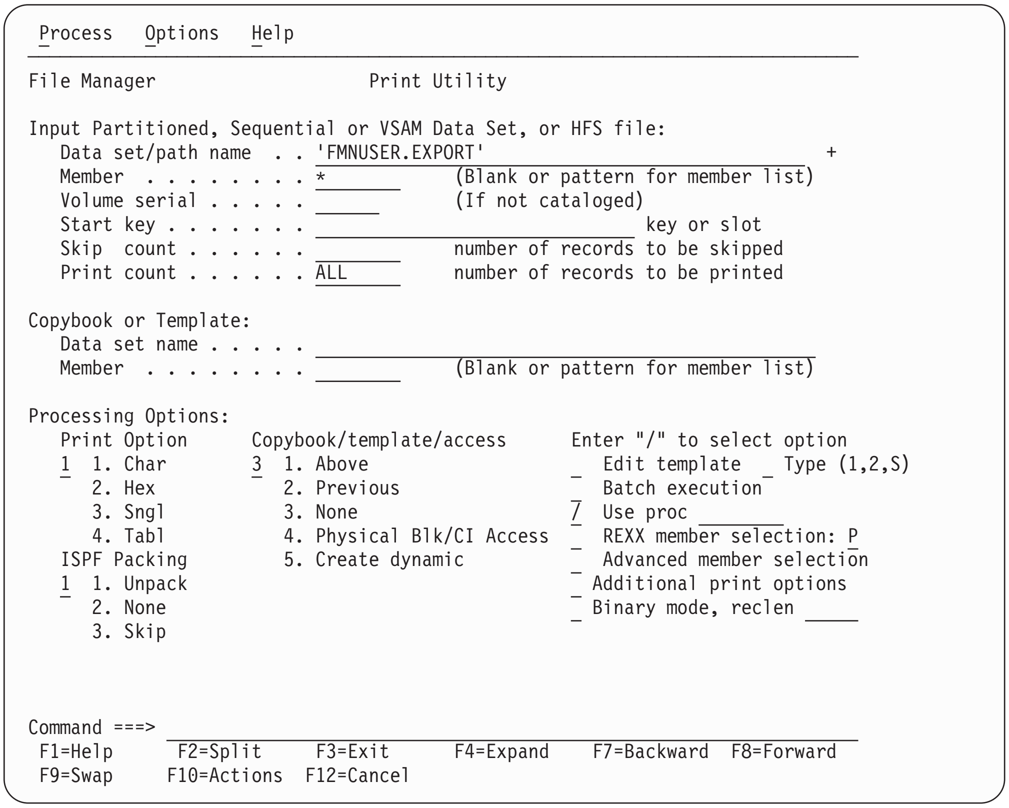
    RENAME (R)

    This is a synonym for Alter with non-VSAM data sets and it works identically. For VSAM data sets it invokes a separate VSAM Entry Rename panel.

  - Change **RECALL (R)** to **RECALL (REC)**.

- **Print Utility panel**:

  - Replace the panel shown with this updated version (showing the new **Additional print options** option):



```
 Process   Options   Help
──────────────────────────────────────────────────────────────────────────────
File Manager                    Print Utility

Input Partitioned, Sequential or VSAM Data Set, or HFS file:
   Data set/path name  . . 'FMNUSER.EXPORT'                               +
   Member  . . . . . . . . *             (Blank or pattern for member list)
   Volume serial . . . . .               (If not cataloged)
   Start key . . . . . . .                               key or slot
   Skip  count . . . . . .               number of records to be skipped
   Print count . . . . . . ALL           number of records to be printed

Copybook or Template:
   Data set name . . . . .
   Member  . . . . . . . .               (Blank or pattern for member list)

Processing Options:
   Print Option      Copybook/template/access      Enter "/" to select option
   1  1. Char        3  1. Above                   _  Edit template  _ Type (1,2,S)
      2. Hex            2. Previous                _  Batch execution
      3. Sngl           3. None                    /  Use proc ________
      4. Tabl           4. Physical Blk/CI Access  _  REXX member selection: P
   ISPF Packing         5. Create dynamic          _  Advanced member selection
   1  1. Unpack                                    _ Additional print options
      2. None                                      _ Binary mode, reclen _____
      3. Skip



Command ===> ________________________________________________________________
 F1=Help      F2=Split     F3=Exit      F4=Expand    F7=Backward  F8=Forward
 F9=Swap     F10=Actions  F12=Cancel
```

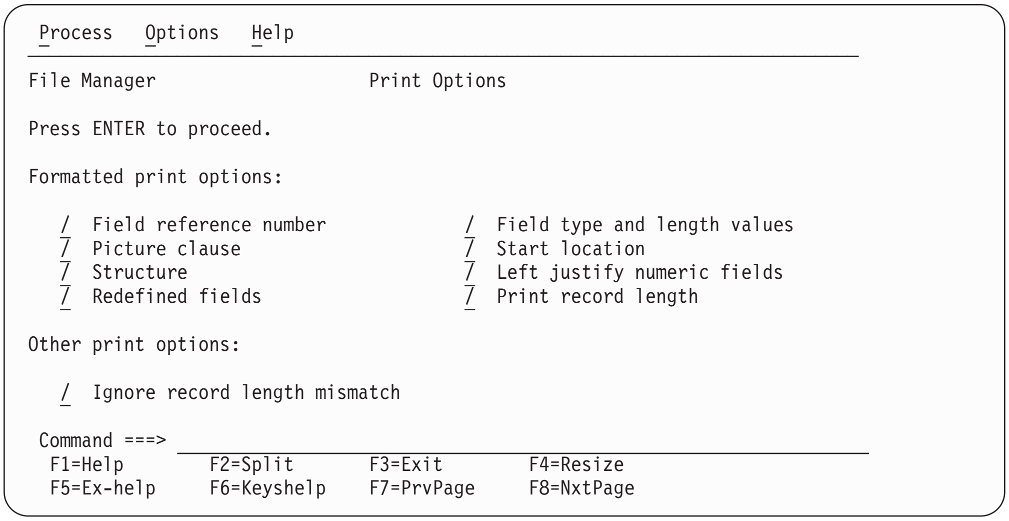
Figure 18. Print Utility panel

  - In the list of panel field descriptions, add:

Additional print options
Displays the Print Options panel allowing you to specify additional print options.

– Under ″**Child panels**″, add a reference to the (new) Print options panel.

- Add the new **Print options** panel as pictured here:



```
 Process   Options   Help
────────────────────────────────────────────────────────────────────────────────
File Manager                    Print Options

Press ENTER to proceed.

Formatted print options:

   /  Field reference number             /  Field type and length values
   /  Picture clause                     /  Start location
   /  Structure                          /  Left justify numeric fields
   /  Redefined fields                   /  Print record length

Other print options:

   /  Ignore record length mismatch

 Command ===> _________________________________________________________________
  F1=Help       F2=Split      F3=Exit       F4=Resize
  F5=Ex-help    F6=Keyshelp   F7=PrvPage    F8=NxtPage
```

Figure 19. Print options panel

Formatted print options:

Field reference number (REF)
Changes the session default to show the field reference (for example, #2) next to each field, in SNGL display or print format. In an editor session, this display can also be turned on or off temporarily, by using the REFS command.

Field type and length values (TYPE)
Changes the session default to show the field data type and length next to each field, in SNGL display or print format. In an editor session, this display can also be turned on or off temporarily, by using the TYPE command.

Picture clause (PIC)
Changes the session default to show the Picture clause when in SNGL display or print mode. Also shows the length and scale (if non zero) for binary and packed fields and the bit length for bit fields. In an editor session, this display can also be turned on or off temporarily, by using the PIC command.

Start location (SLOC)
Changes the session default to show the start location of each field, when in SNGL display or print mode. In an editor session, this display can also be turned on or off temporarily, by using the SLOC command.

Structure (STR)
Changes the session default to show the full structure of the template when using SNGL display or print mode. In an editor session, this display can also be turned on or off temporarily, by using the STR command.

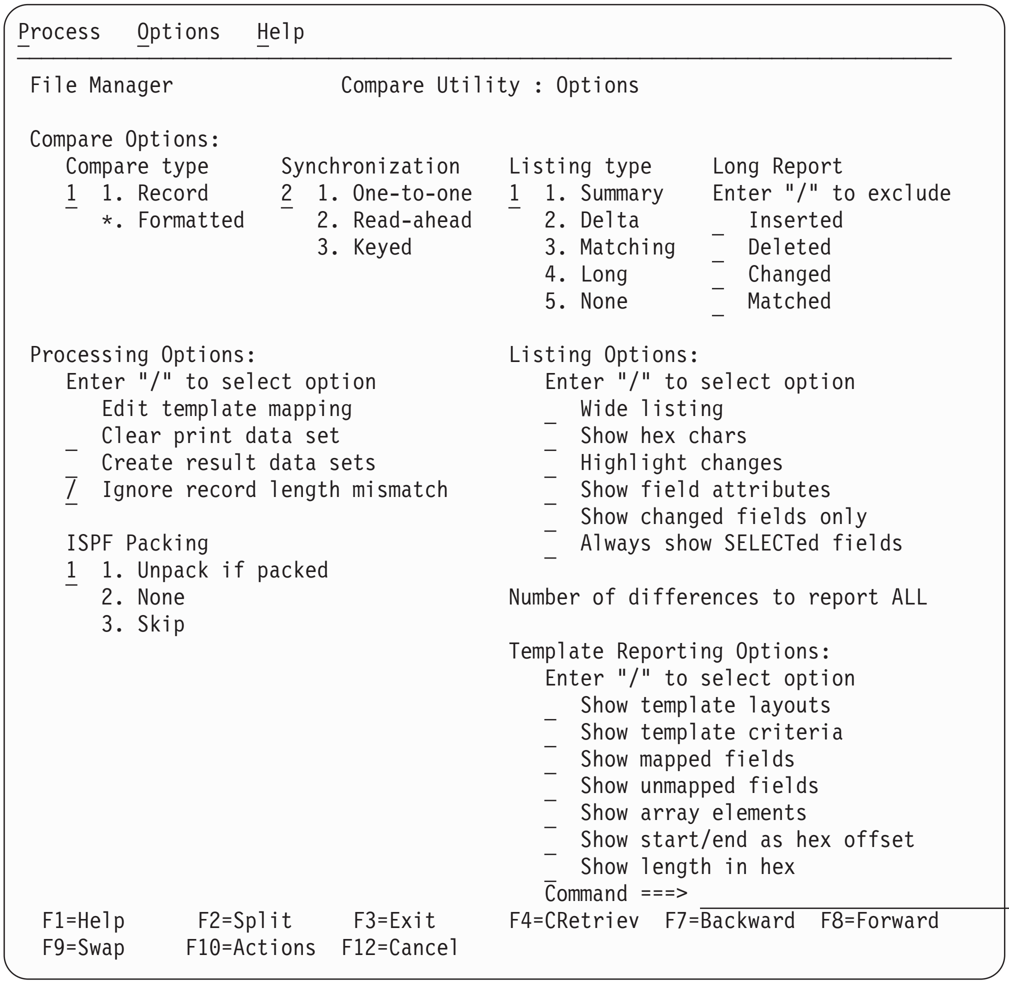
Left justify numeric fields (JUST)
Changes the session default to show numeric fields as left justified, when using SNGL display or print mode. In an editor session, this display can also be turned on or off temporarily, by using the JUST command.

Redefined fields (RDF)
Changes the session default to show the redefined information as part of the field name, when in SNGL display or print mode and display or print the redefined fields when in SNGL or TABL display or print mode. In an editor session, this display can also be turned on or off temporarily, by using the RDF command.

Print record length
Turns on or off the display of record lengths in multiple line formats (TABL,CHAR,HEX,LHEX). In an editor session, this display can also be turned on or off temporarily, by using the RECLEN command.

Ignore record length mismatch
Records failing the selection process because of length mismatch with template layouts) are considered selected for processing if they meet other selection criteria.

- **Compare Utility: Options panel**:
  - Replace the panel shown with this updated version (showing the new **Ignore record length mismatch** option):



```
Process   Options   Help
─────────────────────────────────────────────────────────────────────────────
 File Manager             Compare Utility : Options

 Compare Options:
    Compare type      Synchronization    Listing type     Long Report
    1  1. Record      2  1. One-to-one   1  1. Summary    Enter "/" to exclude
       *. Formatted      2. Read-ahead      2. Delta      _  Inserted
                         3. Keyed           3. Matching   _  Deleted
                                            4. Long       _  Changed
                                            5. None       _  Matched

 Processing Options:                     Listing Options:
    Enter "/" to select option              Enter "/" to select option
       Edit template mapping                _  Wide listing
    _  Clear print data set                 _  Show hex chars
    _  Create result data sets              _  Highlight changes
    /  Ignore record length mismatch        _  Show field attributes
                                            _  Show changed fields only
    ISPF Packing                            _  Always show SELECTed fields
    1  1. Unpack if packed
       2. None                           Number of differences to report ALL
       3. Skip
                                         Template Reporting Options:
                                            Enter "/" to select option
                                            _  Show template layouts
                                            _  Show template criteria
                                            _  Show mapped fields
                                            _  Show unmapped fields
                                            _  Show array elements
                                            _  Show start/end as hex offset
                                            _  Show length in hex
                                            Command ===> ____________________
  F1=Help      F2=Split     F3=Exit      F4=CRetriev  F7=Backward  F8=Forward
  F9=Swap     F10=Actions  F12=Cancel
```

Figure 20. Compare Utility - Options panel

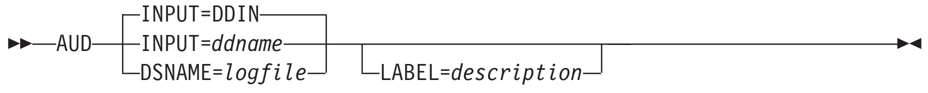
– Add this description for the new option:

Ignore record length mismatch
Enter ″/″ to ignore length mismatches with the template. If you do not select this option, records that are shorter or longer than the matching structure length in the template are not selected for processing.

Chapter 16, ″Functions″: In the section, ″File Manager functions″, subsection ″**AUD (Print audit Trail Report)**″:

- Change the syntax diagram to show:

Syntax



⋮

⋮

- Document INPUT as:

  INPUT=*ddname*
  Defines a reference to a DD or TSO ALLOC statement for the input data set or HFS file.
- Add the following comment to the DELETELOG description:

  DELETELOG is not allowed for log data sets allocated using a DD statement.

In the section, ″File Manager functions″, subsection, ″**SET (Set Processing Options)**″:

- In the description of the HEADERPG parameter, add a comment:

  For the HEADERPG=NO setting to be effective for the duration of the entire BATCH processing, it needs to be specified as the first FM control statement. The HEADERPG=NO setting also suppresses the output of the message ″FMN0062I Current SET function values″ as long as it is active.
- In the description of the DATAHDR parameter, add:

  DATAHDR setting affects only DSP and FCH processing.

In the section, ″File Manager functions″, subsection, ″**DSC (Data Set Copy)**″, under ″Usage notes″ add two new bullet point items:

- These changes are visible in the printed DSC BATCH processing report:
  - Member names are printed as specified on the CPYMBR list (when used).
  - Whenever a member name, its alias or new name (prompt value) contains unprintable characters, an additional line of output is printed below the regular output containing the hexadecimal values of the respective member names.
  - Whenever a member was located in the input library and appeared on the CPYMBR list but was not selected for processing because of the member mask or advanced member selection criteria, then it is shown in the processing report as ″Not selected″ (as opposed to ″Not found″) and be counted in the ″not copied″ category (as opposed to ″in error″).
  - For example, with these control cards:

    ```
    $$FILEM MEMBER=X'5C22',
    $$FILEM CPYMBR=(C'allocplx',
    $$FILEM         x'8289879784a222',
    $$FILEM         autotest,
    $$FILEM         X'84A282')
    ```

    One would see this output:

    ```
                                                      Member Copy Report
    Member         Newname    Alias      Status
    ----------------------------------------------------------------------
    allocplx                             Not selected
    bigpds                               Replaced
    X'8289879784A222'
    AUTOTEST                             Not found
    dsb                                  Not selected
    FMN4688I 0 member(s) copied; 1 replaced; 2 not copied; 1 in error
    ```
- **Member names containing lowercase or unprintable characters:**
  - Member names specified with the CPYMBR, MEMBER, MEMSTART, MEMEND, or MEMOUT keywords may contain lowercase or unprintable characters.

- To specify a member name containing lowercase or mixedcase characters, use the character literal form of the name surrounded by quotes and preceded with character C. For example, `C'aBc'`.
- To specify a member name containing unprintable characters, use the hexadecimal literal form of the name surrounded by quotes and preceded with character X. For example, X'81C283'. Mask characters (their hexadecimal value) may be included within the string.

In the section, ″File Manager functions″, subsection, ″**DSC (Data Set Copy)**″, under ″Performance tips″ add the following paragraph:

File Manager does not use IEBCOPY when processing members of a PDS(E) when it detects any member names containing unprintable or lowercase characters since IEBCOPY is not capable of processing such member names. This may negatively affect the performance of the DSC operation.

In the descriptions of the MEMBER, MEMSTART, MEMEND, CPYMBR, and MEMOUT keywords following the syntax diagram for the DSC function, add the following note:

Note: See ″Usage notes″ for DSC.

In the section, ″File Manager functions″, subsection, ″**DSEB (Data Set Edit Batch)**″:

- Add this extra syntax to the existing syntax diagram:

```
       ┌─IGNLEN=YES─┐
►►─────┤            ├───────────────────────────────────────────────────────►◄
       └─IGNLEN=NO──┘
```

- Add this description:

  IGNLEN
  Specifies whether or not File Manager ignores length mismatches when selecting records for processing.

  NO Do not ignore length mismatches. Records that are shorter than the matching structure length in the template are not selected for processing.

  YES Use this option to ignore length mismatches.

Make the same change for DSP (Data Set Print) and DSM (Data Set Compare).

In the section, ″External REXX Functions″, subsection, ″**WRITE**″, before the paragraph starting ″The record is written from the contents of one of the following REXX variables:″, add the following comment:

Sequential data sets specified on a WRITE statement to which no records have been written during the execution of a program are still opened and closed by File Manager. This means that:

- Newly allocated data sets with DISP=(NEW,...) are initialized to an ″empty″ state and only contain an EOF record.
- Existing data sets with data are handled as follows:
  - DISP=(OLD,...) data sets are reset to an ″empty″ state and contain only an EOF record; all previously existing data is lost.
  - DISP=(MOD,...) data sets have their data preserved unaffected.

Chapter 17, ″Messages″: Add these new messages:

FMN0075I Required HOSTNAME parameter not specified.

Explanation: A client on a system that supports IPv6 attempted to establish a connection with File Manager but did not pass the required HOSTNAME parameter.

User response: Contact your system support.

FMN0076I IPV is not at the minimum required service level.

Explanation: IPV is not at the minimum required service level to interface with your installed version of File Manager/CICS.

User response: Upgrade IPV to the required service level and then rerun File Manager.

FMN2515I FMNOPTS allocated to name.

Explanation: File Manager has detected that the FMNOPTS DD statement is allocated to name, where name is either the name of the data set allocated to FMNOPTS, or ″SYSIN″ indicating that the FMNOPTS statements have been provided instream. A listing of the options read from FMNOPTS follows.

User response: None.

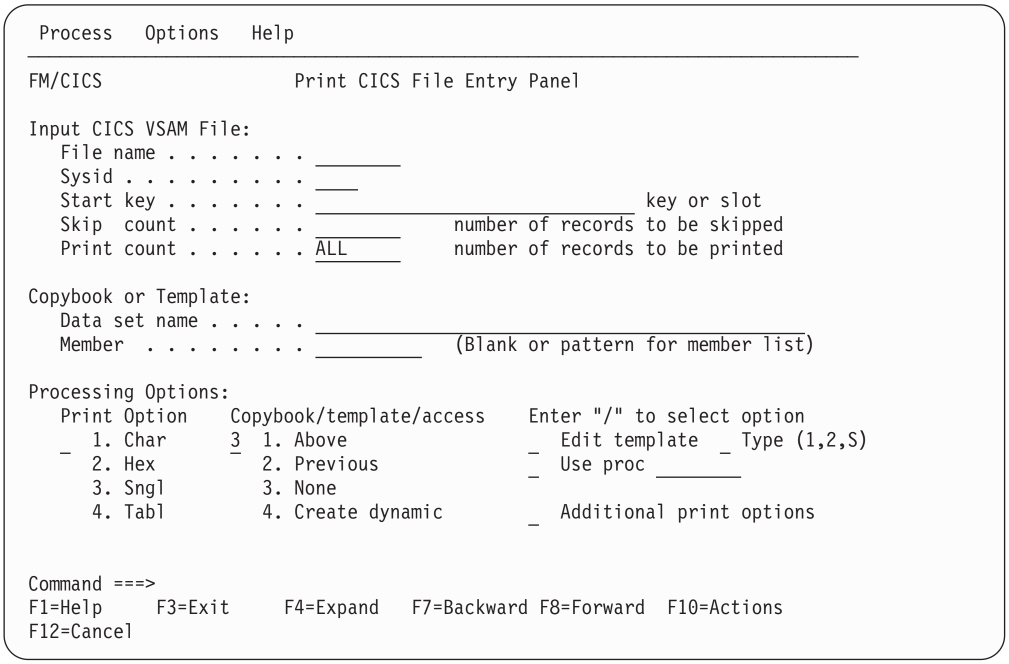
FMN2516I Error processing FMNOPTS options file.

Explanation: File Manager has encountered one or more errors while processing the options file allocated to the FMNOPTS DD statement. This message is preceded on one more error messages describing the error(s) encountered.

User response: Refer to the previous error messages related to the FMNOPTS options file. Correct the errors, and then rerun File Manager.

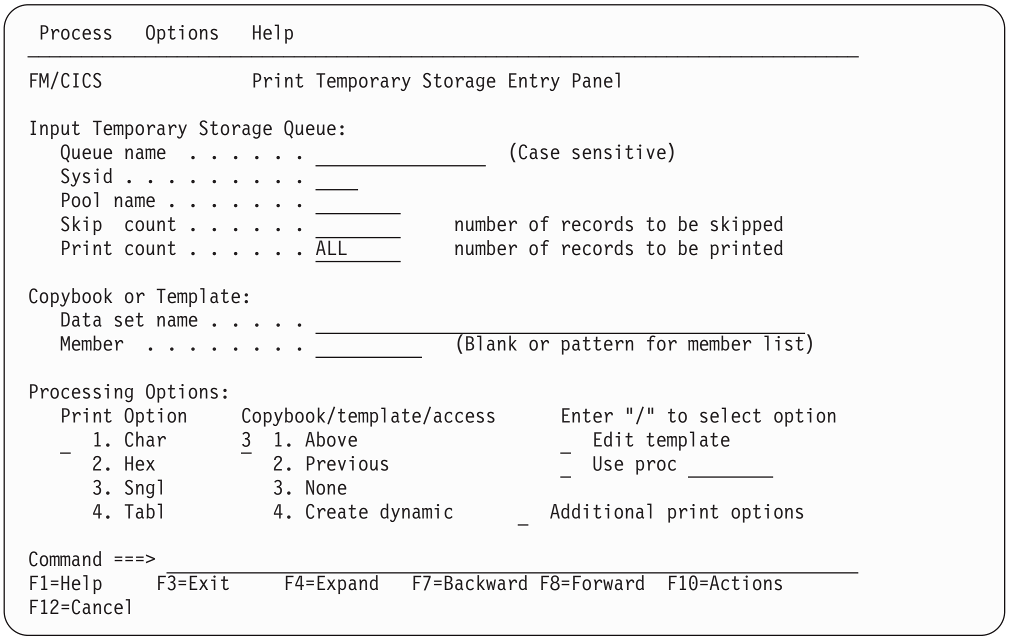
Changes to the User’s Guide and Reference for CICS

Chapter 10, ″FM/CICS panels and fields″: In the section, ″Print Entry panels″, replace the three entry panels with these (showing the new **Additional print options** option):



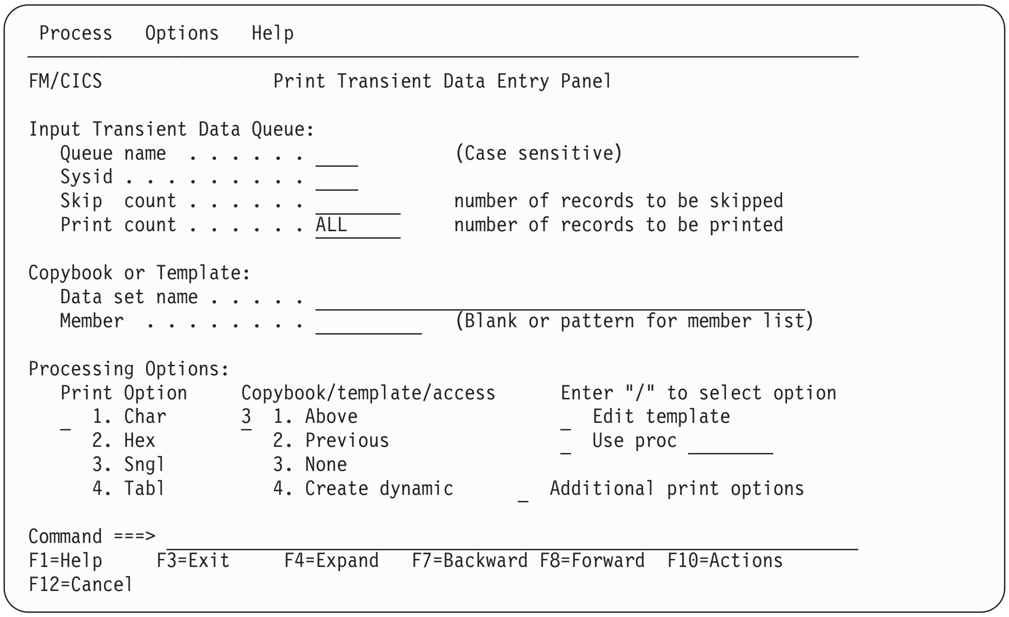
```
 Process   Options   Help
──────────────────────────────────────────────────────────────────────────────
FM/CICS                  Print CICS File Entry Panel

Input CICS VSAM File:
   File name . . . . . . . ________
   Sysid . . . . . . . . . ____
   Start key . . . . . . . ______________________________ key or slot
   Skip  count . . . . . . ________     number of records to be skipped
   Print count . . . . . . ALL_____     number of records to be printed

Copybook or Template:
   Data set name . . . . . ______________________________________________
   Member  . . . . . . . . __________   (Blank or pattern for member list)

Processing Options:
   Print Option    Copybook/template/access    Enter "/" to select option
   _  1. Char     3  1. Above                 _  Edit template  _ Type (1,2,S)
      2. Hex         2. Previous              _  Use proc ________
      3. Sngl        3. None
      4. Tabl        4. Create dynamic        _  Additional print options


Command ===>
F1=Help     F3=Exit     F4=Expand   F7=Backward F8=Forward  F10=Actions
F12=Cancel
```

Figure 21. Print CICS File Entry panel



```
 Process   Options   Help
------------------------------------------------------------------------------
FM/CICS             Print Temporary Storage Entry Panel

Input Temporary Storage Queue:
   Queue name  . . . . . . ________________  (Case sensitive)
   Sysid . . . . . . . . . ____
   Pool name . . . . . . . ________
   Skip  count . . . . . . ________     number of records to be skipped
   Print count . . . . . . ALL_____     number of records to be printed

Copybook or Template:
   Data set name . . . . . ______________________________________________
   Member  . . . . . . . . __________   (Blank or pattern for member list)

Processing Options:
   Print Option     Copybook/template/access      Enter "/" to select option
   _  1. Char      3  1. Above                    _  Edit template
      2. Hex          2. Previous                 _  Use proc ________
      3. Sngl         3. None
      4. Tabl         4. Create dynamic       _  Additional print options

Command ===> _________________________________________________________________
F1=Help     F3=Exit     F4=Expand   F7=Backward F8=Forward  F10=Actions
F12=Cancel
```

Figure 22. Print Temporary Storage Entry panel



```
 Process   Options   Help
------------------------------------------------------------------------------
FM/CICS               Print Transient Data Entry Panel

Input Transient Data Queue:
   Queue name  . . . . . . ____         (Case sensitive)
   Sysid . . . . . . . . . ____
   Skip  count . . . . . . ________     number of records to be skipped
   Print count . . . . . . ALL_____     number of records to be printed

Copybook or Template:
   Data set name . . . . . ______________________________________________
   Member  . . . . . . . . __________   (Blank or pattern for member list)

Processing Options:
   Print Option     Copybook/template/access      Enter "/" to select option
   _  1. Char      3  1. Above                    _  Edit template
      2. Hex          2. Previous                 _  Use proc ________
      3. Sngl         3. None
      4. Tabl         4. Create dynamic       _  Additional print options

Command ===> _________________________________________________________________
F1=Help     F3=Exit     F4=Expand   F7=Backward F8=Forward  F10=Actions
F12=Cancel
```

Figure 23. Print Transient Data Entry panel

In the list of panel field descriptions, add:

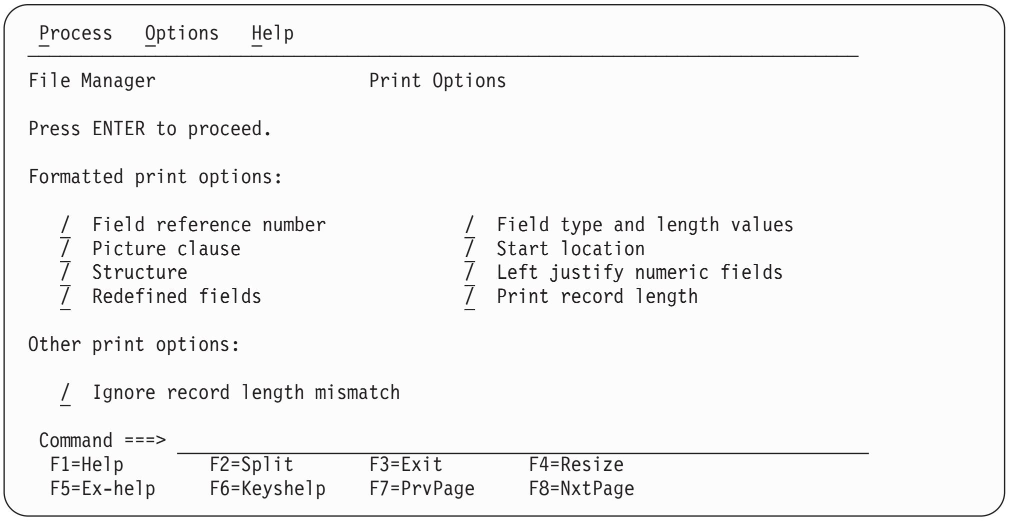
Additional print options
Displays the Print Options panel allowing you to specify additional print options.

Under ″**Child panels**″, add a reference to the (new) Print options panel.

Add this new panel:

Print Options panel:

Panel and field definitions:



```
 Process   Options   Help
─────────────────────────────────────────────────────────────────────────────
File Manager                    Print Options

Press ENTER to proceed.

Formatted print options:

   /  Field reference number            /  Field type and length values
   /  Picture clause                    /  Start location
   /  Structure                         /  Left justify numeric fields
   /  Redefined fields                  /  Print record length

Other print options:

   /  Ignore record length mismatch

 Command ===> _________________________________________________________________
  F1=Help        F2=Split       F3=Exit        F4=Resize
  F5=Ex-help     F6=Keyshelp    F7=PrvPage     F8=NxtPage
```

Figure 24. Print options panel

Formatted print options:

Field reference number (REF)
Changes the session default to show the field reference (for example, #2) next to each field, in SNGL display or print format. In an editor session, this display can also be turned on or off temporarily, by using the REFS command.

Field type and length values (TYPE)
Changes the session default to show the field data type and length next to each field, in SNGL display or print format. In an editor session, this display can also be turned on or off temporarily, by using the TYPE command.

Picture clause (PIC)
Changes the session default to show the Picture clause when in SNGL display or print mode. Also shows the length and scale (if non zero) for binary and packed fields and the bit length for bit fields. In an editor session, this display can also be turned on or off temporarily, by using the PIC command.

Start location (SLOC)
Changes the session default to show the start location of each field, when in SNGL display or print mode. In an editor session, this display can also be turned on or off temporarily, by using the SLOC command.

Structure (STR)
Changes the session default to show the full structure of the template when using SNGL display or print mode. In an editor session, this display can also be turned on or off temporarily, by using the STR command.

Left justify numeric fields (JUST)
Changes the session default to show numeric fields as left justified,

when using SNGL display or print mode. In an editor session, this display can also be turned on or off temporarily, by using the JUST command.

Redefined fields (RDF)

Changes the session default to show the redefined information as part of the field name, when in SNGL display or print mode and display or print the redefined fields when in SNGL or TABL display or print mode. In an editor session, this display can also be turned on or off temporarily, by using the RDF command.

Print record length

Turns on or off the display of record lengths in multiple line formats (TABL,CHAR,HEX,LHEX). In an editor session, this display can also be turned on or off temporarily, by using the RECLEN command.

Ignore record length mismatch

Records failing the selection process because of length mismatch with template layouts) are considered selected for processing if they meet other selection criteria.

Parent panels:

- Print Entry panels.

Child panels: None.

Part 2. General documentation changes

This section describes enhancements and updates in the documentation for File Manager for z/OS Version 8 Release 1. These changes are not associated with individual APAR or PTF numbers, as they do not require the application of any code updates.

The changes are grouped by manual and listed within each section in reverse date order. That is, the most recent documentation change appears at the beginning of each manual section.

Customization Guide (SC19-1238-01)

Change #1 July 2008

Appendix B. ″File Manager/DB2 Options″

LIST option:

Change the description of the LIST option to read:

LIST controls which items appear in the subsystem selection list. FM/DB2 scans the sub-system vector table of the operating system when preparing the subsystem selection list. All DB2 systems found are considered **eligible** for inclusion on the subsystem selection list. If a DB2 system is not defined to z/OS it **will not appear** in the sub-system vector table and will therefore **not appear** in the FM/DB2 sub-system selection list, even if there is an FMN2SSDM macro entry for it.

Change the description of the ALL parameter to read:

All eligible (see above) subsystems and group identifiers defined in the FMN2SSDM macro appear in the subsystem selection list, with the exception of any entries specified DISPLAY=HIDDEN in the FMN2SSDM macro. The default is **LIST=ALL**

At the end of the section add the following explanatory information:

The following information is provided to assist you in understanding why items appear on the FM/DB2 subsystem selection list. It is not necessary to change the entries in member IEFSSNxx.

DB2 systems are defined to z/OS using entries in member IEFSSNxx in SYS1.PARMLIB. Here are three examples:

```
SUBSYS SUBNAME(DFB2) INITRTN(DSN3INI) INITPARM('DSN3EPX,-DFB2,S')      A
SUBSYS SUBNAME(DFA2) INITRTN(DSN3INI) INITPARM('DSN3EPX,-DFA2,S,DFA2') B
SUBSYS SUBNAME(DFS2) INITRTN(DSN3INI) INITPARM('DSN3EPX,-DFS2,S,DFG2') C
```

A defines a DB2 subsystem DFB2 that is not part of a DB2 data sharing group.

B defines a DB2 subsystem DFA2 that is also part of a DB2 data sharing group. The data sharing group is also called DFA2.

C defines a DB2 subsystem DFS2 that is also part of a DB2 data sharing group. The data sharing group name is DFG2.

A generates a single item on the FM/DB2 subsystem selection list. The corresponding FMN2SSDM macro entry should specify:

```
FMN2SSDM  SSID=DFB2,TYPE=SUBSYS, ...
```

B and **C** generate two items each on the FM/DB2 subsystem selection list. For **B** the corresponding FMN2SSDM macro entries are:

```
FMN2SSDM  SSID=DFA2,TYPE=SUBSYS, ...
FMN2SSDM  SSID=DFA2,TYPE=GROUP, ...
```

For C the required FMN2SSDM macro entries are:

```
FMN2SSDM  SSID=DFS2,TYPE=SUBSYS, ...
FMN2SSDM  SSID=DFG2,TYPE=GROUP, ...
```

Note that when DB2 system DFA2 is active, DB2 group DFA2 is also active. Similarly when DB2 system DFS2 is active, DB2 group DFG2 is also active. This occurs regardless of whether DB2 data sharing groups DFA2 and DFG2 has been completed implemented or not. From the perspective of z/Os, definitions B and C in IEFSSNxx are all that is required to define the DB2 data sharing groups DFA2 and DFG2.

In the situation where a DB2 subsystem and DB2 group have the same name (for example, B), you can hide one of the entries using DISPLAY=HIDDEN on the corresponding FMN2SSDM macro definition. You can also use the SUBSYS or GROUP options to show only subsystems or groups, respectively, although the latter method applies to every subsystem or group.

User’s Guide and Reference (SC19-1239-00)

Change #3: May 2009

Chapter 6, ″Managing data sets″, section ″Comparing data sets″, subsection ″Record synchronization″

- Change the sentence ″Any records that do not match are regarded as paired insertions and deletions.″

  to:

  ″Non-matching records are regarded as changed records.″

Change #2: December 2008

Chapter 16, ″Functions″, section ″External REXX functions″

- In the list of functions at the beginning of the section:
  - In the description for RSTR_OUT, add the sentence:

    ″(Can be used in FASTREXX condition expressions.)″
  - In the description for SAVE_OUT, add the sentence:

    ″(Can be used in FASTREXX condition expressions.)″
- In the subsection, ″RSTR_OUT″, after the syntax diagram add the sentence:

  ″(Can be used in FASTREXX condition expressions.)″
- In the subsection, ″SAVE_OUT″, after the syntax diagram add the sentence:

  ″(Can be used in FASTREXX condition expressions.)″

Change #1 October 2007

Chapter 16, ″Functions″

Section, ″DSI (Disk Info)″

Change the heading of this section from ″DSI (Disk Info)″ to ″DSI (Data Set Information)″.

User’s Guide and Reference for DB2 Data (SC19-1240-00)

There are no general documentation changes.

User’s Guide and Reference for IMS Data (SC19-1241-00)

There are no general documentation changes.

User’s Guide and Reference for CICS (SC19-1242-00)

There are no general documentation changes.

Part 3. Appendixes

Index

Printed in USA

Spine information:

File Manager for z/OS V8R1

Addenda to V8R1 User's Guides and Customization Guide