

1.1 IXLCSP API

IXLCSP is updated to support structure sizing at a specified CFLevel. In conjunction with that support, it will provide a new query service that returns as output a list of CFLevels for which sizing is supported by the invoking system.

The following sections describe associated publication updates.

1.1.1 Sysplex Services Reference

Affected publication: z/OS MVS Sysplex Services Reference SA38-0658.

Changes in this section refer to Chapter "IXLCSP – XES Structure Computation Service"

1.1.1.1 Section "Understanding IXLCSP Version Support"

The IXLCSP macro now supports versions 1 – 4.

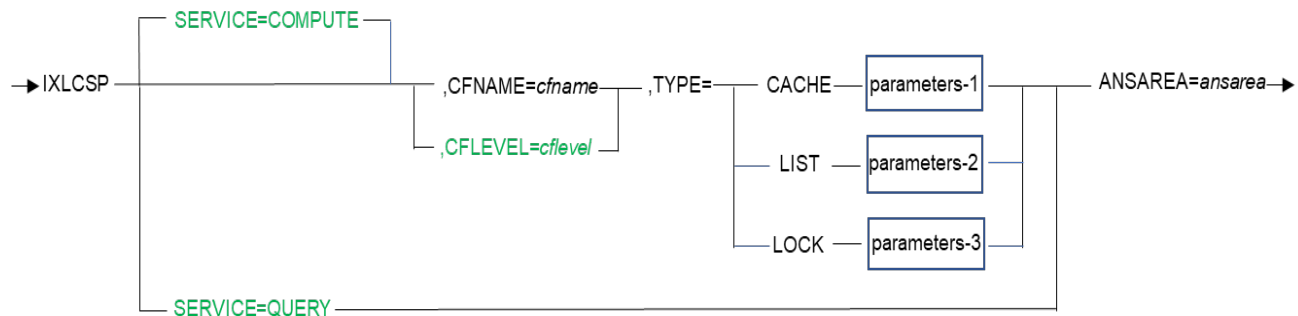
Add the following bullet after the one describing version 3 keywords:

- The following keywords are supported by version 4 and subsequent versions of the IXLCSP macro:

CFLEVEL
SERVICE

1.1.1.2 Section "Syntax"

In the box marked "main diagram", replace the first line of the syntax diagram with the following:



1.1.1.3 Section "Parameter Descriptions"

Between the ASYNCDUPLEX and CFNAME parameter descriptions, add the following:

CFLEVEL=*cflevel*

Use this input parameter to specify the CFLEVEL for which computations are to be performed. The system can support calculations for any CFLEVEL that is installed in a CF accessible to the invoking system and for CFLEVEL(s) that are emulated in software. The set of emulated CFLEVELs varies by z/OS release and maintenance level. Use IXLCSP SERVICE=QUERY to obtain a list of CFLEVELs supported by the invoking system.

An input value of 0 requests calculation at the highest CFLEVEL available to the invoking system. On successful completion, CSPA_ActualCfLevel will contain the CFLEVEL for which the calculation was performed.

Test the QuReqRflxlcspCFLevel bit in the QuReqFeatures string returned by an IXCQUERY REQINFO=FEATURES invocation to determine whether the current system supports the CFLEVEL keyword. Macro IXCYQUAA defines the QuReqRflxlcspCFLevel flag and QuReqFeatures string.

Between the SCMMAXSIZE and STRSIZE parameter descriptions, add the following:

SERVICE=COMPUTE
SERVICE=QUERY

Use this input parameter to specify the service requested.

COMPUTE

Perform a structure sizing calculation.

QUERY

Report the set of CFLEVELs for which the invoking system can perform structure sizing calculations. Answer area mapping CSPA_CFLevelInfo describes the set of supported CFLEVELs.

Test the QuReqRflxlcspCFLevel bit in the QuReqFeatures string returned by an IXCQUERY REQINFO=FEATURES invocation to determine whether the current system supports SERVICE=QUERY. Macro IXCYQUAA defines the QuReqRflxlcspCFLevel flag and QuReqFeatures string.

1.1.1.4 Section "Return and Reason Codes"

In the table entitled "Return and Reason Codes for the IXLCSP Macro", add the following table entries:

| Hexadecimal Return Code | Hexadecimal Reason Code | Equate Symbol Meaning and Action |
|-------------------------|-------------------------|---|
| 8 | xxxx0824 | Equate Symbol: IxIRsnCodeWrongStrType Meaning and Action: Incorrect structure type specified. |
| 8 | xxxx0886 | Equate Symbol: IxIRsnCodeBadRequest Meaning and Action: Incorrect SERVICE or REQUEST value specified. |
| 8 | xxxx08BE | Equate Symbol: IxIRsnCodeUnsupportedCFLEVEL Meaning and Action: The request specified a CFLEVEL value not available to this system, because there is no connected CF that provides the requested level and the z/OS system from which the request was submitted does not support emulation of the requested level in software. Answer area mapping CSPA_CFLevelInfo describes the set of supported CFLEVELs. |

1.1.2 Sysplex Services Guide

Affected publication: z/OS MVS Sysplex Services Guide SA23-1400.

1.1.2.1 Using the IXLCSP Service

Changes in this section refer to Chapter "Connection Services", section "Connecting to a Coupling Facility Structure", subsection "Using the IXLCSP Service to Determine Structure Size or Attributes"

Change the paragraph beginning "IXLCSP directs a request ..." to read as follows:

IXLCSP can compute a structure's size or object counts either by directing a computation request to a coupling facility specified by name or CFLEVEL, or by emulating in software the computations that would be performed by a coupling facility at a specified CFLEVEL. The coupling facility or software emulation will perform the requested calculation just as if it were actually allocating the structure. However, no structure allocation occurs and the contents of the target coupling facility, if applicable, are unchanged at the conclusion of the IXLCSP calculation.

The following considerations apply to the calculation:

- The target coupling facility, if applicable, must be described in the CFRM active policy and accessible from the system invoking IXLCSP.
- The calculations performed by the coupling facility are idealized in the sense that they do not account for any constraints or conditions (such as storage shortages) that might prevent a structure from actually being allocated in the coupling facility.
- Calculations performed by a coupling facility are appropriate to the CFLEVEL of that coupling facility. Calculations for different CFLEVELs will, in all likelihood, return different answers.
- Storage-class memory calculations require a coupling facility with a CFLEVEL greater than or equal to 19.

1.1.2.2 Using the IXCQUERY macro

Changes in this section refer to Chapter "Using the Cross-System Coupling Facility(XCF)", section "Obtaining XCF Information", subsection "Using the IXCQUERY Macro" / "Information returned inline to IXCQUERY".

After QuReqRflxcNoteResiliency, add:

QuReqRflxcspCFLevel

The IXLCSP CFLEVEL keyword and the SERVICE=QUERY function are supported on this system.

1.1.3 Data Areas

Affected publication: z/OS MVS Data Areas Volume 3 (ITK – RQE) GA32-0937

1.1.3.1 IXLYCSPA

The following table describes changes to the existing IXLYCSPA (IXLCSP Answer Area) mapping.

| Offset Dec | Offset Hex | Type | Len | Name (Dim) | Description |
|------------|------------|-----------|-----|--------------------|---|
| 0 | (0) | STRUCTURE | 0 | CSPA | IXLCSP answer area |
| 0 | (0) | CHARACTER | 32 | CSPA_HEADER(0) | Header information |
| 0 | (0) | SIGNED | 2 | CSPA_VERSION | IXLYCSPA version number |
| 2 | (2) | SIGNED | 2 | CSPA_LENGTH | Length of the answer area |
| 4 | (4) | SIGNED | 2 | CSPA_OFFSET | Offset from the beginning of the structure (CSPA) to the answer area data (CSPA_Data) |
| 6 | (6) | CHARACTER | 26 | | Reserved |
| 32 | (20) | CHARACTER | 224 | CSPA_DATA(0) | Data returned by IXLCSP |
| 32 | (20) | CHARACTER | 64 | CSPA_COMMONDATA(0) | Data common to all IXLCSP SERVICE=COMPUTE requests |
| 32 | (20) | SIGNED | 4 | CSPA_STRSIZE | Structure size in 4K blocks |

| | | | | | |
|-----|------|--------|-----|--------------------|---|
| ... | ... | ... | ... | ... | ... |
| 54 | (36) | SIGNED | 2 | | Reserved |
| 56 | (38) | SIGNED | 4 | CSPA_ACTUALCFLEVEL | CFLEVEL for which calculation was performed. Valid if not zero. |
| ... | ... | ... | ... | ... | ... |

The following table describes the new CSPA_CFLevelInfo mapping, containing information about CFLevels for which computations are supported on the system invoking IXL CSP.

| Offset Dec | Offset Hex | Type | Len | Name (Dim) | Description |
|--|------------|-----------|-----|-----------------------|--|
| Output from IXL CSP SERVICE=QUERY Located by adding CSPA_Offset to the address of the CSPA. | | | | | |
| 0 | (0) | STRUCTURE | 0 | CSPA_CFLEVELINFO | IXL CSP SERVICE=QUERY output |
| 0 | (0) | SIGNED | 4 | CSPA_CFLEVELCOUNT | Number of valid entries in CSPA_CFLevelArray |
| 4 | (4) | STRUCTURE | 8 | CSPA_CFLEVELARRAY | Array of CFLEVEL information |
| 4 | (4) | SIGNED | 4 | CSPA_CFLEVELVALUE | CFLEVEL value |
| 8 | (8) | BITSTRING | 1 | CSPA_CFLEVELFLAGS | CFLEVEL flags |
| | | 1... .. | | CSPA_CFLEVELCONNECTED | On => A CF with this CFLEVEL is connected to the system initiating the IXL CSP request. In processing a sizing request specifying this CFLEVEL, the system may select any connected CF that supports it. |
| | | .1... .. | | CSPA_CFLEVELEMULATED | On => The system has support for emulating this CFLEVEL in software. |
| 9 | (9) | CHARACTER | 3 | | Reserved |

1.1.3.2 IXLYCON

The following table identifies new constants defined by IXLYCON.

| Offset Dec | Offset Hex | Type | Len | Name (Dim) | Description |
|---|------------|-----------|-----|------------------------|---|
| 0 | (0) | STRUCTURE | 0 | | |
| 0 | (0) | X'0' | 0 | IXLRETCODEOK | "0" |
| ... | ... | ... | ... | ... | ... |
| Reason Codes - IXLRETCODEPARMERROR (Note that the reason codes are of the form "xxxxYYYY" where "xxxx" is used to contain internal diagnostic information) | | | | | |
| 0 | (0) | BITSTRING | 0 | IXLRSNCODEBADPARMLIST | "X'00000801'" Parameter list could not be accessed |
| ... | ... | ... | ... | ... | ... |
| 0 | (0) | BITSTRING | 0 | IXLRSNCODEMINIMUMCOUNT | "X'000008BB'" Computation would result in counts smaller than the minimum entry count or the minimum element count. Used only when computing with |

| | | | | | |
|--|-----|-----------|-----|------------------------------|---|
| | | | | | ScmMaxSize>0 (CFLEVEL 19) |
| '000008BC'X - '000008BD'X - reserved for higher release - do not use | | | | | |
| 0 | (0) | BITSTRING | 0 | IXLRSNCODEUNSUPPORTEDCFLEVEL | "X'000008BE'" Request specified a CFLEVEL value not supported by the invoked service for the current configuration or release of z/OS |
| ... | ... | ... | ... | ... | ... |

1.2 IXLMG API

IXLMG is updated to support the new CHAINTYPE keyword and modify-form invocation.

The following sections describe associated publication updates.

1.2.1 Sysplex Services Reference

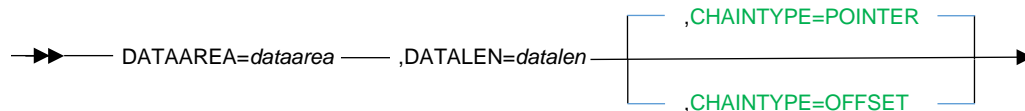
Affected publication: z/OS MVS Sysplex Services Reference SA38-0658.

Changes in this section refer to Chapter "IXLMG – Coupling Facility Measurement"

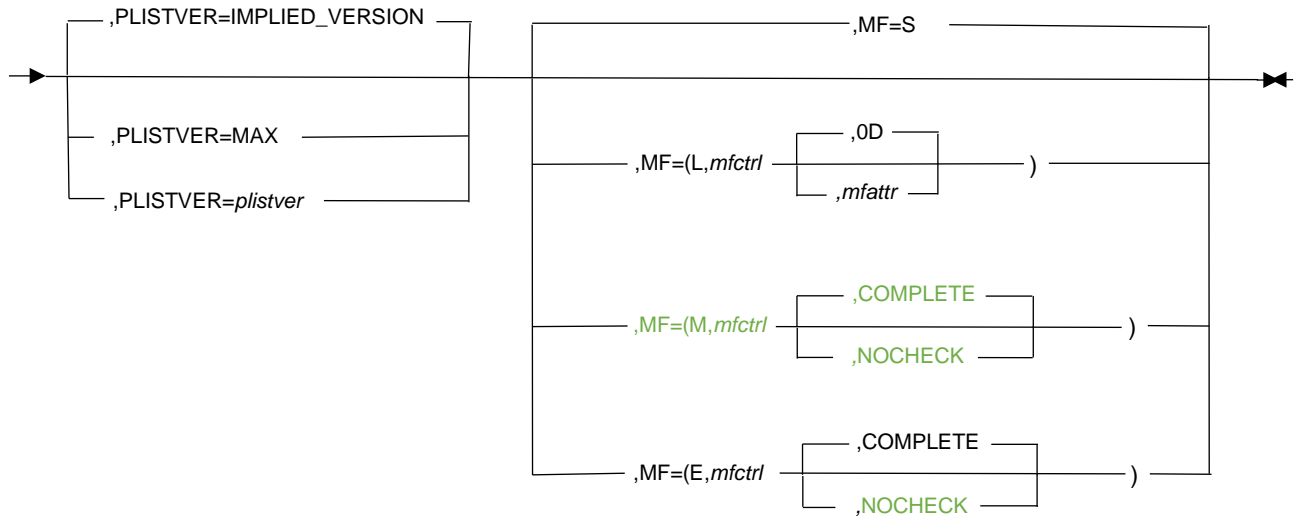
1.2.1.1 Section "Syntax Diagram"

In the box labeled "main diagram":

Replace the first line with the following:



Replace the bottom section with the following:



1.2.1.2 Section "Parameter Descriptions"

Insert the following text between the descriptions of the CFNAME and COCLASSB parameters:

,CHAINTYPE=POINTER **,CHAINTYPE=OFFSET**

Use this input parameter to specify how the entries in the IXLYAMDA data area should be chained. This selection applies to all chain fields in the data area, including:

- IXLYAMDAREA_CFEnt@
- Chain fields within entry types that link to the next entry of the same type. These fields are mapped generically by IXLYAMDHD_Next.
- Chain fields within entry types that link to entries of another type, such as IXLYAMDCF_SL@ or IXLYAMDCF1_RFAddr.

Test the QuReqRflxImgChainType bit in the QuReqFeatures string returned by an IXCQUERY REQINFO=FEATURES invocation to determine whether the current system supports the CHAINTYPE keyword. Macro IXCYQUAA defines the QuReqRflxImgChainType flag and QuReqFeatures string.

DEFAULT: POINTER

POINTER

Data area entries are to be chained by pointers. Each chain field contains an address within the area identified by the DataArea key.

OFFSET

Data area entries are to be chained by offsets. Each chain field contains an offset from the start of the area identified by the DataArea key. The address of the chained entry can be obtained by adding the offset to the address of the data area.

Replace the description of the MF (macro form) keyword with the following:

,MF=S
,MF=(L,mfctrl)
,MF=(L,mfctrl,mfattr)
,MF=(L,mfctrl,0D)
,MF=(M,mfctrl)
,MF=(M,mfctrl,COMPLETE)
,MF=(M,mfctrl,NOCHECK)
,MF=(E,mfctrl)
,MF=(E,mfctrl,COMPLETE)
,MF=(E,mfctrl,NOCHECK)

Use MF=S to specify the standard form of the macro, which builds an in-line parameter list and generates the macro invocation to transfer control to the service.

Use MF=L to specify the list form of the macro. Use the list form together with the execute form of the macro for applications that require reentrant code. The list form defines an area of storage that the execute form uses to store the parameters. Only the PLISTVER parameter can be coded with the list form of the macro.

Use MF=M to specify the modify form of the macro. Use the modify form to generate code to put the parameters into the parameter list.

Use MF=E to specify the execute form of the macro. Use the execute form together with the list form of the macro for applications that require reentrant code. The execute form stores the parameters into the storage area defined by the list form, and generates the macro invocation to transfer control to the service.

,mfctrl

Use this output parameter to specify a storage area to contain the parameters.

To Code: Specify the RS-type name or address (using a register from 2 to 12) of the parameter list.

,mfattr

Use this input parameter to specify the name of a 1- to 60-character string that can contain any value that is valid on an assembler DS pseudo-op. You can use this parameter to force boundary alignment of the parameter list. If you do not code mfattr, the system provides a value of 0D, which forces the parameter list to a doubleword boundary.

,COMPLETE

,NOCHECK

Use this input parameter to specify the degree of macro parameter syntax checking the system is to do.

COMPLETE

Use this parameter to require that the system check for required parameters and supply defaults for omitted optional parameters.

NOCHECK

Use this parameter to specify that the system is not to check for required parameters nor to supply defaults for omitted optional parameters.

Note: In the macro expansion you might see some defaults for optional parameters that are not documented here. The ones that are not documented do not have any effect on the macro. For example, if SMILE=var were an optional parameter and the default is SMILE=NO_SMILE then it would not be documented. However, if the default was SMILE=-), then it would be documented because a value would be the default.

1.2.2 Sysplex Services Guide

Affected publication: z/OS MVS Sysplex Services Guide SA23-1400.

Changes in this section refer to Chapter "Using the Cross-System Coupling Facility(XCF)", section "Obtaining XCF Information", subsection "Using the IXCQUERY Macro" / "Information returned inline to IXCQUERY".

After QuReqRflxlcspCFLevel (see section 1.1.2.2), add:

QuReqRflxlmChainType

The IXLGM CHAINTYPE keyword is supported on this system.

1.2.3 Data Areas

Affected publication: z/OS MVS Data Areas Volume 3 (ITK – RQE) GA32-0937

1.2.3.1 IXLYAMDA

The following tables describe changes to the existing IXLYAMDA (IXLMG Answer Area) mapping.

Structure IXLYAMDAREA

| Offset Dec | Offset Hex | Type | Len | Name (Dim) | Description |
|------------|------------|-----------|-----|-------------------------|--|
| 0 | (0) | STRUCTURE | 0 | IXLYAMDAREA | Data area returned to caller |
| 0 | (0) | SIGNED | 4 | IXLYAMDAREA_LENGTH | Length of IXLYAMDAREA header mapping |
| 4 | (4) | ADDRESS | 4 | IXLYAMDAREA_CFENT@ | Address of first CF entry. A value of zero means that no CF entries were provided. |
| 4 | (4) | SIGNED | 4 | IXLYAMDAREA_CFEnt_Off | Offset of first CF entry. A value of zero means that no CF entries were provided. |
| 8 | (8) | SIGNED | 4 | IXLYAMDAREA_TLen | Total length of output data area needed to contain all the requested information. This length includes the area for the records that WERE returned on this call. |
| 12 | (C) | SIGNED | 4 | IXLYAMDAREA_#Ent | Total number of entries of all kinds (not including the header) |
| 16 | (10) | BITSTRING | 1 | IXLYAMDAREA_Version | Version number - Maximum CFLEVEL supported by MVS on system where IXLMG was invoked. Can be used to determine if specific fields have valid information. See notes in prolog for more information. |
| 17 | (11) | BITSTRING | 1 | IXLYAMDAREA_Flags | Request flags |
| | | 1... .. | | IXLYAMDAREA_ChainOffset | When on, all chain fields contain offsets rather than addresses |
| 18 | (12) | CHARACTER | 2 | | Unused |

Structure IXLYAMDHD

| Offset Dec | Offset Hex | Type | Len | Name (Dim) | Description |
|------------|------------|-----------|-----|--------------------|--|
| 0 | (0) | STRUCTURE | 0 | IXLYAMDHD | Common header mapping for IXLYAMDA entries |
| 0 | (0) | BITSTRING | 1 | IXLYAMDHD_TYPE | Type of entry |
| 1 | (1) | CHARACTER | 3 | | Unused |
| 4 | (4) | SIGNED | 4 | IXLYAMDHD_Length | Length of entry |
| 8 | (8) | ADDRESS | 4 | IXLYAMDHD_Next | Address of next entry |
| 8 | (8) | SIGNED | 4 | IXLYAMDHD_Next_Off | Offset of next entry |

The following fields are doubly-defined in the same manner as IXLYAMDAREA_CFENT@ / IXLYAMDAREA_CFENT_Off and IXLYAMDHD_Next / IXLYAMDHD_Next_Off. The remaining tables are not reproduced for the sake of simplicity.

- IXLYAMDCF_CFNext / IXLYAMDCF_CFNext_Off
- IXLYAMDCF_SL@ / IXLYAMDCF_SL_Off
- IXLYAMDCF_STR@ / IXLYAMDCF_STR_Off
- IXLYAMDCF_SC@ / IXLYAMDCF_SC_Off
- IXLYAMDCF_MI@ / IXLYAMDCF_MI_Off
- IXLYAMDCF1_RFAddr / IXLYAMDCF1_RF_Off
- IXLYAMDCF1_CPAAddr / IXLYAMDCF1_CP_Off
- IXLYAMDSLL_SLNext / IXLYAMDSLL_SLNext_Off
- IXLYAMDSLC_SLNext / IXLYAMDSLC_SLNext_Off
- IXLYAMDCFMI_INFO@ / IXLYAMDCFMI_INFO_Off
- IXLYAMDCFRF_RFNext / IXLYAMDCFRF_RFNext_Off
- IXLYAMDCFCP_PathInfo@ / IXLYAMDCFCP_PathInfo_Off
- IXLYAMDSTRL_STRNext / IXLYAMDSTRL_STRNext_Off
- IxlyamdStrL1_SscmAddr / IxlyamdStrL1_Sscm_Off
- Ixlyamdstrl1_SSCCAddr / Ixlyamdstrl1_SSCC_Off
- IxlyamdStrL2_ADupAddr / IxlyamdStrL2_ADup_Off
- IXLYAMDSTRC_STRNext / IXLYAMDSTRC_STRNext_Off
- IXLYAMDSTRC_SCSC@ / IXLYAMDSTRC_SCSC_Off
- IXLYAMDSTRC_SCOC@ / IXLYAMDSTRC_SCOC_Off
- Ixlyamdstrc1_SSCCAddr / Ixlyamdstrc1_SSCC_Off
- IXLYAMDSCOCSTATS@ / IXLYAMDSCOCSTATS_Off
- IXLYAMDSC_ScNext / IXLYAMDSC_ScNext_Off

1.3 IXCQUERY API

IXCQUERY is updated to support modify-form invocation.

The following sections describe associated publication updates.

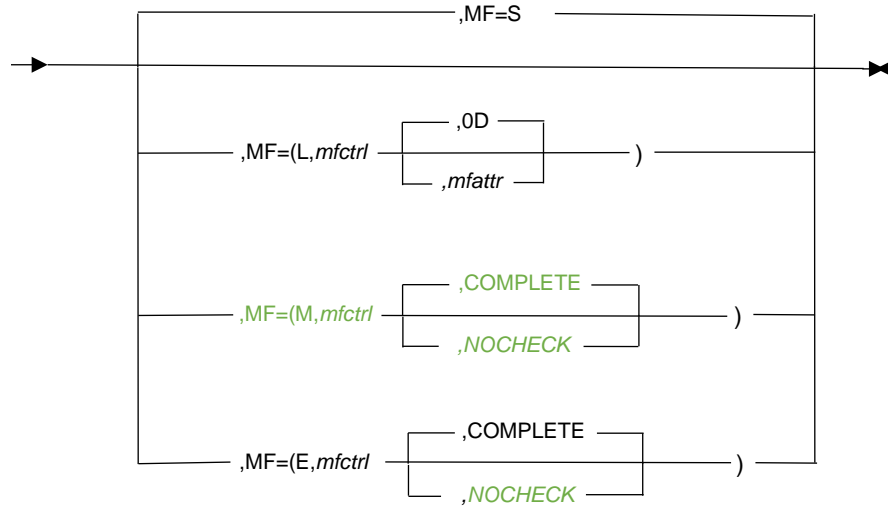
1.3.1 Sysplex Services Reference

Affected publication: z/OS MVS Sysplex Services Reference SA38-0658.

Changes in this section refer to Chapter "IXCQUERY – Obtain XCF Information"

1.3.1.1 Section "Syntax Diagram"

In the box marked "parameters-7", replace the bottom section of the diagram with the following:



1.3.1.2 Section "Parameter Descriptions"

Replace the description of the MF (macro form) keyword with the following:

,MF=S
,MF=(L,mfctrl)
,MF=(L,mfctrl,mfattr)
,MF=(L,mfctrl,OD)
,MF=(M,mfctrl)
,MF=(M,mfctrl,COMPLETE)
,MF=(M,mfctrl,NOCHECK)
,MF=(E,mfctrl)
,MF=(E,mfctrl,COMPLETE)
,MF=(E,mfctrl,NOCHECK)

Use MF=S to specify the standard form of the macro, which builds an in-line parameter list and generates the macro invocation to transfer control to the service.

Use MF=L to specify the list form of the macro. Use the list form together with the execute form of the macro for applications that require reentrant code. The list form defines an area of storage that the execute form uses to store the parameters. Only the PLISTVER parameter can be coded with the list form of the macro.

Use MF=M to specify the modify form of the macro. Use the modify form to generate code to put the parameters into the parameter list.

Use MF=E to specify the execute form of the macro. Use the execute form together with the list form of the macro for applications that require reentrant code. The execute form stores the parameters into the storage area defined by the list form, and generates the macro invocation to transfer control to the service.

,mfctrl

Use this output parameter to specify a storage area to contain the parameters.

To Code: Specify the RS-type name or address (using a register from 2 to 12) of the parameter list.

,mfattr

Use this input parameter to specify the name of a 1- to 60-character string that can contain any value that is valid on an assembler DS pseudo-op. You can use this parameter to force boundary alignment of the parameter list. If you do not code mfattr, the system provides a value of 0D, which forces the parameter list to a doubleword boundary.

.COMPLETE

,NOCHECK

Use this input parameter to specify the degree of macro parameter syntax checking the system is to do.

COMPLETE

Use this parameter to require that the system check for required parameters and supply defaults for omitted optional parameters.

NOCHECK

Use this parameter to specify that the system is not to check for required parameters nor to supply defaults for omitted optional parameters.

Note: In the macro expansion you might see some defaults for optional parameters that are not documented here. The ones that are not documented do not have any effect on the macro. For example, if SMILE=var were an optional parameter and the default is SMILE=NO_SMILE then it would not be documented. However, if the default was SMILE=-), then it would be documented because a value would be the default.