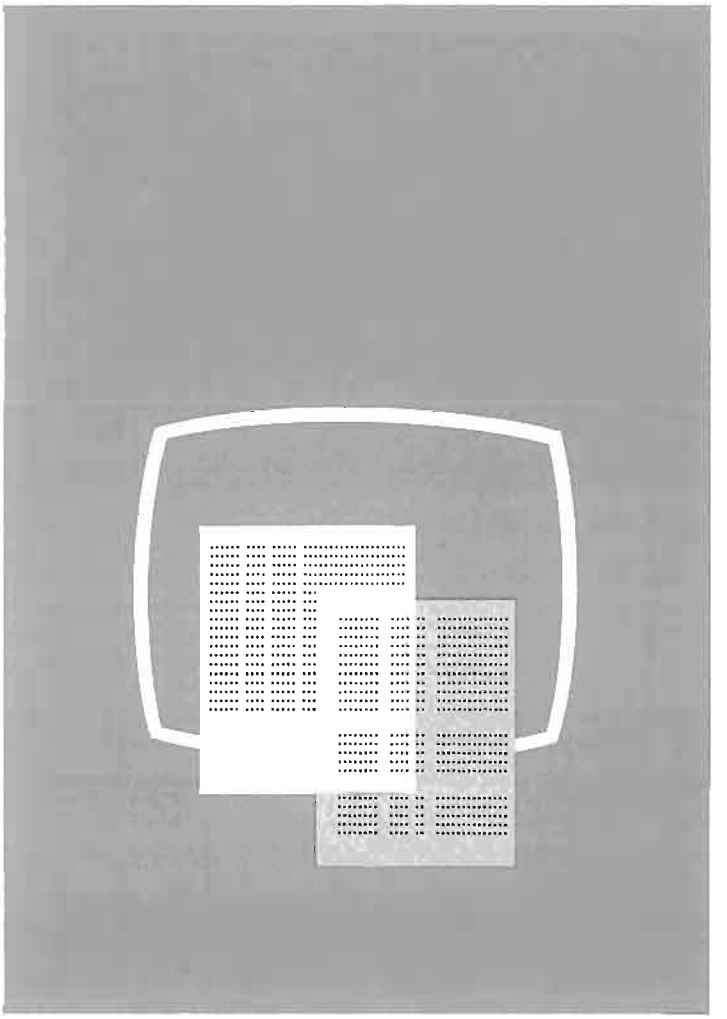
LY24-5217-2
File No. S370/4300-50

Program Product

SQL/Data System Logic Volume 2

Data Areas
Diagnostic Aids
Appendixes



IBM

Third Edition (August 1983)

This edition, LY24-5217-2, is a revision of LY24-5217-1. This edition applies to the Structured Query Language/Data System, Release 2, until otherwise indicated in new editions or Technical Newsletters. Changes are made periodically to the information herein; before using this publication in connection with the operation of IBM systems, consult the latest IBM System/370 and 4300 Processors Bibliography, GC20-0001, for the editions that are applicable and current.

Changes and additions to the SQL/Data System for Release 2 are described in the publication SQL/Data System Release 2 Guide, GH24-5042. Changes or additions to the text and illustrations are indicated by vertical line to the left of the change.

PREFACE

This logic manual provides some measure of detail about the internals of SQL/Data System (SQL/DS). It is intended for persons involved in determining programming problems, especially to determine whether there is a user or IBM programming problem. Further, if it is determined that there is an IBM problem, this manual, along with Volume 1 (and Volume 3 for VSE users with the Extract Facility), helps to provide (a) the information needed to determine where the problem lies or (b) sufficient information to assist IBM personnel in locating the problem and providing the necessary program modification.

ORGANIZATION OF THIS MANUAL

This manual contains three sections:

- "Section 5: Data Areas" describes the major SQL/DS control blocks and includes a control block interconnection diagram.
- "Section 6: Diagnostic Aids" contains a number of additional aids useful in diagnosing problems. It includes, but is not restricted to, information on the following:
  - Storage layout after initialization
  - How to find the modules automatic area
  - A list of DBSS OP codes and their meanings
  - A table of SQL code - RDS code pairs and the detecting modules followed by a table of modules and the code pairs that each module detects
  - A list of DBSS return codes and their meanings
  - A list of all the SQL/DS messages with the issuing modules followed by a list of modules with the messages they cause to be issued
  - A list of SQL/DS reason codes and their meanings
  - A section on DBS Utility processing of output records
  - Sections on tracing
  - A data-areas-to-module cross-reference list
  - A module-to-module cross-reference list.
- "Appendixes" contains two appendixes:

  Appendix A is a glossary of abbreviations and acronyms used in this manual.

  Appendix B is a dictionary of equivalent terms. It lists the terms used in this manual that have other names. Generally, these other names are those found in the code prologs and comments. However, on occasion you may find more than one name used for the same item in this manual.

USING THIS MANUAL

This manual is designed to be used with Volume 1 (LY24-5216) and, for VSE users with the Extract Facility, Volume 3 (LY24-5222). Volume 1 contains a high-level set of diagrams for SQL/DS, a method of operations section showing much of the control flow for the various components and functions, a section containing brief descriptions of each of the modules, and a directory section.

You may find it convenient and useful to have both (or all three for Extract users) volumes of the logic manual available and open, side-by-side for easy access to information pertinent to your current task.

PREREQUISITE PUBLICATIONS

SQL/Data System General Information, GH24-5012
SQL/Data System Concepts and Facilities, GH24-5013
SQL/Data System Planning and Administration -- VSE, SH24-5014
SQL/Data System Planning and Administration -- VM/SP, SH24-5043
SQL/Data System Installation -- VSE, SH24-5015
SQL/Data System Installation -- VM/SP, SH24-5044
SQL/Data System Terminal User's Guide -- VSE, SH24-5016
SQL/Data System Terminal User's Guide -- VM/SP, SH24-5045
SQL/Data System Terminal User's Reference, SH24-5017
SQL/Data System Application Programming, SH24-5018
SQL/Data System Operation, SH24-5020
SQL/Data System Data Base Services Utility, SH24-5046

RELATED PUBLICATIONS

SQL/Data System Messages and Codes, SH24-5019

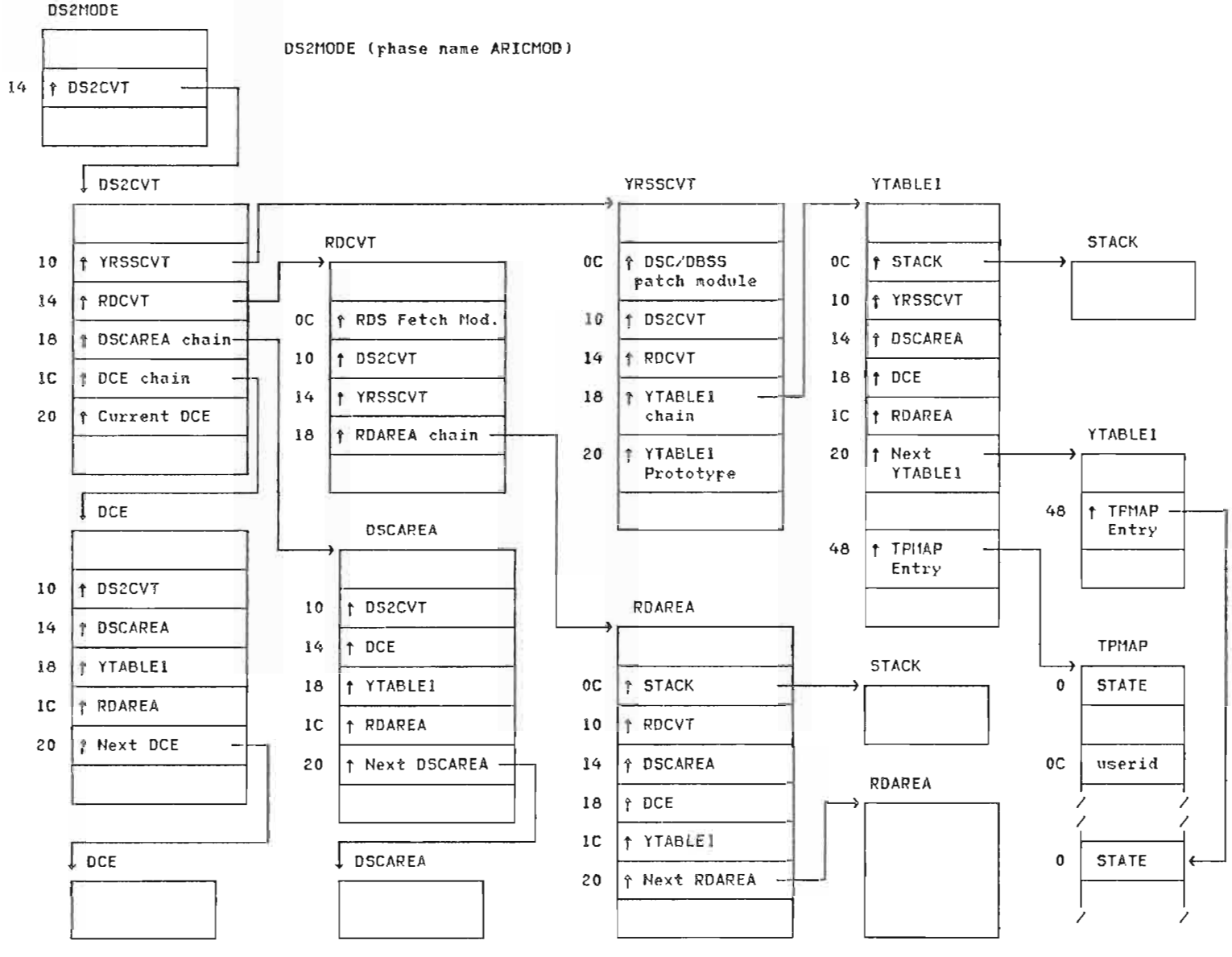
CONTENTS

SECTION 5: DATA AREAS

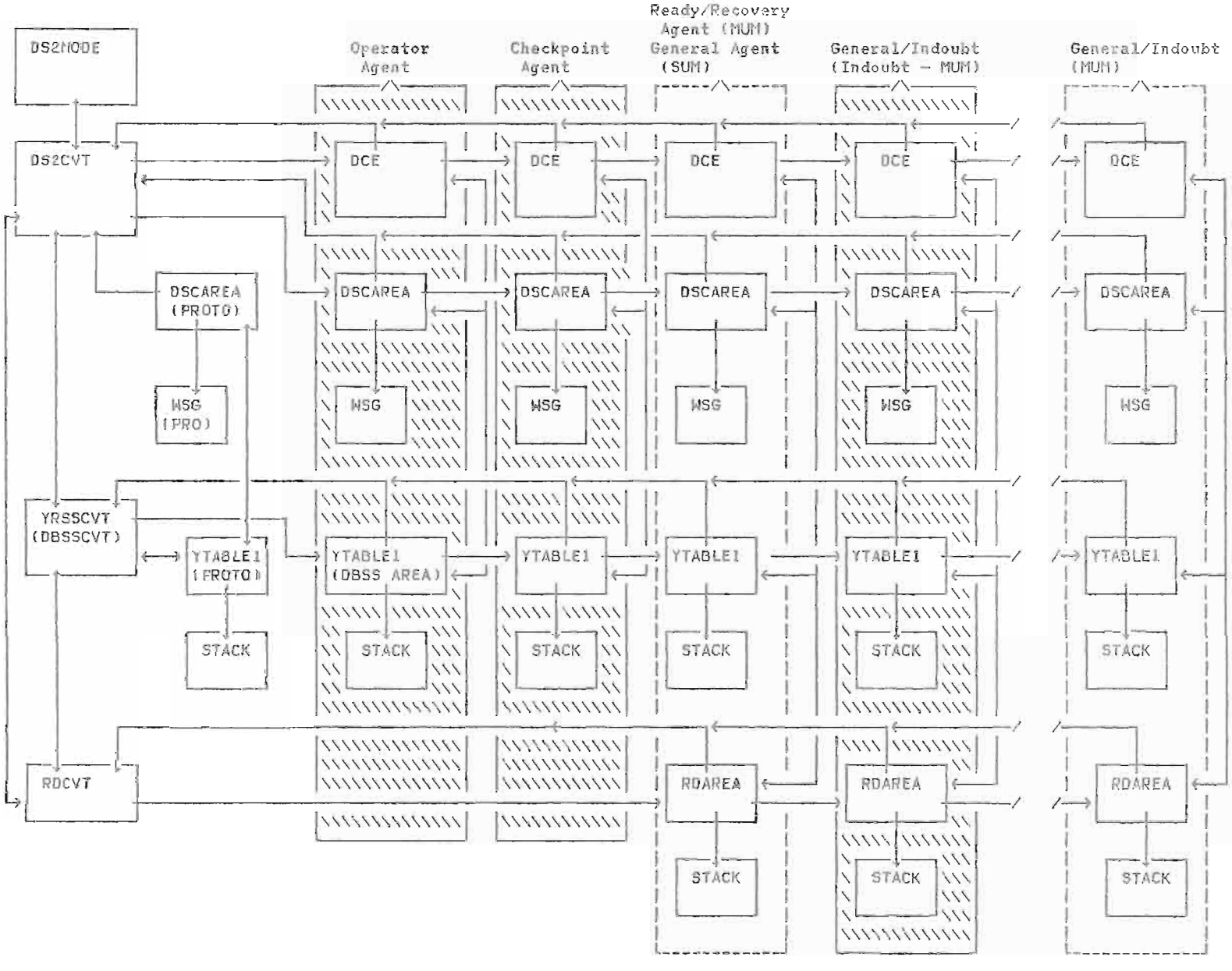
This section contains layouts and descriptions of many of the SQL/DS data areas. It also has some introductory diagrams on interconnection and overview.

MAJOR CONTROL BLOCKS - INTERCONNECTION DIAGRAM

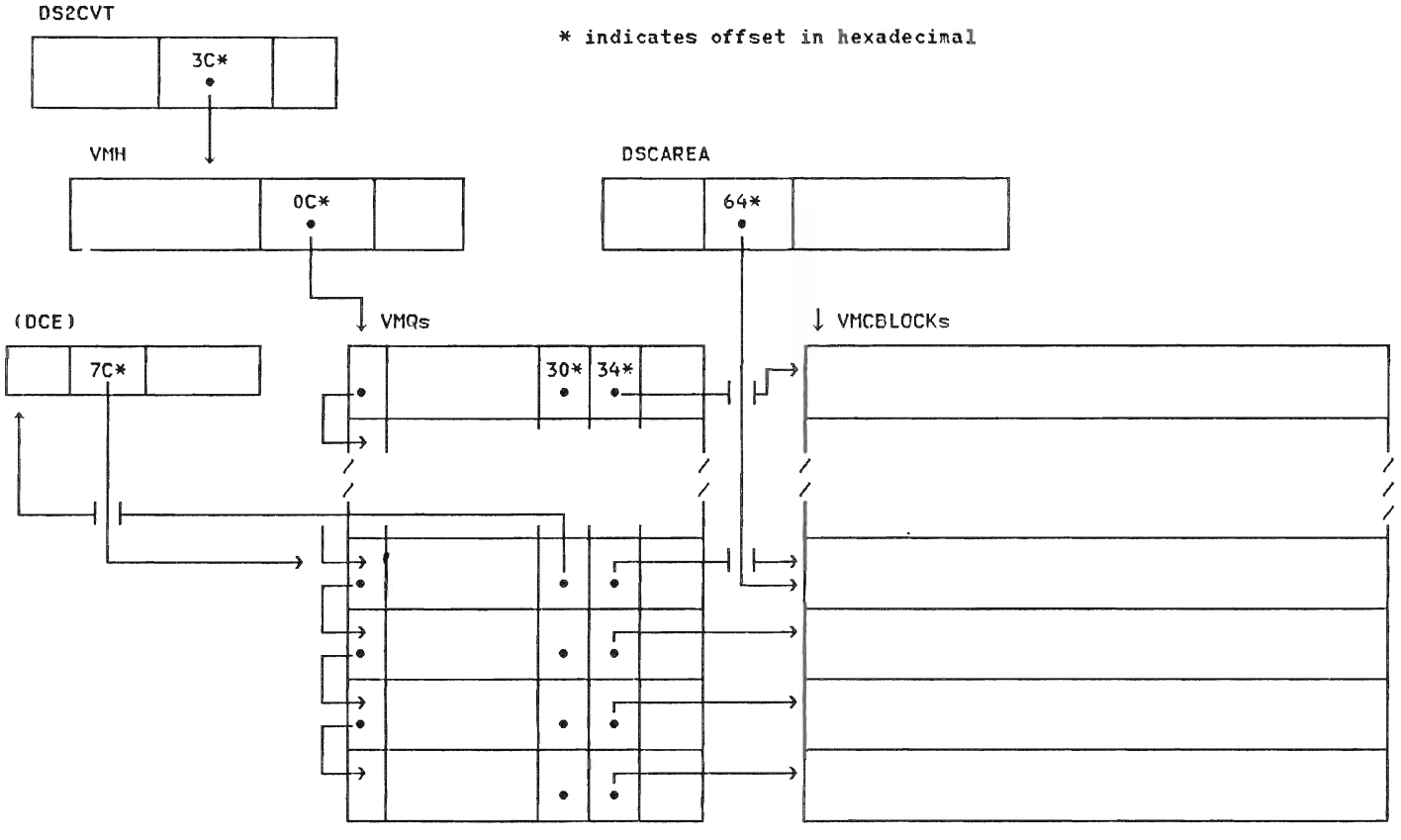
All offsets are given in hexadecimal.



SQL/DS AGENT STRUCTURE INTERCONNECTIONS

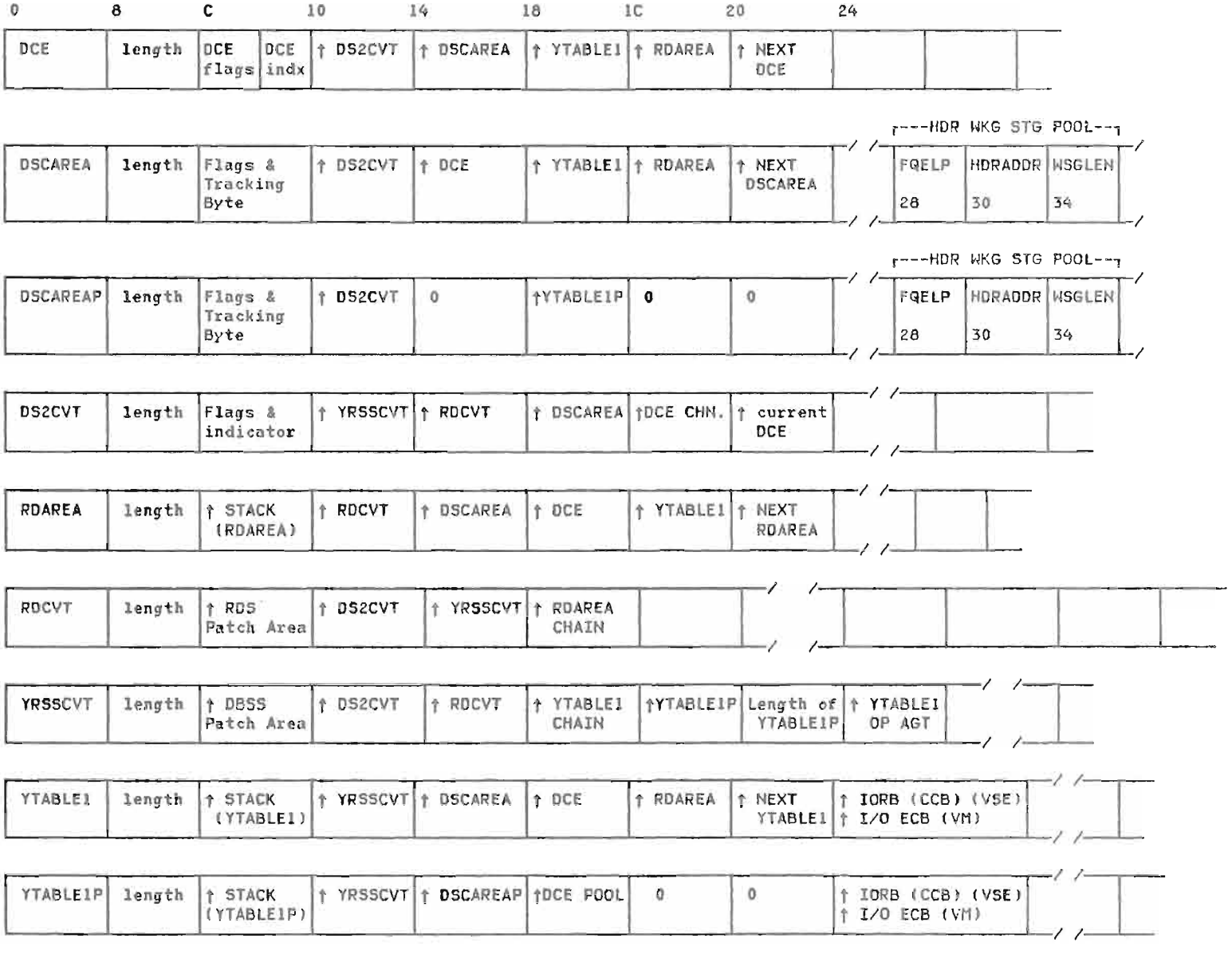


VM CROSS-MACHINE COMMUNICATION DATA AREAS

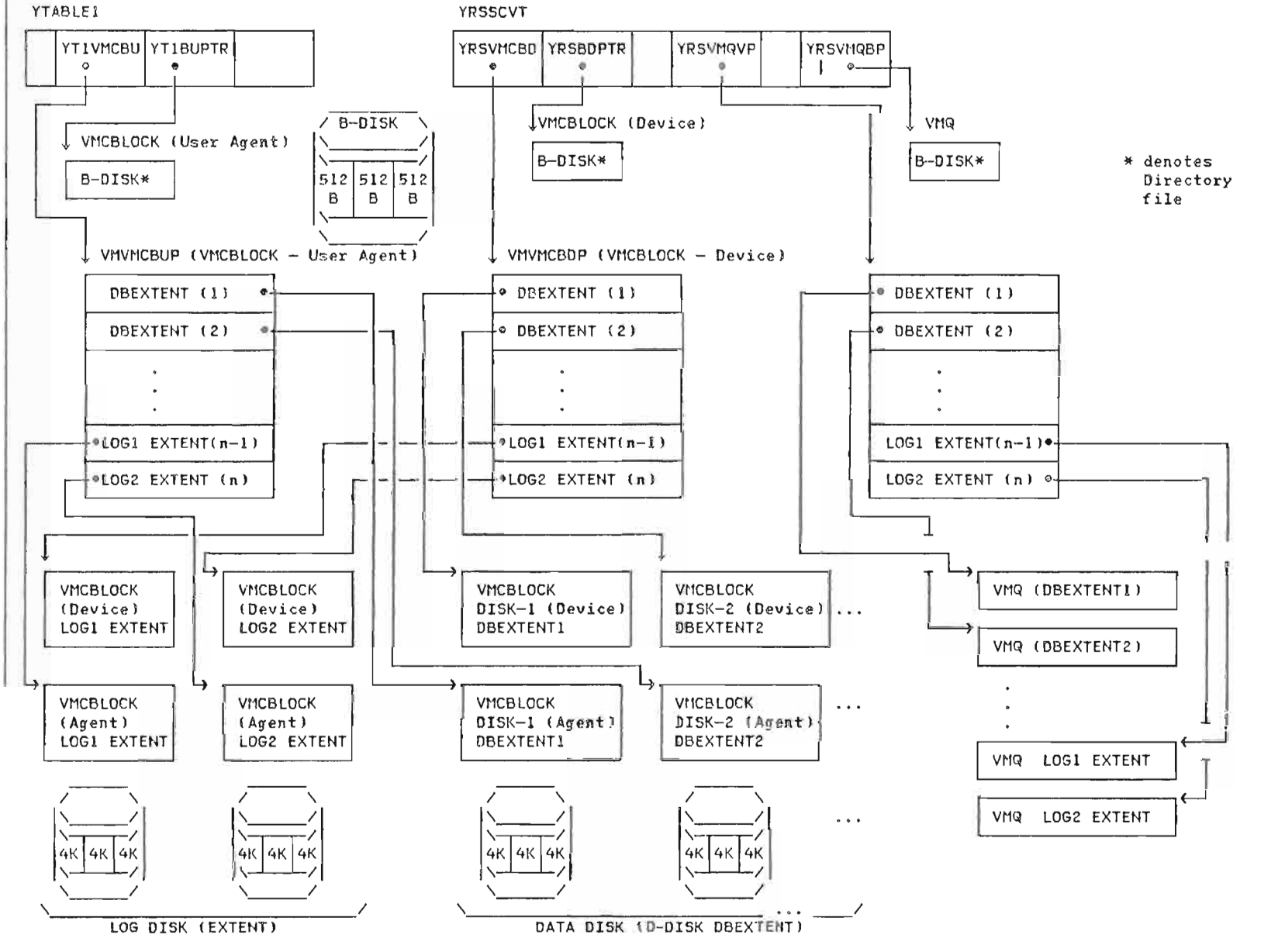


MAJOR CONTROL BLOCK OVERVIEW

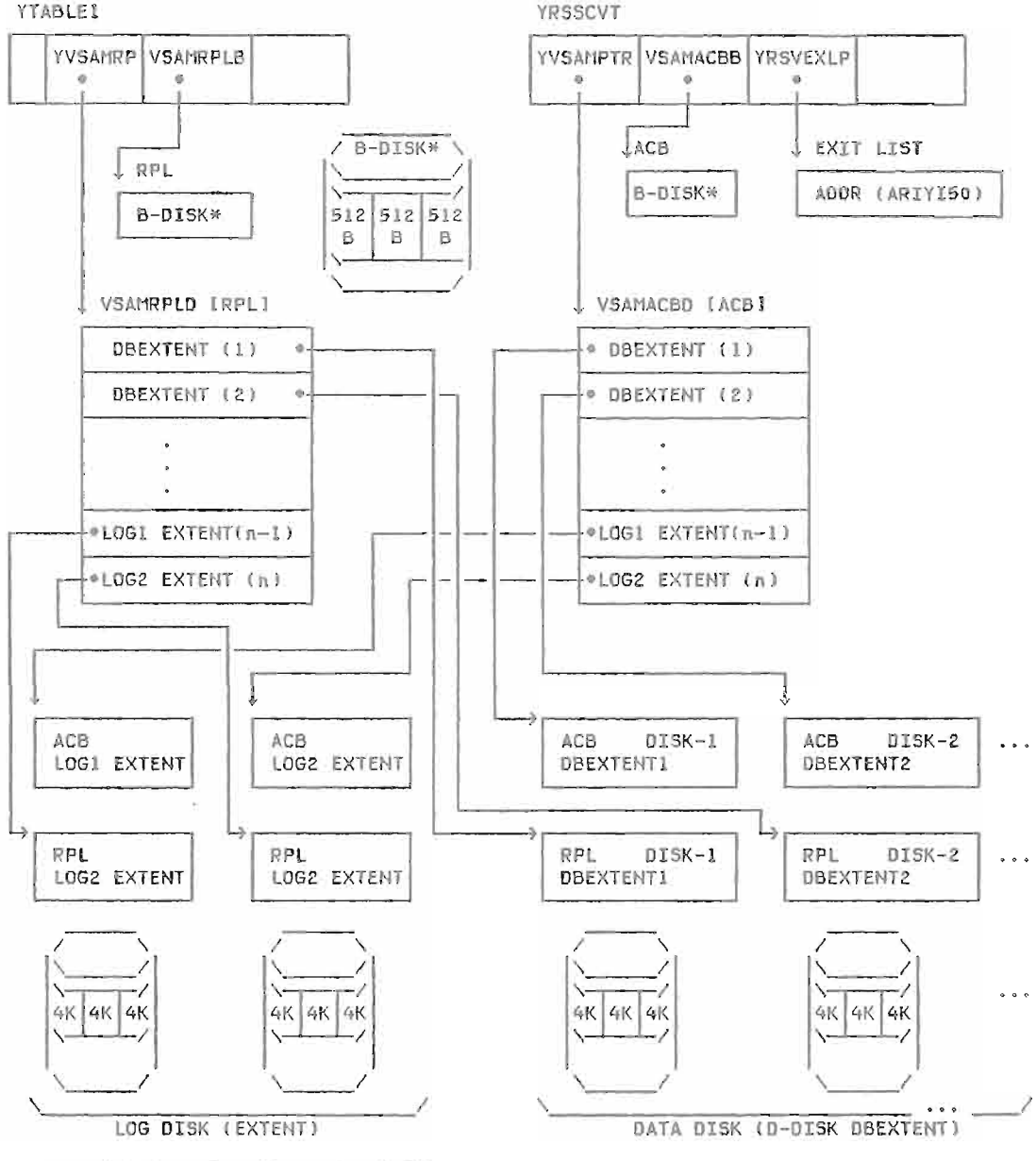
Offsets are in hexadecimal.



SQL/DS ASYNCHRONOUS I/O CONTROL BLOCKS - VM



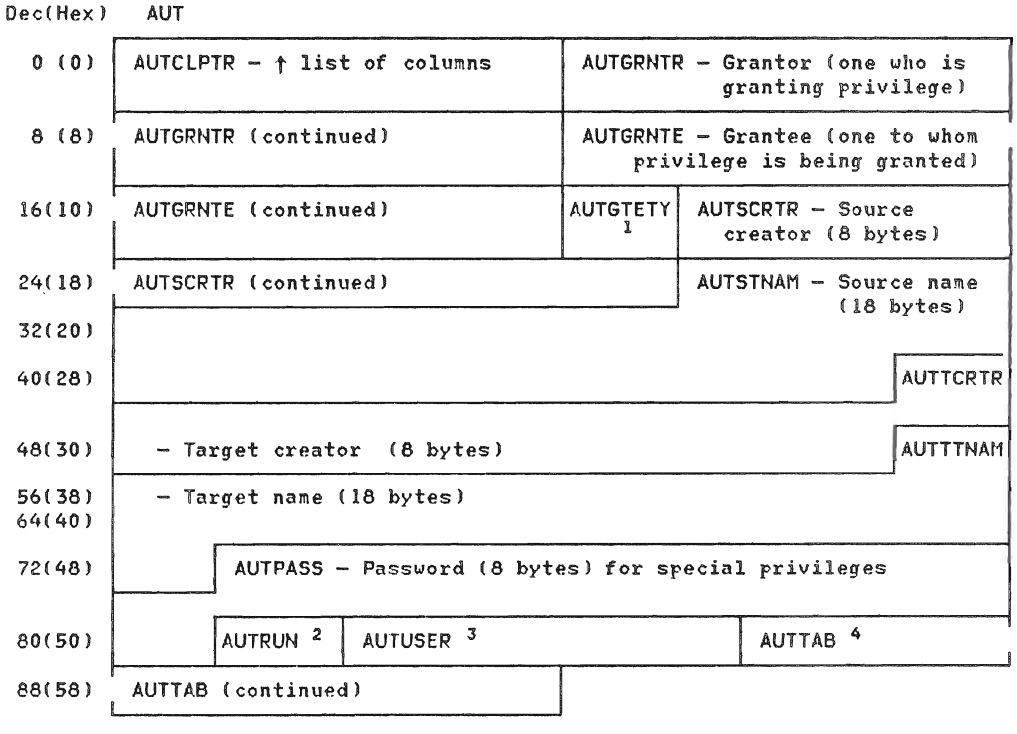
SQL/DS ASYNCHRONOUS I/O CONTROL BLOCKS - VSE



* Directory file

AUT (AUTHORIZATION PARAMETER LIST)

The AUT is pointed to by field PAUTHPTR of the PREP Control Block (PREPDSCB). AUT is the authorization structure used by RDS authorization functions.



[1] Grantee type:
'P' = program
' ' = user

[2] 'Y' = RUNAUTH is granted.
'G' = RUNAUTH is granted with GRANT option.
' ' = Field not used.

[3] Four single-character fields:
1. DBAAUTH 2. RESOURCEAUTH
3. CONNECTAUTH 4. SCHEDULEAUTH
'Y' if the privilege is granted,
' ' if the field is not used.

[4] Six single-character fields for each of the privileges to be granted in the order:
1. SELECTAUTH 2. INSERTAUTH
3. UPDATEAUTH 4. DELETEAUTH
5. EXPANDAUTH 6. INDEXAUTH
'Y' = Privilege is granted.
'G' = Privilege is granted with GRANT option
' ' = Field not used.

AUX LENGTH ROW (FIRST ROW OF AN AUX)

The first row in an access module is known as the access module length row. The format of this row is:

- For an access module not created via CREATE PROGRAM, the row consists of one varying-character column, two bytes long, which contains the number of sections in the access module.
- For an access module created via CREATE PROGRAM, the first row consists of one varying-character column, twelve bytes long, in the following format:

AUXLNROW

0 (0)	ALR#SECT - Number of sections in the access module	ALRFLAGS [1]	Resv'd	ALRLTID - TID of the last SQL statement row in the access module 7(7)
8 (8)	ALRSTID - TID of last SLT row 11(B)			

[1] ALRFLGS - Flags giving information about the access module:

ALRDESC	1...	DESCRIBEs are permitted for queries in this access module.
ALRMODFY	.1..	AUX may be modified.
	..xx xxxx	Reserved.

AUX SLT (ACCESS MODULE SECTION LOCATION TABLE)

The section location table contains information about each section within an access module.

PREP AUX SLT (PSLT)

This form of the SLT is used by the RDS Executives to keep track of the sections that are dynamically being added or dropped from an AUX. The three times in which this may happen are during:

1. The PREP process (PREP INIT CALL, SQL CALL, FINISH CALL)
2. The CREATE PROGRAM process (CREATE PGM, Extended PREPARE, COMMIT)
3. The AUX UPDATE process (DROP statement or Extended PREPARE, COMMIT)

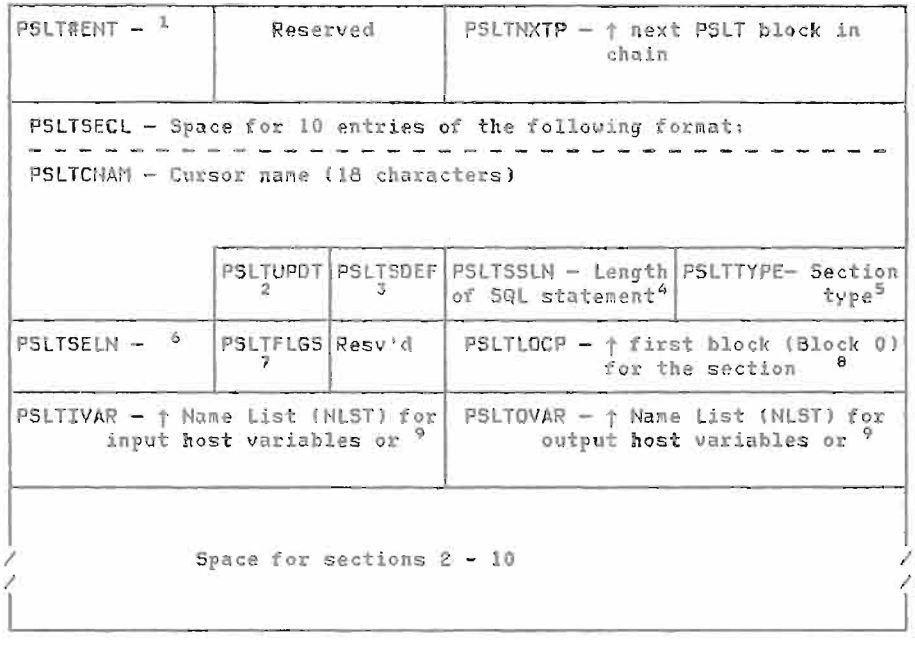
At the conclusion of 1 and 2, the PSLT entries are inserted into the AUX with at most 10 PSLT entries per row. At conclusion of 3, the updated PSLT is updated in the AUX, and, if necessary, it may have additional rows inserted.

During (2) and (3), AUX sections may be executed because the sections are accessible in storage and their addresses are saved in PSLTLOCP of each PSLT entry.

The following PSLT pointers are in the PREP Control Block (PREPDSCB):

- RETSLTPT - PREP's current SLT entry
- PPSLTPTR - Current SLT
- FRSTSLTP - First SLT

PSLT



> Header

1 Number of SLT entries (sections in access module)

2 Flag byte 1 - 'Y' if this section's cursor is updateable; 'D' if deletable. Necessary for WHERE CURRENT OF cursor operations. Set by Optimizer.

3 Flag byte 2 - 'Y' or 'N' indicates whether a DECLARE CURSOR is associated with this section.

4 Filled in by STORE AUX (ARIXESX)

5 INTERPSECT, CLASSSECT, PARSEDSECT, INDEFSECT, COMPILESECT.

6 Length of section. Filled in by STORE AUX (ARIXESX).

7 See "PSLT Flags" on page 11.

8 This field is changed when the SLT entry is stored in the data base. The first SLT entry contains the TID (row ID) of the last row of the last stored AUX section. For other SLT entries, this field is not used.

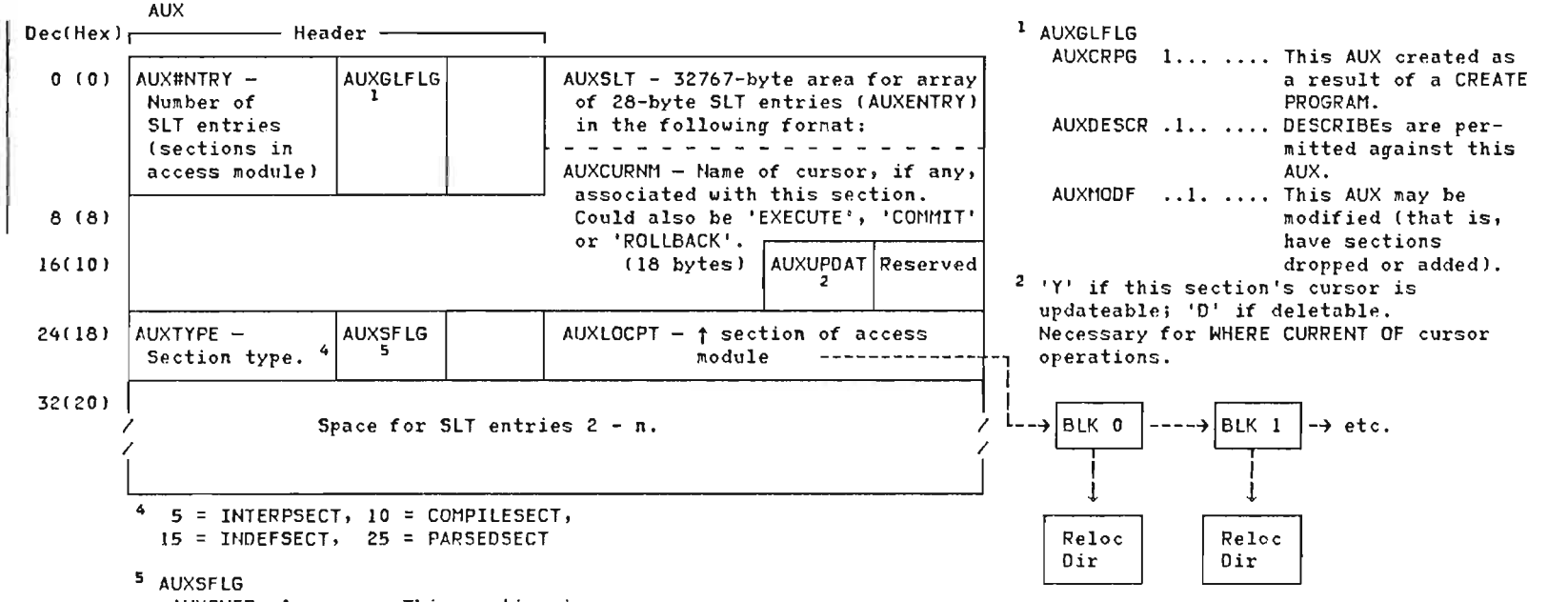
9 For access modules created using CREATE PROGRAM with the MODIFY option, these fields are used as:
PSLTIVAR - TID of the first stored row for that section.
PSLTOVAR - TID of the first SQL statement row for this section.

PSLT Flags

Offset	Field Name	Bits	Meanings
34(22)	PSLTFLGS		
	PSLTQUER	1...	This section is a query.
	PSLTREUS	.1..	This section may be reused because its section and SQL statement rows were deleted from the access module.
		..xx xxxx	Reserved.

RUN-TIME AUX SLT (AUX)

This form of the SLT is used at run-time to execute an access module. Its location is established from its access module entry in the AUX cache (PROGS). This form is a watered-doun version of the SLT as it is stored in the access module.



1 AUXGLFLG
- AUXCRPG 1... .... This AUX created as a result of a CREATE PROGRAM.
- AUXDESCR .1.. .... DESCRIBEs are permitted against this AUX.
- AUXMODF ..1. .... This AUX may be modified (that is, have sections dropped or added).

2 'Y' if this section's cursor is updateable; 'D' if deletable. Necessary for WHERE CURRENT OF cursor operations.

4 5 = INTERPSECT, 10 = COMPILESECT, 15 = INDEFSECT, 25 = PARSEDSECT

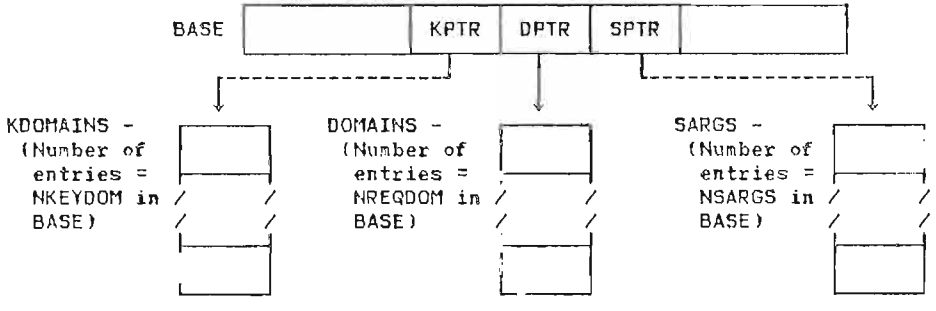
5 AUXSFLG
- AUXQUER 1... .... This section is a query.
- AUXREUS .1.. .... This section has been deleted or may be reused.

BASE AND AUXILIARY STRUCTURES

The BASE and auxiliary structures (KDOMAINS, DOMAINS, and SARGS) are used by the Data Manipulation component of DBSS in performing various row operations against SQL/DS tables.

The pointer to BASE, along with the opcode and return code parameters, is passed to the DBSI (ARIYM00) from RDS by module ARIXEDB. The auxiliary structure DOMAINS is pointed to by BASE field DPTR. Its array dimension is BASE field NREQDOM. The auxiliary structure KDOMAINS is pointed to by BASE field KPTR. Its array dimension is BASE field NKEYDOM. The auxiliary structure SARGS is pointed to by BASE field SPTR. Its array dimension is BASE field NSARGS.

During DBSS execution, YTABLE1 field YP points to BASE, and BASE itself is copied into YTABLE1 substructure YBASE. The following diagram shows the connection of the structures:



BASE

Dec(Hex) BASE

0	SEGMENT - DBSPACE number of the main row	RID - Table ID of the main row	TID - Row ID of the main row		
8	PSEGMENT - DBSPACE number of the secondary row[1]	PRID - Table ID of the secondary row[1]	PTID - Row ID of the secondary row[1]		
16(10)	SCANID - Scan ID	NKEYDOM - Number of submitted KDOMAINS key-fields	NREQDOM - Maximum field number for requested/ submitted DOMAINS fields	NSARGS - Number of submitted SARGS entries	
24(18)	KPTR - Pointer to KDOMAINS		DPTR - Pointer to DOMAINS		
32(20)	SPTR - Pointer to SARGS		LID - Link ID	IID - Index ID	
40(28)	ICOMP[2] - Comparison operator using index 'M', 'A', 'MA', 'B', 'MB', 'FI'	HOLDIND[3]	QUALF[4]	RCODE - Return code	
48(30)	FEEDBACK - Feedback Area				

Notes:

[1] Identifies the T2 row on a CONNECT request. Identifies the parent row on a GET NEXT request (for a link scan), an OPEN SCAN request (for a line), and a GET PARENT ROW request.

[2] M = Match
A = After
MA = Match or After
B = Before
MB = Match or Before
FI = First

[3] Hold Indicator - 'H' is hold is requested.

[4] Call Qualifier used in OPEN, NEXT, CONNECT, and INSERT.
- 'R', 'I', 'C', 'P', 'F', 'L' for OPEN of a table, index, link on a child, link on a parent, link on a first child, or open on a list.
- 'B', 'H', 'A' for before, here, and after
- 'B', 'A' are used in CONNECT and INSERT
- 'H', 'A' are used in NEXT

KDOMAINS

KDOMAINS describes the submitted key-fields. It is an array of dimension NKEYDOM (in BASE). Array entry format:

Dec(Hex) KDOMAINS - Used to describe submitted key-fields

0	KDOMPTR - Pointer to submitted key-field	KDOMLTH - Length of the submitted key-field	FILLER - Padding for alignment

DOMAINS

DOMAINS identifies the fields to be retrieved or submitted. DOMAINS is pointed to by DPTR in the BASE structure. It is an array of dimension NREQDOM (in BASE). Array entry format:

Dec(Hex) DOMAINS

0	FLDPTR - Pointer to field IOAREA	FACLTH - Length of the stored/ submitted field	FREQLTH[1]

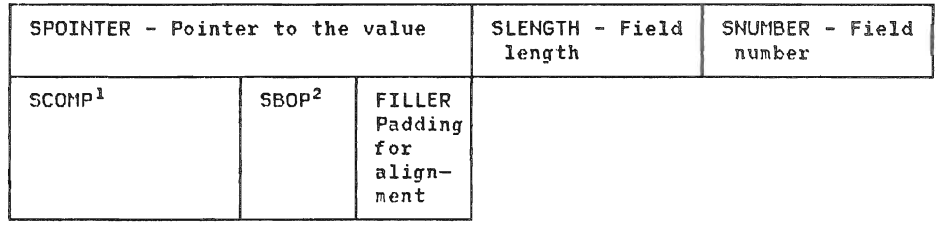
Notes:

[1] FREQLTH -
Read Calls: A maximum requested field length. A negative value indicates that a field is not being requested.
Update: A negative value indicates that a field is not to be updated.

SARGS

SARGS represents a disjunctive form of the comparisons on the field values. SARGS is an array of dimension NSARGS (a field in BASE). SARGS is pointed to by SPTR in the BASE structure. Only FETCH, OPEN, and NEXT look at NSARGS. Array entry format:

Dec(Hex) SARGS



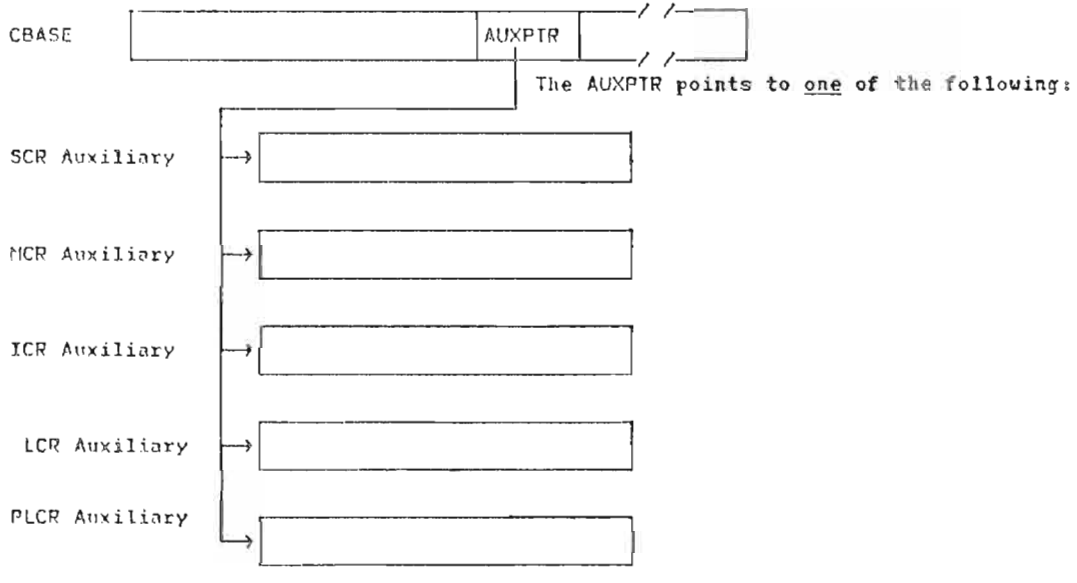
Notes:

[1] SCOMP - Field Comparison Operators: '= ', '> ', '< ', '>=', '<=', '¬=', 'LK'

[2] SBOP - Boolean Operators: '|' (or), '&' (and)

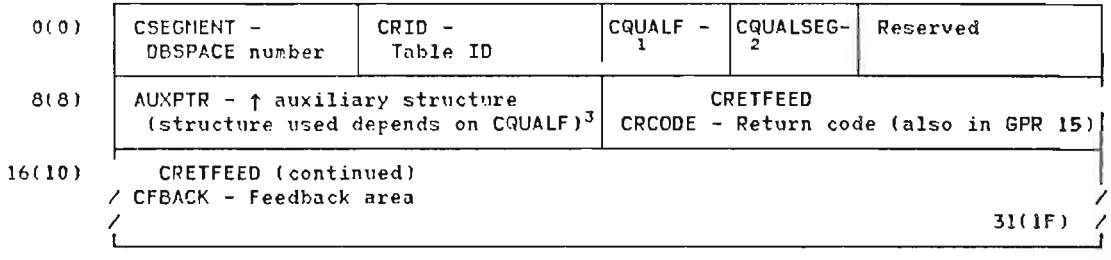
CBASE AND AUXILIARY STRUCTURES

The CBASE and auxiliary structures are used by the Data Control component of DBSS in updating the Control Header information in a DBSPACE. The pointer to the base structure, along with the opcode and return code parameter, is passed to DBSI (ARIYM00) from RDS via module ARIXEDB. The auxiliary structures are located using the AUXPTR field of CBASE. The following diagram shows the connection of the structures ("relations" and "tables" may be used interchangeably):



CBASE

Dec(Hex) CBASE



1 CALL qualifier:
S, 1, I, C, or P

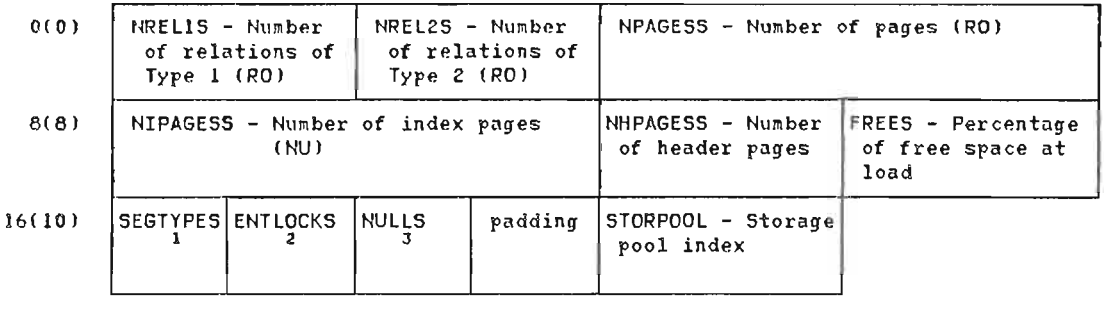
2 Secondary qualifier S, P, or T.
Used in CNEXT when CSEGMENT = 0.

3 Pointer dependent on CQUALF:

CQUALF	Pointer to
S	SCR
R,1	MCR,LIST
I	ICR
C	LCR
P	PLCR

SCR AUXILIARY

Dec(Hex) SCR



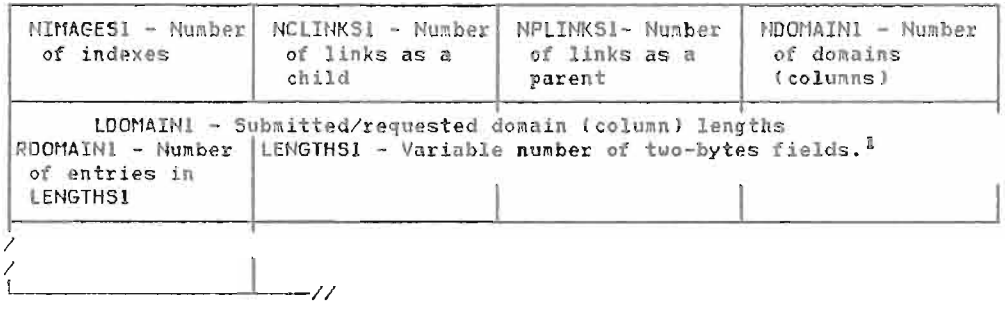
1 DBSPACE type: 1 to 5 (RO)

2 Entity lock protocols:
'P' if page only locks
'T' if TID locks
'S' if DBSPACE lock

3 Null (initial) character (NU)

MCR AUXILIARY

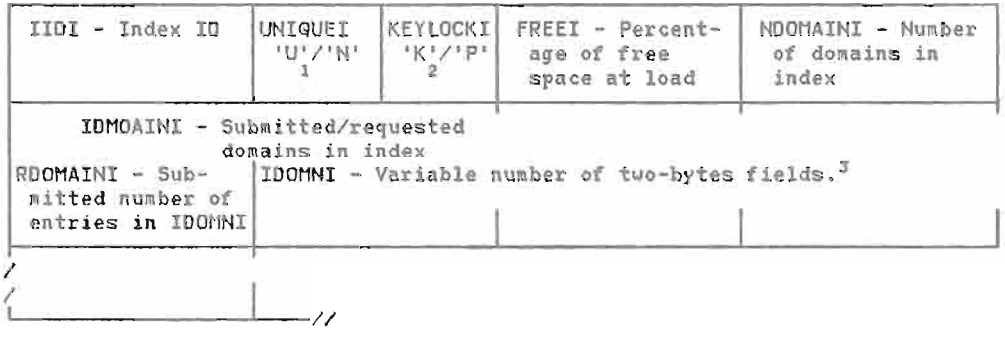
Dec(Hex) MCR1



[1] Submitted/returned domain lengths. Negative value for variable-length domain. Updates to the first (existing) NDOMAIN1 entries are ignored.

ICR AUXILIARY

Dec(Hex) IMAGE1



[1] 'U' if unique-key index; 'N' if otherwise

[2] 'K' if key lock 'P' if page lock

[3] Submitted/requested domains index. Sign is positive if domain is ascending, or negative if domain is descending

LCR AUXILIARY

Dec(Hex) LINKL

0(0)			
	LIDL - Link ID	PSEGMENL - Parent DBSPACE number. 0 if unary link	PRIDL - Parent table ID.

PLCR AUXILIARY

Dec(Hex) PLINKP

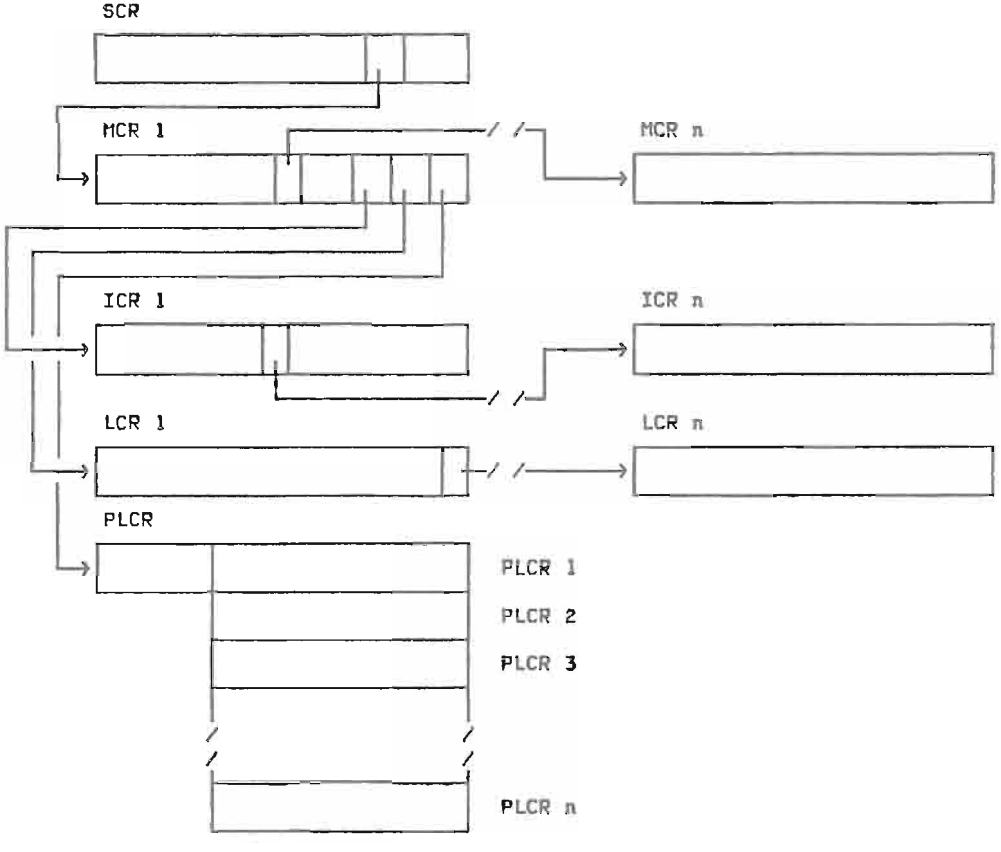
0(0)			
	LIDP - Link ID	CSEGMENP - Child DBSPACE number.	CRIDP - Child table ID.

CONTROL HEADER INFORMATION IN DBSPACE

Control header information resides in and describes the DBSPACE. DBSS uses the CBASE and auxiliary structures passed from RDS to update the Control Header Information.

The TID of SCR is always '8000'X. Copies of the major portions of SCR, MCR, and LCR are in YTABLE1 at YSHEADER, YMASTREC, and YLINKREC, respectively.

The SCR, MCR, ICR, LCR, and PLCR are all mapped by mapping macro YCONTROL.



SCR (DBSPACE CONTROL RECORD)

Dec(Hex) SHEADER

0(0)	SRFLAGS- Record flags	STID - TID in page=0	SRLTH - Record length = 22		SRTYPE- Record type = S	SFLAGS[1]	SFREE[2]	SNULL - Null char.
8(8)	HMAXPN[3]	MAXPN - Maximum page number in DBSPACE			Reserved	EMAXPN - Maximum entity page number in DBSPACE, or last used page number if List DBSPACE		
16(10)	NRELS[4]	Zero	SRELNEXT - ↑ first table (of Type 1 row)		Zeros			
						21(15)		

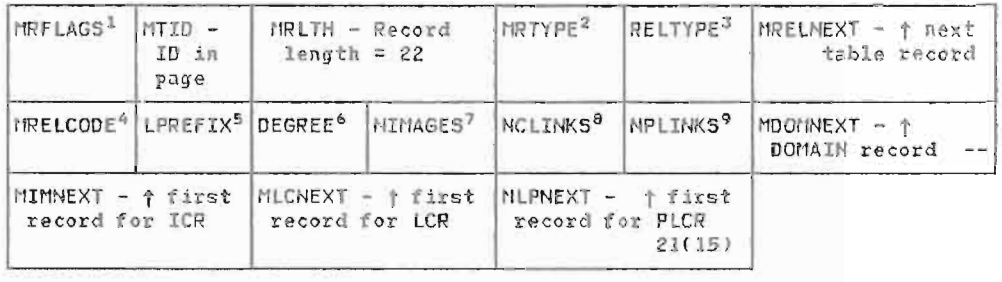
[1] See SHEADER Flags
[2] Percentage of free entity space at load. SFREE = 0 if a List DBSPACE
[3] Maximum page number (third byte) in DBSPACE header. HMAXPN = '80'X if a List DBSPACE.
[4] Number of tables of Type 1 in DBSPACE. NRELS = 1 if a List DBSPACE

SHEADER Flags

Offset	Field Name	Bits	Meanings
5(5)	SFLAGS		
	TLOCK	0...	Entity page logical locking
	LISTFLG	.0..	DBSPACE of tables of Type 1
		.1..	DBSPACE containing a list
	SGLOCK	..0.	No DBSPACE locking
		..1.	DBSPACE locking
		...x xxxx	Reserved

MCR (MASTER CONTROL RECORD FOR TYPE 1 TABLE OR LIST)

Dec(Hex) MASTREC



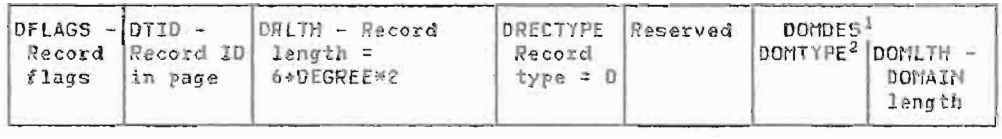
-- See next block for DOMAIN record

[1] Record flags
[2] Record type:
R = Relation (table)
L = List
[3] Relation (table) type:
Bit 1 = 0: Type 1
[4] Relation (table) ordinal (stored in rows)
[5] Size (bytes) of link prefix in row
[6] Number of DOMAINS
[7] Number of indexes
[8] Number of links in which child
[9] Number of links in which parent

DOMAIN RECORD

This record describes all DOMAINs (columns) of a Type 1 table.

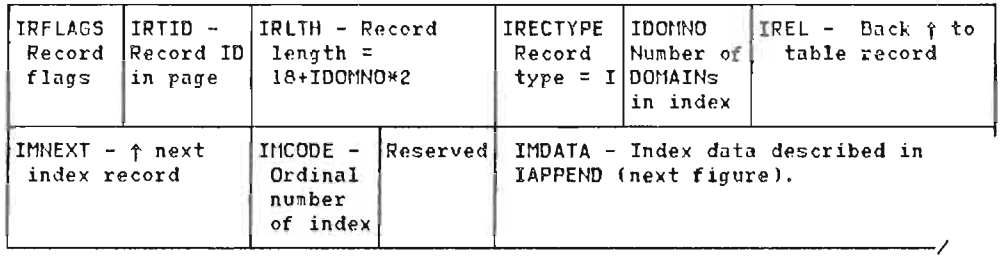
Dec(Hex) DOMREC



[1] DOMAIN descriptors. DOMAIN may be of variable length.
[2] Bit 1 = 0 if variable length DOMAIN.
Bit 2 = 0 if Indexed DOMAIN.

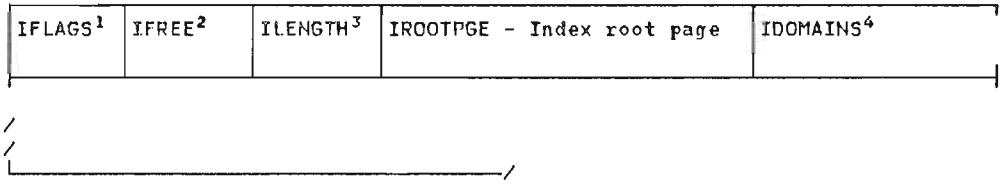
ICR (INDEX CONTROL RECORD)

Dec(Hex) IMAGEREC



Common Appendix to IMAGEREC

Dec(Hex) IAPPEND



[1] See "Common Appendix Flags"
[2] Percentage of free space at load
[3] Length for fixed--length case.
[4] Array of DOMAIN numbers (2 bytes each) with sign: positive for ascending; negative for descending.

Common Appendix Flags

Offset	Field Name	Bits	Meanings
0(0)	IFLAGS		
	IUNIQUE	1...	Unique key
	IFIXLTH	.1..	Fixed-length keys
	ITYPE	..00	Type 1 table
		..01	Type 2 unary
		..11	Type 2 binary
	ILOCKING	.... 1...	Fine key interval locking
		.... 0...	Page locking at leaf
	IORDER	.... .x..	For variable-length keys: 0 = short keys are high-order 1 = short keys are low-order
	IALL4SC	.... ..0.	All ascending DOMAINs
		x	Reserved

LCR (LINK CONTROL RECORD)

Dec(Hex) LINKREC

0(0)	LRFLAGS[1]	LRTID - Record ID in page	LRLTH - Record length = 18	LRECTYPE Record type = L	Reserved	LCODE[2]	LCHILDPR[3]
8(8)	LCREL - Table ID of child		LPSEG - DBSPACE number of parent. 0 if unary link.	LPREL - Table ID of parent		LCTIDP[4]	LPTIDP[5]
	LCNEXT[6]						

[1] Record flags.
[2] Ordinal number of link.
[3] Length of prefix of child. Used in open of a parent to allocate SCB space.
[4] Index of first TID pointer in child vector of TID pointers. TID pointers are consecutive in this order: RIGHT TWIN, LEFT TWIN, PARENT.
[5] Index of first-child TID pointer in parent vector of TID pointers.
[6] Pointer to next LCR row in which LCREL is a child.

PLCR (PARENT LINK CONTROL RECORD)

Dec(Hex) PLINKREC

0(0)	PLRFLAGS[1]	PLRTID[2]	PLRLTH - Record length = 6 + 6*NPLINKS	PLRTYPE- Record type = P	Reserved	PLINKS[3] PLIDSEG - DBSPACE number of link
	PLINKS (continued)					
8(8)	PLIDLID - LID of link		PLIDRID - Relation (table) ID of child in link			

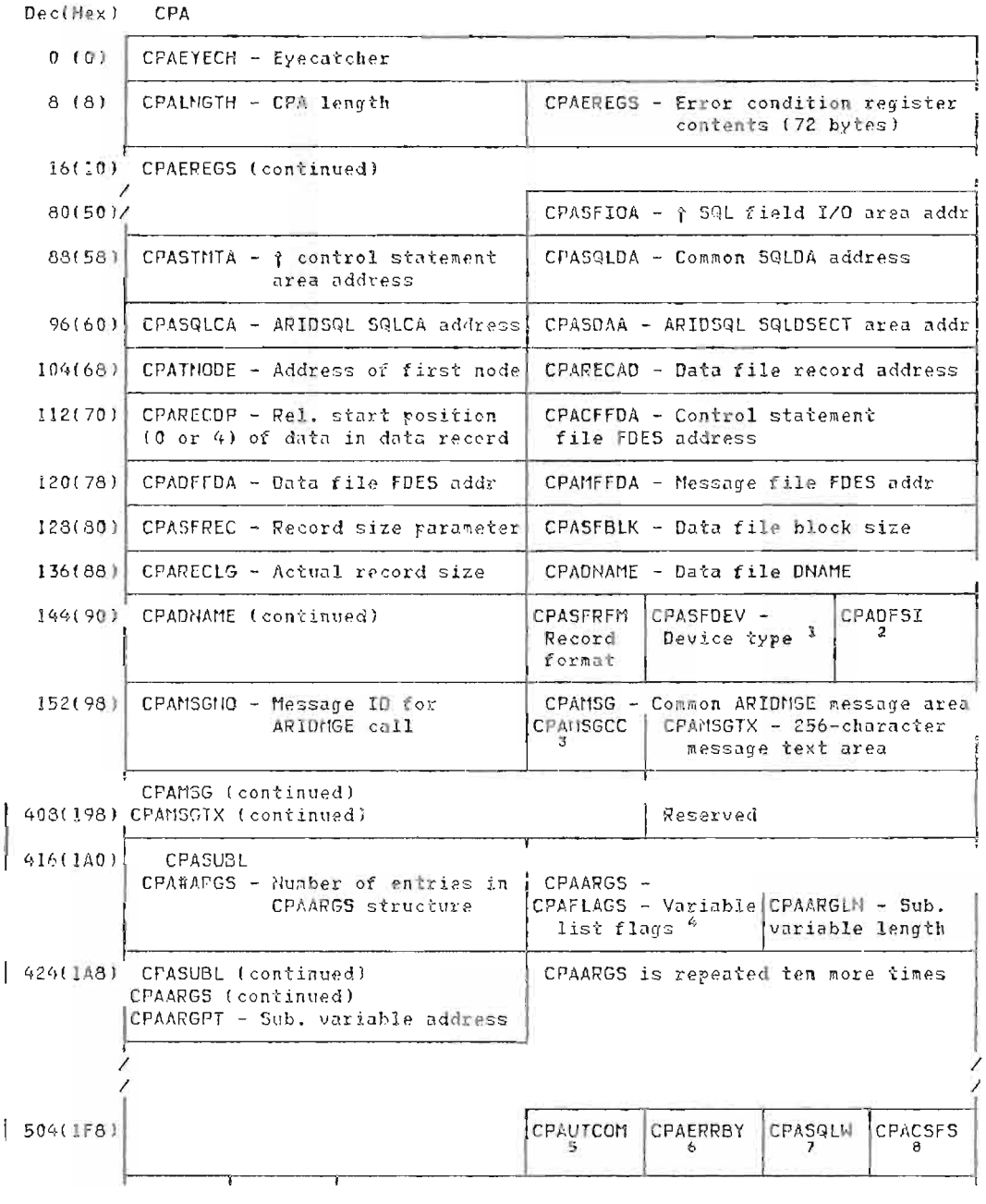
[1] Record flags.
[2] Record ID in page.
[3] PLINKS consists of a variable number of six-byte entries (broken as shown above into three two-byte fields) for links in which relation (table) is a parent.

CPA (DBS UTILITY COMMON PROCESSING AREA)

The CPA contains information relating to the DBS Utility environment. Register 4 points to the CPA during execution of the utility.

Alphabetic List of Field Names

CPA#ARGS 280(118)
CPAARGLN 286(11E)
CPAARGPT 288(120)
CPAARGS 284(11C)
CPACFFDA 116(074)
CPACIPI 376(178)
CPACONTR 630(276)
CPACOPO 710(2C6)
CPACPL 482(1E2)
CPACPLH 854(356)
CPACSCTL 377(179)
CPACSDLO 468(1D4)
CPACSEP 467(1D3)
CPACSFDN 378(17A)
CPACSFS 375(177)
CPACSSP 466(1D2)
CPACWA 386(182)
CPADBSC 606(25E)
CPADBSN 586(24A)
CPADBSNL 586(24A)
CPADBSNT 588(24C)
CPADFFDA 120(078)
CPADFSI 151(097)
CPADID 520(208)
CPADIDOP 856(358)
CPADIDVT 858(35A)
CPADIDCV 524(20C)
CPADIDSP 520(208)
CPADIDV 526(20E)
CPADIDVL 524(20C)
CPADNAME 140(08C)
CPAEMGO 851(353)
CPAEREGS 012(00C)
CPAERRBY 373(175)
CPAEYECH 000(000)
CPAFLAGS 284(11C)
CPAGENER 492(1EC)
CPAINMOD 500(1F4)
CPAINMP1 692(2B4)
CPAINMP2 700(2BC)
CPALNGTH 008(008)
CPALPP 478(1DE)
CPALPPCA 504(1F8)
CPALPPH 852(354)
CPALPPMA 508(1FC)
CPALPPMX 480(1E0)
CPAMFDN 469(1D5)
CPAMFFDA 124(07C)
CPAMODIO 708(2C4)
CPAMSG 156(09C)
CPAMSGCC 156(09C)
CPAMSGFS 477(1DD)
CPAMSGNO 152(098)
CPAMSGTX 157(09D)
CPANOLST 629(275)
CPANOVAR 284(11C)
CPAP1ADR 496(1F0)
CPAP2ADR 500(1F4)
CPAP2HOB 508(1FC)
CPARECAD 108(06C)
CPARECDP 112(070)
CPARECLG 136(088)
CPASDAA 100(064)
CPASFBLK 132(084)
CPASFDEV 149(095)
CPASFIOA 084(054)
CPASFITC 614(266)
CPASFITN 622(26E)
CPASFREC 128(080)
CPASFRFM 148(094)
CPASFTNL 622(26E)
CPASFTNT 624(270)
CPASQLCA 096(060)
CPASQLDA 092(05C)
CPASQLSW 711(2C7)
CPASQLW 374(176)
CPASTMTA 008(058)
CPASUBL 280(118)
CPASYSID 848(350)
CPATCICD 670(29E)
CPATCICL 670(29E)
CPATCICN 672(2A0)
CPATCIFL 516(204)
CPATCIFS 512(200)
CPATCIFT 690(2B2)
CPATCISC 631(277)
CPATERMI 860(35C)
CPATERMO 861(35D)
CPATNODE 104(068)
CPATODC 642(282)
CPATODID 709(2C5)
CPATODN 650(28A)
CPATODNL 650(28A)
CPATODNT 652(28C)
CPAUSER 484(1E4)
CPAUSTAT 862(35E)
CPAUTCOM 372(174)



1 Device type:
TA = Tape
DA = Disk

2 Data file status indicator:
0 = open for output
2 = open for input
4 = normal EOF or close request
8 = error, close required
12 = closed
16 = open for output required
18 = open for input required
20 = close error
24 = open error

3 Message line control character

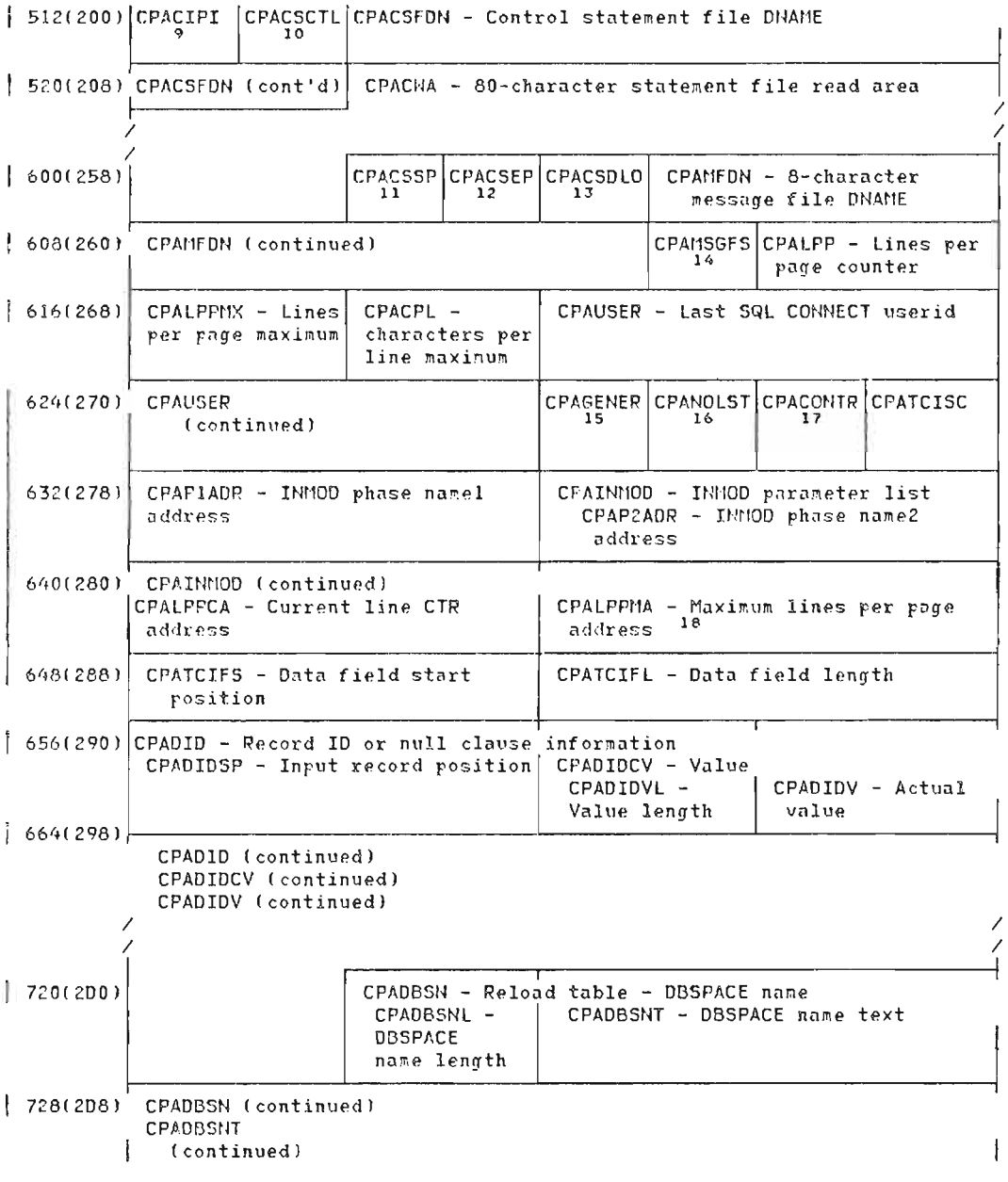
4 CPAFLAGS - Variable list flags:
CPANOVAR 1... .... Null substitution variable

5 AUTOCOMMIT ON/OFF indicator:
0 = OFF; 1 = ON

6 Error parse only underway:
0 = NO; 1 = YES

7 SQLWARNING indicator:
0 = NO; 1 = YES

8 See flags on page 30.



9 In-process control statement ID

10 See flags on page 30.

11 Control statement start position

12 Control statement end position

13 DATALOAD control statement processing indicator:
- 0 = Not in-progress
- 31 = Possible NEXT is a TCI
- 33 = Embedded input in progress

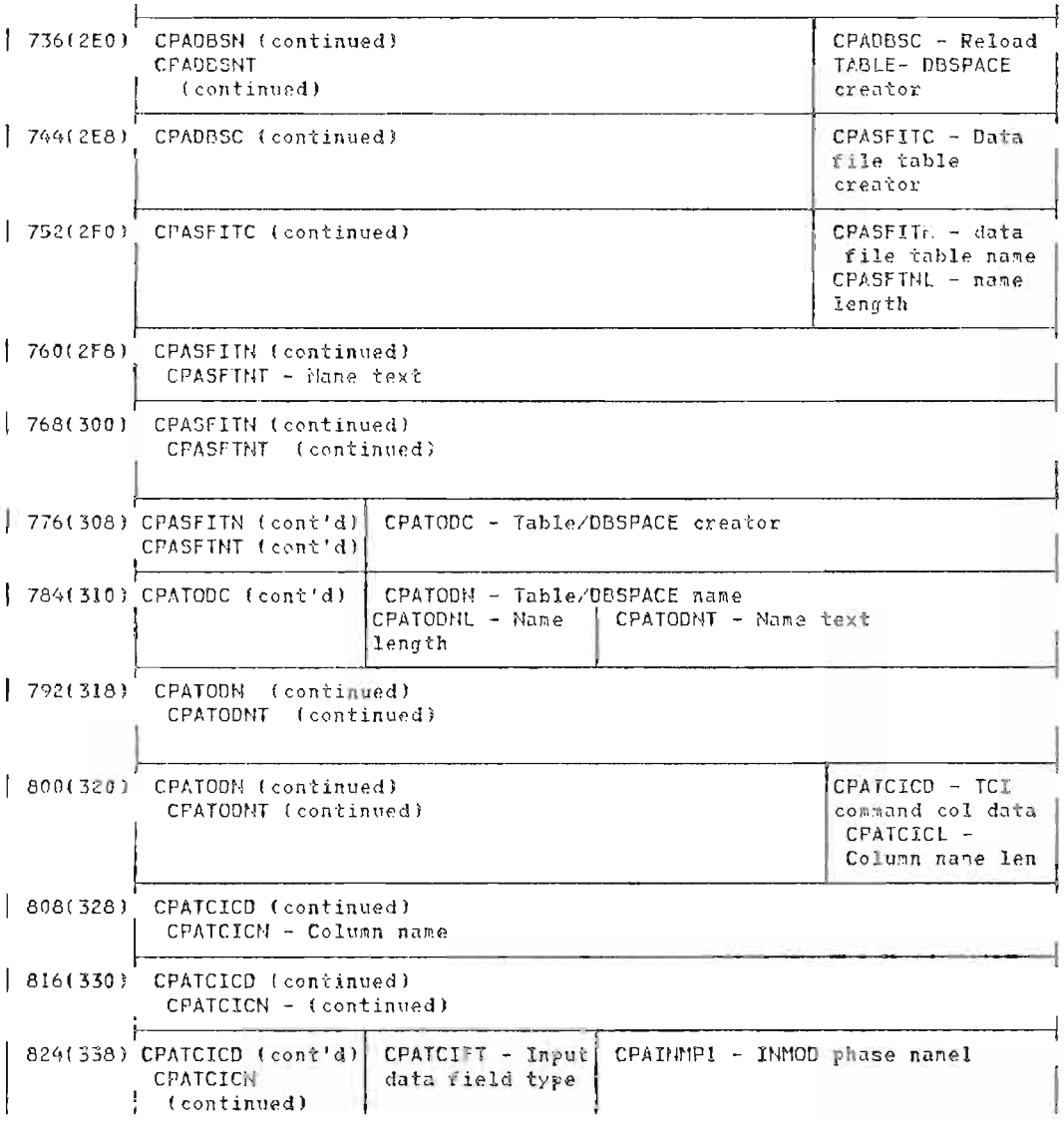
14 See flags page 30.

15 Generate COMMIT or ROLLBACK:
- 0 = OFF - not generated by DBS
- 1 = ON - generated by DBS

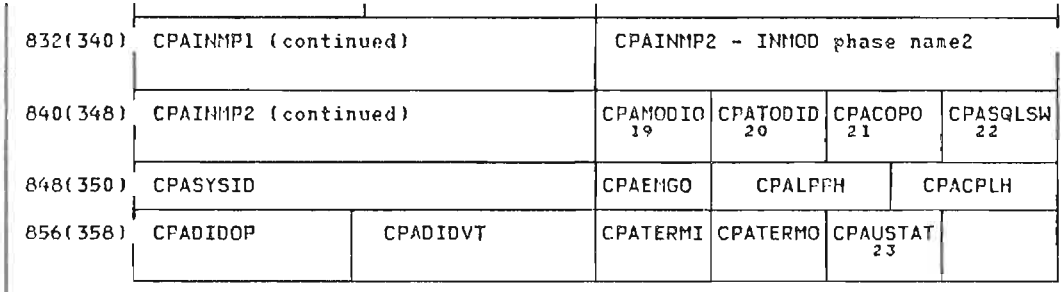
16 DATALOAD DATA LIST Suppression:
- 0 = Not in effect
- 1 = In effect

17 DATALOAD CONTINUED RECORD:
- 0 = Not in effect
- 1 = In effect

18 Bit 0 (CPAP2HOB) - parameter list flag bit



[19] INMOD I/O indicator
[20] Table/DBSPACE indicator:



T = Table
D = DBSPACE

21 CREATE/PURGE option ID

22 SQL processing indicator:
0 = OFF - Do not continue
1 = ON - Continue

23 Update Statistics:
0 = OFF
1 = ON (Default)

CPA Flags and Indicators

Offset	Field Name	Meaning
375(177)	CPACSFS	Control statement file status, 'CPACSFS' value descriptions: 0 = Open 4 = Normal EOF or close request 8 = Error; close required 12 = Closed 16 = Open required 20 = Close error 24 = Open error
377(179)	CPACSCTL	ARIDCSP process control value descriptions: 0 = Next record not read 4 = Next record already read 8 = EOF, another call expected 12 = Error, another call expected 16 = No more calls expected
477(1DD)	CPAMSGFS	Message file status value descriptions: 0 = Open 4 = Normal EOF, close required 8 = I/O error, close required 12 = Closed 16 = Open required 20 = Close error 24 = Open error

CPLIST (SQL/DS COMMUNICATION MANAGER PARAMETER LIST) - VSE

The CPLIST is used to drive the CPC (Cross-Partition Communication) function. It is used to isolate the various users of the CPC functions from the idiosyncrasies of the host operating system CPC control blocks.

The CPLIST contains the cross-partition (XPTN) link names, the XPTN ECBs, the pointers to the buffers, the length of the buffers and the data transferred, the pointers to the host operating system CPC control blocks, and the fields for passback from the host system.

A CPLIST is pointed to by the DSCAREA (DSCCMPLP, offset X'60') for that agent structure. (See page 3 for an overview of the agent structures.) A CPLIST is also pointed to by field RMLTCPLP (offset 16(10) of the Resource Manager data area RMLT (see page 194).

Dec(Hex) CPLIST

Dec(Hex)					
0	CPLAPNAM - Name of this application				
8	CPLTOAP - Name of partner application				
16(10)	CPLTUDTA - User data to be transmitted CPLTDTA1 - Transmitted user data, Part 1	CPLTDTA2 - Transmitted user data, Part 2			
24(18)	CPLRUDTA - Receive area for user data. CPLRDTA1 - Receive user data, Part 1	CPLRDTA2 - Receive user data, Part 2			
32(20)	CPLIDBLK - ↑ LOGON control block	CPLCNBLK - ↑ CONNECT control block.			
40(28)	CPLCECB - ↑ CONNECT ECB	CPLSECB - ↑ SEND(R) ECB			
48(30)	CPLRECB - ↑ RECEIVE ECB	CPLBUFLN - Length of buffer (general message).			
56(38)	CPLBUFAD - ↑ buffer (general message).	CPLREPLN - Length of buffer to receive reply.			
64(40)	CPLREPAD - ↑ buffer to receive reply.	CPLRECLN - Length of transmitted message.			
72(48)	CPLWTECB - ↑ ECB to be waited on.	CPLUSEXT - Address of user exit routine.			
80(50)	CPLTID - LOGON task-id (task-id from system)	CPLRETCD Return code [1]	CPLREQCD [2]	CPLWATIN [3]	CPLSBTYP [4]
88(58)	CPLLPFN [5]	Reserved 91(5B)			

[1] See "Return Codes" on page 32.
[2] Communication Manager request code (see "Request Codes" on page 32.
[3] WAIT/NOWAIT indicator:
Y = WAIT; N = NOWAIT
[4] Subsystem request type indicator:
N = NOTIFY; R = REMOVE
[5] See "CPLIST Flags" page 32.

Return Codes - VSE

When register 15 = 4:
Note: CPLNOC2 will be converted into the CPLDISC code.
CPLNOC3 will be converted into the CPLABDC code.
Reason codes IJBXDAPP through IJBXNCNN will be converted to CPLDAPP to CPLNCNN respectively (the CPL codes will be equal to the IJBX codes plus 7).

Dec (Hex)
01 (01) Partner issued a RESETR
02 (02) Partner issued a RESETS
03 (03) Partner issued a RECV (SENDR)
04 (04) Buffer list changed after command
05 (05) Partner went to EOJ without connect
06 (06) Partner disconnected (normal)
07 (07) Partner disconnected (abnormal)
08 (08) Duplicate application name in system
09 (09) Duplicate application name in partner
10 (0A) Partner has more CONNECT requests
11 (0B) No logon by partner
12 (0C) No connect by partner
17 (11) Request was cleared
25 (19) Partner terminated normally
26 (1A) Partner terminated abnormally
27 (1B) Partner just issued a CLEAR

When register 15 = 8:

8 (08) Warning code indicator
23 (17) Partner issued a SHUTDOWN

When register 15 = 12(0C):

06 (06) Control block error
07 (07) Wrong logon (identify) key
08 (08) Wrong connect path id
09 (09) Task does not own control blk
10 (0A) Buffer indicator in error
11 (0B) Buffer length too large
12 (0C) Receive buffer too small
13 (0D) More than 64 CONNECT requests
14 (0E) No system storage for control block
15 (0F) Issued a sub-system command
16 (10) No request pending
18 (12) Request already pending
19 (13) Invalid request sequence
20 (14) Connection still busy - owner
21 (15) SEND(R) still pending
22 (16) Connection still busy - partner
24 (18) No CONNECT done by partner
28 (1C) Wrong XPCCB address

Request Codes - VSE

Dec (Hex)
01 (01) LOGON
02 (02) CONNECT SPECIFIC
03 (03) SEND
04 (04) SEND WITH REPLY
05 (05) RECEIVE
06 (06) REPLY
07 (07) RESET SENDR
08 (08) RESET REC/REPLY
09 (09) DISCONNECT
10 (0A) DISCONNECT UNCONDITIONALLY
11 (0B) DISCONNECT ALL
12 (0C) LOGOFF
13 (0D) LOGOFF UNCONDITIONALLY
14 (0E) SHUTDOWN
15 (0F) WAIT
16 (10) SUBSID
18 (12) CONNECT ANY

CPLIST Flags

Offset	Field Name	Bits	Meanings
88(58)	CPLLPFN (Cross-Partition		Communication flags (other side))
	CPLFRPLY	..1.	REPLY indicator.
	CPLFRSTR	...1	PURGE indicator.
	CPLFRCV	.... 1...	RECEIVE indicator.
	CPLFRSTS	.... .1..	CLEAR indicator.
	CPLFSNDR	.... ..1.	SENDR indicator.
	CPLFSEND	1	SEND indicator.
		xx..	Reserved

CPLIST (SQL/DS COMMUNICATION MANAGER PARAMETER LIST) - VM

The CPLIST is used to drive the IUCV (Inter-User Communication Vehicle) function. It is used to isolate the various users of the IUCV functions from the idiosyncrasies of the host operating system communication control blocks.

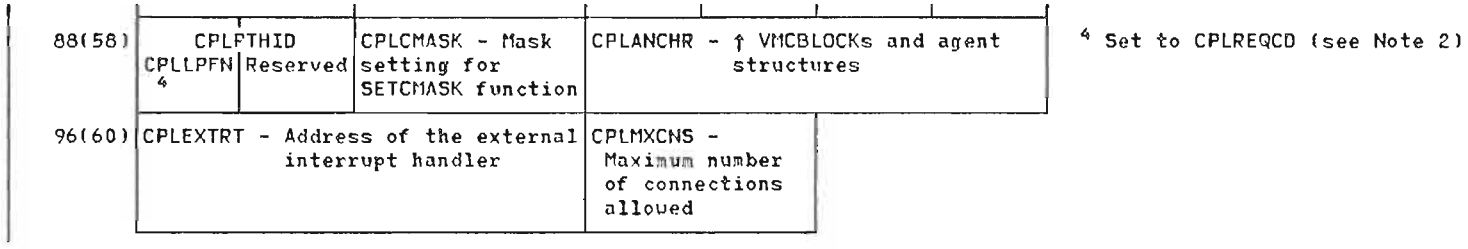
The CPLIST contains the cross-machine link names, the ECBs, the pointers to the buffers, the length of the buffers and the data transferred, the pointers to the host operating system communication control blocks, and the fields for passback from the host system.

A CPLIST is pointed to by the DSCAREA (DSCCMPLP, offset X'60') for that agent structure. (See page 3 for an overview of the agent structures.) A CPLIST is also pointed to by field RMLTCPLP (offset 16(10) of the Resource Manager data area RMLT (see page 194).

Dec(Hex) CPLIST

0	CPLAPNAM - Name of this application					
8	CPLTOAP - Name of partner application					
16(10)	CPLTUDTA - User data to be transmitted CPLDDNAM - DDNAME for block I/O CPLTDTA1 - Transmitted user data, Part 1	CPLTDTA2 - Transmitted user data, Part 2				
24(18)	CPLRUDTA - Receive area for user data. CPLRDTA1 - Receive user data, Part 1	CPLRDTA2 - Receive user data, Part 2				
32(20)	CPLIDBLK - ↑ LOGON control block	CPLCNBLK - ↑ CONNECT control block.				
40(28)	CPLCECB - ↑ CONNECT ECB	CPLSECB - ↑ SEND(R) ECB				
48(30)	CPLRECB - ↑ RECEIVE ECB CPLNOBLK - Number of blocks for block I/O	CPLBUFLN - Length of buffer (general message). CPLIOBSZ - Blocksize for block I/O				
56(38)	CPLBUFAD - ↑ buffer (general message).	CPLREPLN - Length of buffer to receive reply.				
64(40)	CPLREPAD - ↑ buffer to receive reply.	CPLRECLN - Length of transmitted message. CPLIOFST - Offset for block I/O				
72(48)	CPLWTECB - ↑ ECB to be waited on.	CPLUSEXT - Address of user exit routine.				
80(50)	CPLTID - LOGON task-id (task-id from system) CPLIOCUU - Virtual device address (CUU) | CPLIOFLG - Status flags for block I/O	CPLRETCD Return code [1]	CPLREQCD [2]	CPLWATIN [3]	CPLSBTYP Not used for VM	

[1] See "Return Codes" on page 34.
[2] Communication Manager request code (see "Request Codes" on page 35.
[3] WAIT/NOWAIT indicator:
Y = WAIT; N = NOWAIT



[4] Set to CPLREQCD (see Note 2)

Return Codes for VM

When register 15 = 4:

Dec (Hex)
- 01 (01) Partner issued a RESETR
- 06 (06) Partner disconnected (normal)
- 07 (07) Partner disconnected (abnormal)
- 10 (0A) Target communicator not logged on
- 11 (0B) No identify by partner
- 12 (0C) No connect by partner
- 17 (11) Request was cleared
- 175 (AF) Virtual device is read only

When register 15 = 8:

Dec (Hex)
- 08 (08) Running Single Partition Mode
- 23 (17) Partner issued a SHUTDOWN
- 254 (FE) Invalid external interrupt

When register 15 = 12(0C):

- 04 (04) HNDLIUCV SET already issued
- 08 (08) No HNDLIUCV SET issued
- 10 (0A) DASD not reserved
- 12 (0C) User does not own the path
- 16 (10) NAME parameter not specified/address = 0
- 20 (14) EXIT parameter not specified/address = 0
- 24 (18) Parameter list not specified/address = 0
- 28 (1C) SEVER with IPALL = 1 is not allowed
- 28 (1C) DDNAME not defined
- 32 (20) Declare buffer already issued by a non-CMS IUCV user
- 36 (24) Error in reading the directory
- 36 (24) Disk not accessed
- 40 (28) Invalid CMS IUCV function name
- 101 (65) Invalid path id
- 102 (66) Path quiesced - no sends allowed
- 103 (67) Message limit exceeded
- 104 (68) Priority messages not allowed
- 105 (69) Receive/reply buffer too short
- 108 (6C) Invalid message class or path id
- 110 (6E) Message length is negative
- 113 (71) More than 64 CONNECT requests
- 115 (73) No authorization found
- 116 (74) Invalid CP systen service name
- 117 (75) Invalid function code - IPFCND
- 118 (76) Message limit exceeds 255
- 119 (77) Declare buffer already issued
- 121 (79) Fetch protection on send buffer
- 122 (7A) Address exception on send buffer
- 123 (7B) Fetch protection on answer buffer
- 124 (7C) Address exception on answer buffer
- 125 (7D) Fetch protection on receive buffer
- 126 (7E) Address exception on receive buffer
- 127 (7F) Fetch protection on reply buffer
- 128 (80) Address exception on reply buffer
- 129 (81) No message found (condition code = 2)
- 151 (97) Virtual device is not defined
- 152 (98) Virtual device not supported
- 153 (99) Blocksize not supported
- 154 (9A) IUCV path already exists
- 155 (9B) Connection not using PRMMSG=Y

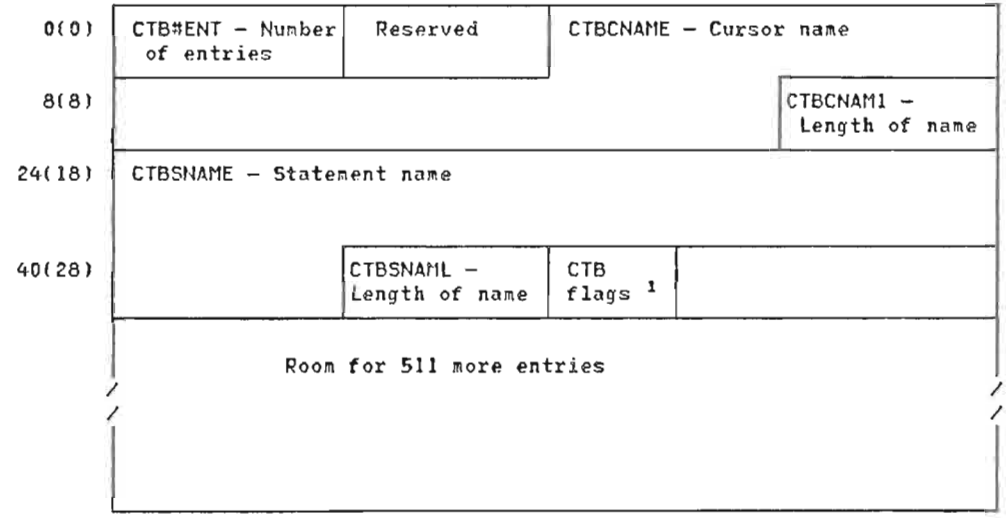
Request Codes for VM

Dec (Hex)	
01 (01)	LOGON
02 (02)	CONNECT
03 (03)	OPEN
04 (04)	SEND WITH REPLY
05 (05)	RECEIVE
06 (06)	REPLY
07 (07)	SETMASK
08 (08)	RESETR
09 (09)	DISCONNECT PURGE
11 (0B)	DISCONNECT ALL
12 (0C)	LOGOFF
13 (0D)	CLOSE
15 (0F)	WAIT
18 (12)	ACCEPT

CTB (CURSOR TABLE)

The Cursor Table maintains a list of all cursors found in an application program. If a cursor was declared directly on a statement name, the statement name will also be in the cursor table. If the cursor was declared on a SELECT statement, the statement name field will be blank. The Cursor Table is local to the Preprocessor and is not linked to other control blocks.

Dec(Hex) CTB



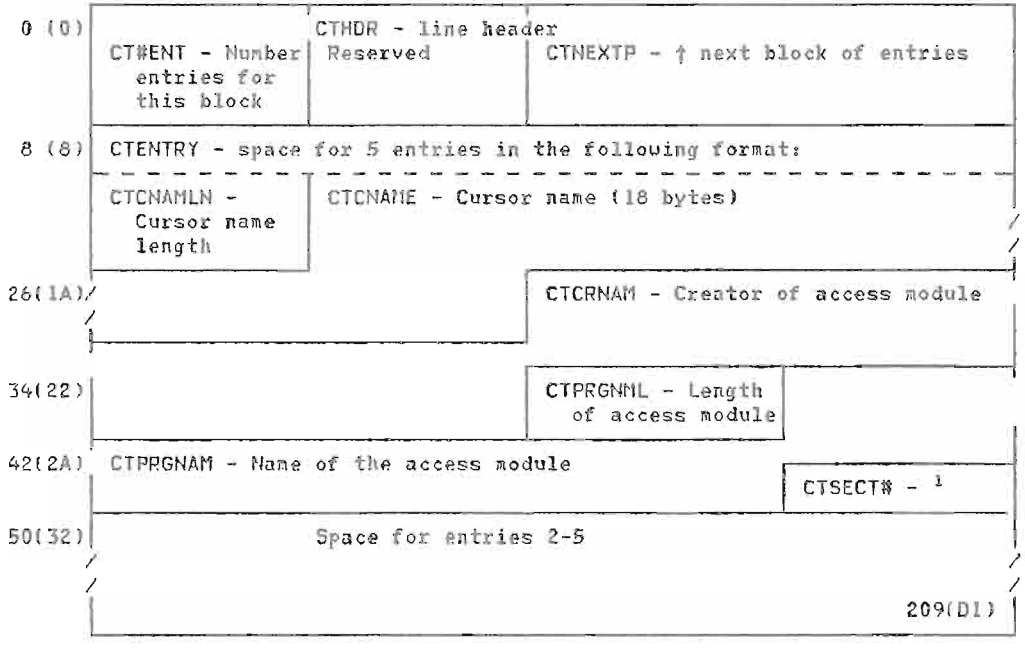
1 CTBFLAGS
CTBCKFET = '0': FETCH statement does not need INTO clause.
'1': FETCH statement needs INTO clause.
CTBDCLER = '0': No error on select.
'1': Error on select.

CURTABLE (EXECUTIVE CURSOR NAME TABLE)

An entry is made in this table as a result of an Extended DECLARE CURSOR. Extended cursor statements (OPEN, FETCH, CLOSE) within a logical unit of work may use only cursor names that are in this table (or that have been declared in the same logical unit of work).

The CURTABLE is pointed to by RDAREA (RDACTABP, offset X'94').

Dec(Hex) CURTABLE



DBSPACE ENTITY PAGE FORMAT (WITH STORED ROW FORMAT)

Rows in SQL/DS tables are stored in DBSPACE entity pages. The entity page contains:

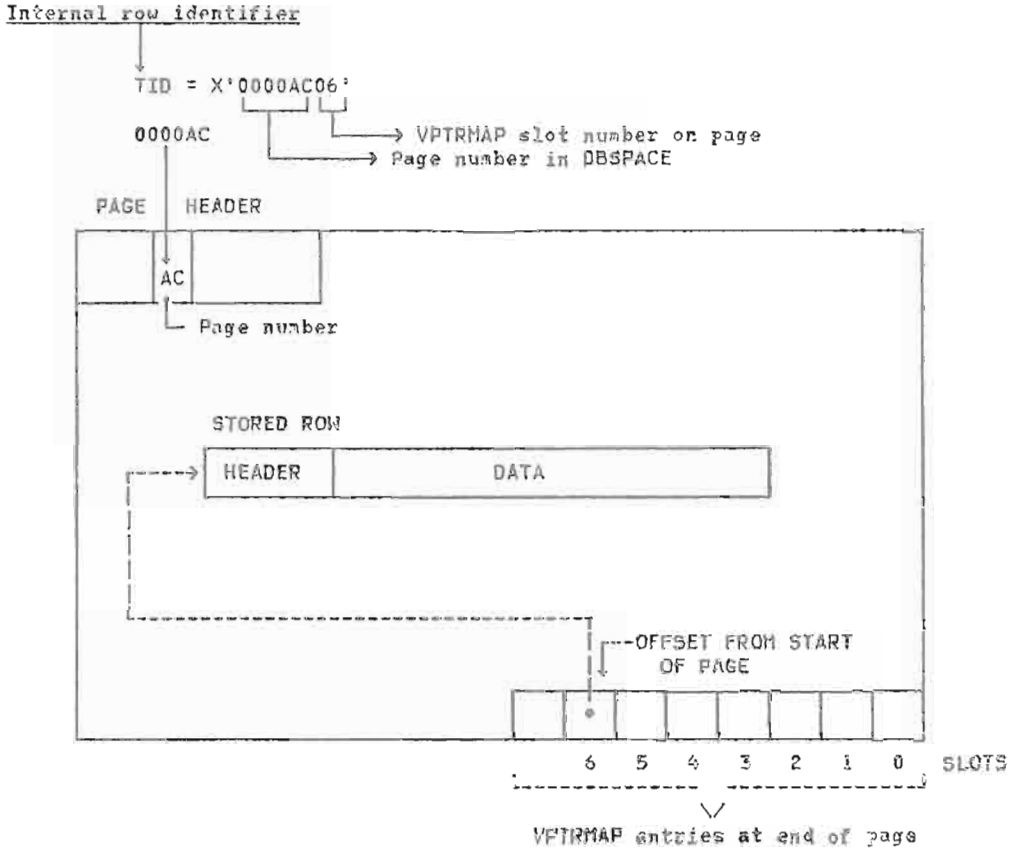
- A page header (VHEADER) used to identify the page and manage the space within the page.
- Zero to n rows stored in the page. These rows may belong to any SQL/DS table defined in DBSPACE.
- Zero to n "holes". Holes occur when:
  - A row is deleted from the page.
  - A row in the page is shortened because (a) a row update - including possible link removal - causes the row to become shorter, or (b) a row update causes the row to become larger, causing the row to be moved either to a new location on the page or to another page. In the latter case, the old row is shortened to a 'pointer to overflow' row with the excess space becoming a hole.

  Note that holes are interspersed with stored rows.
- Page contiguous free space (reflected in page header), which is candidate space for:
  - Newly inserted rows.
  - Relocation of existing rows on the page that became larger due to a row update.
  - Insertion of overflow rows from other entity pages in the DBSPACE. This occurs when a row update causes the row to become larger and there is insufficient contiguous free space on the original page to contain the longer row.

  Pages are compacted to remove holes and reclaim their space into contiguous free space when the page contains sufficient free space (including holes) to satisfy a row insert, but the contiguous free space is not large enough to contain the inserted row. The inserted row may be a new row or an updated row that had to be moved to a new page (an overflow row).
- Zero to n halfword slots (VPTRMAP entries) at the end of the page. These slots contain the displacement within the page of a stored row. When a row is first created (Insert Row operation), it is assigned a permanent (cannot change until row is deleted) SQL/DS internal 4-byte identifier called a TID. The first three bytes identify the page within the DBSPACE, and the fourth byte identifies the VPTRMAP slot number in the page that contains the row's displacement within the page. The first row inserted gets slot 0, the second gets slot 1, etc. The 0 slot is the last halfword on the page, the 1 slot is the next-to-last halfword on the page, etc. When a row is deleted, its slot contains a zero displacement. The page header indicates how many VPTRMAP slots are currently allocated on the page. Slot number n's displacement can be computed by the formula 4094-(2*n).

The following diagram shows a DBSPACE entity page containing a stored row whose TID is '0000AC06'X. Note that the first three bytes identify the entity page and that the low-order byte gives the VPTRMAP slot entry in the page. This allows DBSS to locate the stored row displacement within the page.

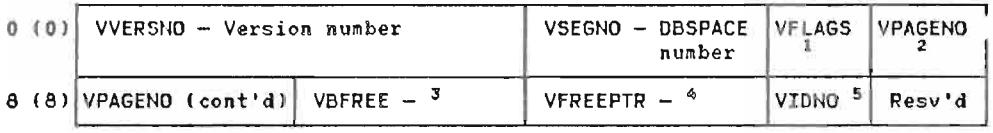
Entity Page Containing Stored Row (Example)



ENTITY PAGE HEADER (VHEADER)

The entity page header is described by the structure VHEADER below. The mapping macro is YVPAGE.

Dec(Hex) VHEADER



[1] VFLAGS Variable page flags:
- 0... .... Data page
- 1... .... Header page
- .0.. .... Entity page
- ..1. .... List page
- ...1 .... Intermediate page for sort object

[2] VPAGEND - Logical page number

[3] VBFREE - Total number of free bytes in page (including holes)

[4] VFREEPTR - Displacement to beginning of contiguous free space

[5] VIDNO - Maximum ID with entry in VPTRMAP (number of slots allocated)

Stored rows and holes (see entity page contents at beginning of this subsection) are described by the following structures. VTUPLE describes the stored row header, including the link pointer arrays (VTLAREA). Row data (columns) immediately follow the row header. Fixed-length columns are described by the simple structure FFDATA. Variable-length columns are described by the structure VFIELD and start with a 1-byte column length field. A table's columns are described to DBSS within the DBSPACE by the DOMAIN record (see CBASE structure MCR).

The header details are:

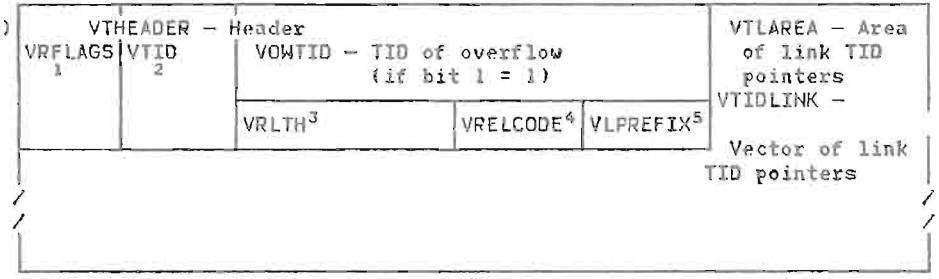
- If VTUPLE describes a small hole (less than four bytes), then VRFLAGS has the HOLE bit on, and SMLENGTH gives the hole length (no other bytes are meaningful)
- If VTUPLE describes a large hole (four or more bytes), then VRFLAGS has the HOLE bit on, and SMLENGTH bits are set to zero. VRLTH contains the length of the hole.
- If VTUPLE describes a pointer row (meaning that the row containing the data has been relocated to another page due to a row update that increased the row length), then VRFLAGS bit DATAFOLL is 0 and bit HOLE is 0, and Header field VOWTID contains the TID of the overflow row. A pointer row is six bytes long.
- VTUPLE describes a normal stored row if VRFLAGS DATAFOLL and HOLE are 0. If VRFLAGS NOOW is 1, it is an overflow stored row; otherwise, it is a base stored row.
- The owning table of a normal stored row is described in the Header by field VRELCODE, which contains the RID (internal identifier of the table-mapping to the DBSPACE header Master Control Record).
- Normal stored rows may be connected to one or more links. They can potentially be connected to any link defined for the owning table. Field VLPREFIX in the Header gives the length of the link pointer area (VTLAREA) in the stored row. VTALAREA contains, for each link, three 4-byte TID pointers (Left Twin pointer, Right Twin pointer, and Parent pointer). If the stored row is connected to a link, there will be a nonzero Left Twin pointer (unless the row is first child), a nonzero Right Twin pointer (unless the row is last child), and a nonzero Parent pointer unless the link is unary. For each link, the CBASE LCR structure contains the offset within the row VTLAREA of the link pointers.

STORED ROW/HOLE FORMAT (FROM MAPPING MACRO YVPAGE):

Stored Row Header (VTUPLE)

This form of the entity row consists of a header, a vector of (fullword) TID pointers for links, and a vector of domains (stored data fields).

Dec(Hex) VTUPLE

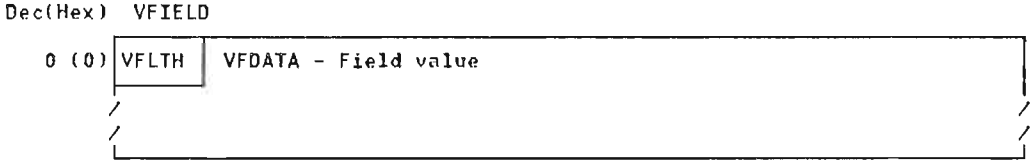


[1]

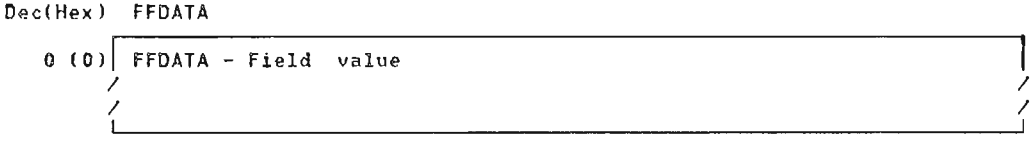
Offset	Field Name	Bits	Meanings
0 (0)	VRFLAGS		Record Flags:
	DATAFOLL	0...	Data follows
	NOOW	.0..	No overflow record
	HOLE	..1.	Hole (free space)
	LHOLE	...1	Locked hole (always zero if TID locking not used)
		.... 00..	Not used, but always zero
	SMLENGTH	.... ..nn	Length of small (=<3 bytes) hole following row pointer if original length=<9. Original TID is same as row pointer. '00'B if long hole (that is, length>=4, hence contains VTID, VRLNTH).

[2] VTID - Row TID or hole TID (if VRFLAGS SMLENGTH = 0) used in page (VPTRMAP slot number).
[3] VRLTH - Stored row or big hole length.
[4] VRELCODE - RID of owning table (if DATAFOLL=0).
[5] VLPREFIX - Size of VTLAREA in bytes.

Structure for Variable-Length Columns (VFIELD)

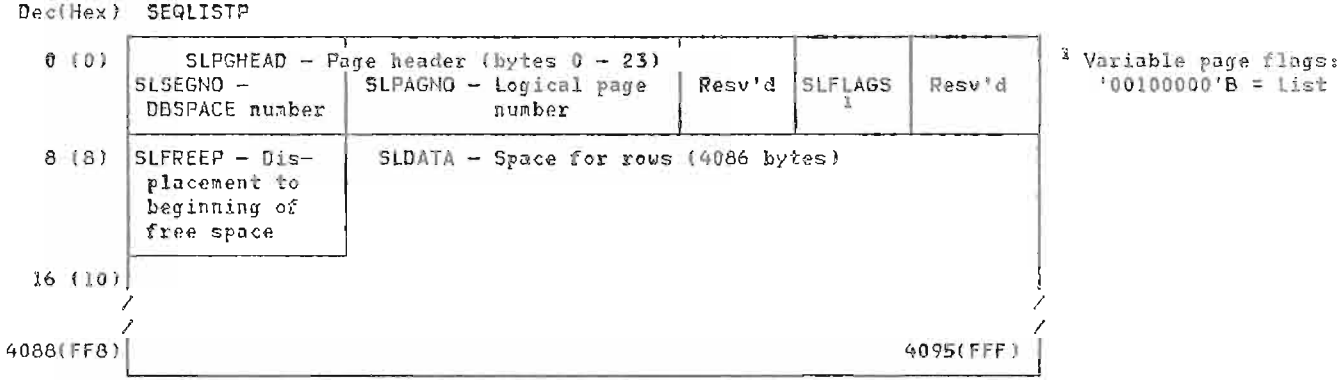


Structure for Fixed-Length Columns (FFDATA)

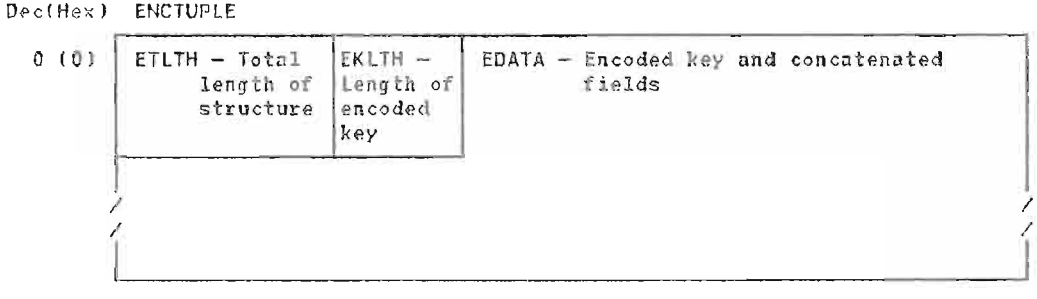


LIST PAGE FORMAT (OUTPUT OF DBSS SORT OPERATION)

The output of the DBSS Sort subcomponent is an object called a list. This list contains the ordered rows from the object that was sorted. The list is placed in a temporary DBSPACE in list pages. The format of the list page is:



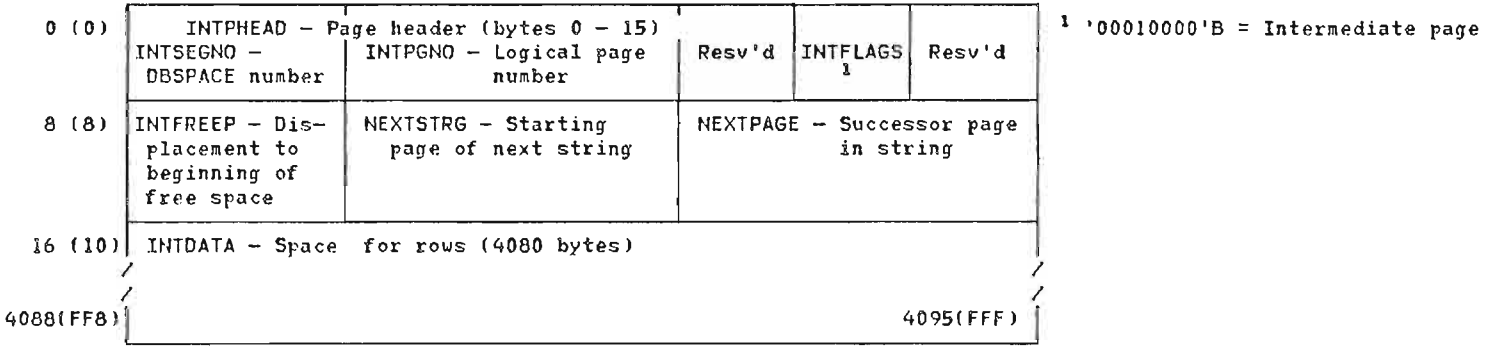
The sorted rows in a list page consist of the encoded key concatenated with the remaining domains that were being selected for the sort. The encoded key can contain from one to sixteen domains. This is referred to as an encoded row. The format of the encoded row is:



Lists can be retrieved only via DBSS Data Manipulation operations. First an OPEN SCAN must be performed on the List. Then a GET NEXT operation can be performed to sequentially retrieve the rows in the List. Note that a DBSPACE containing a List can contain only one List and can contain no tables.

INTERMEDIATE SORT PAGE FORMAT

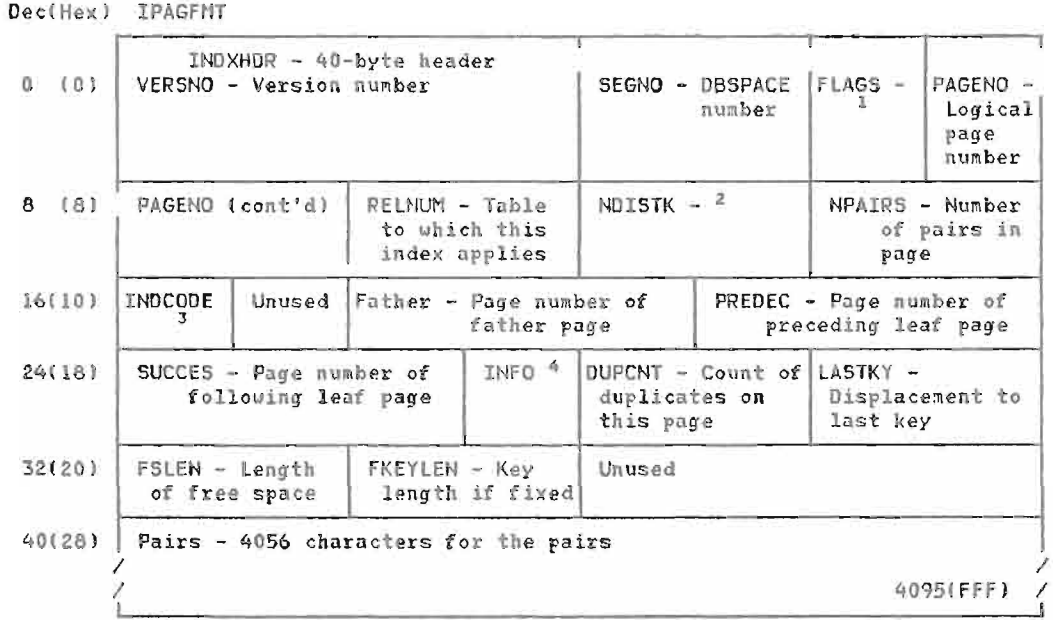
During a sort operation, the DBSS Sort subcomponent allocates intermediate DBSPACEs to be used as external sort work areas. The data in these DBSPACEs is kept in intermediate sort pages. These pages are used only by the Sort subcomponent and exist only for the duration of a particular sort operation. The only difference between the sorted list page and the intermediate page is the header. The intermediate data is in the form of encoded rows, the same as in a sort list page. The format of the intermediate sort page is:



DBSPACE INDEX PAGE FORMAT

This subsection consists of templates for an index page and the entries, called pairs, within the index page.

IPAGFMT - TEMPLATE FOR AN INDEX PAGE



[1] See "Template Flags" on page 46.

[2] For a leaf page: Number of clusters consisting of a key and a number (<256) of TIDs.
For a non-leaf page: Number of descendents minus one.

[3] Index code: Ordinal number of the index.

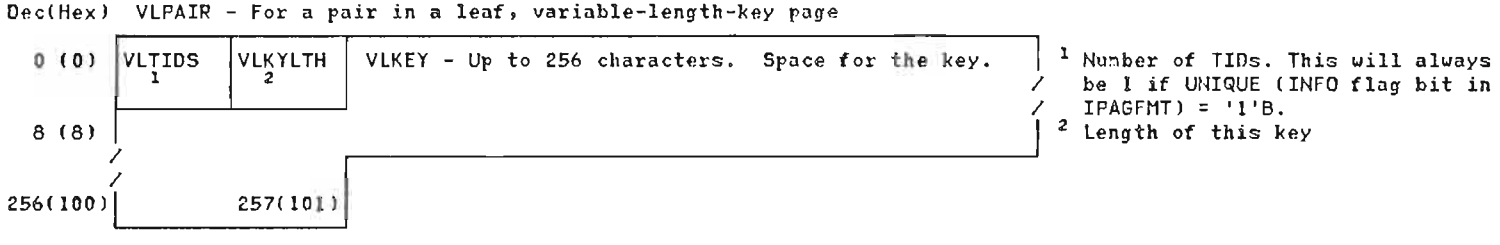
[4] See "Template Flags" on page 46.

Template Flags

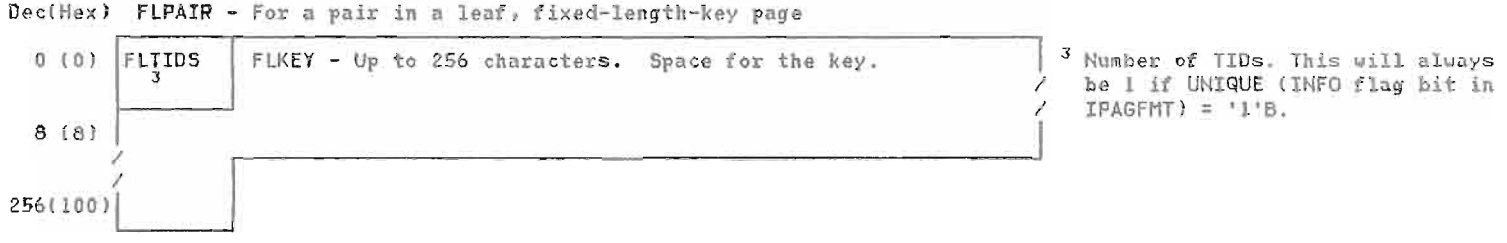
Offset	Field Name	Bits	Meanings
6 (6)	FLAGS		Page flags:
	HEADPAGE	0...	Data page
	ENTPAGE	.1..	Index page
		..xx xxxx	Reserved
27 (1B)	INFO		
	LEAF	1...	Leaf
	ROOT	.1..	Root
	VARIABL	..1.	Variable length keys
	UNIQUE	...1	Unique keys
	NPLEAF	.... 1...	Child pages are leaves (If LEAF=1, then NPLEAF must be 0)
		xxx	Reserved

TEMPLATES FOR VARIOUS TYPES OF ENTRIES (PAIRS) FOUND IN INDEX PAGES

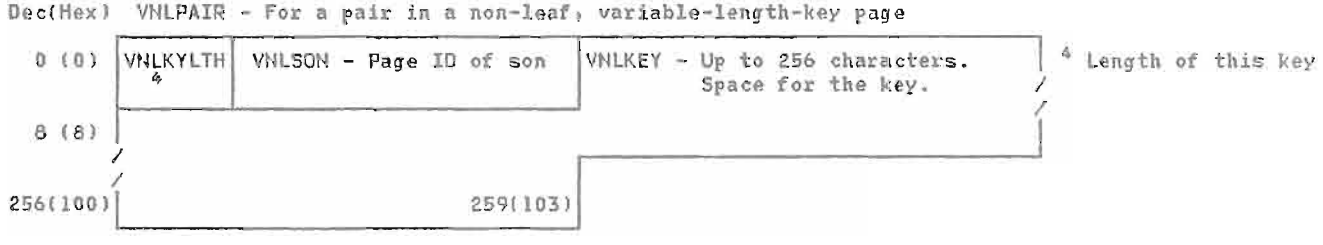
VLPAIR (Pair in a Leaf, Variable-Length-Key Page)



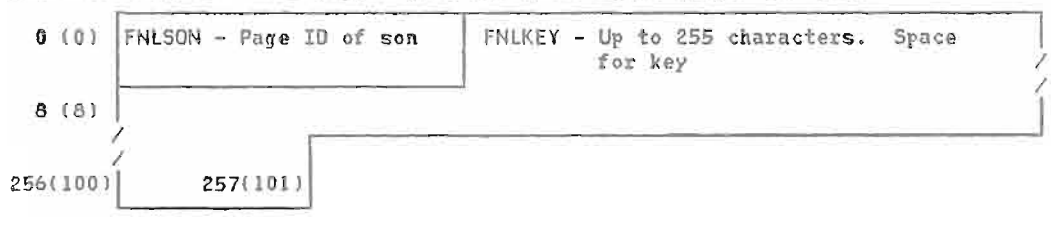
FLPAIR (Pair in a Leaf, Fixed-Length-Key Page)



VNLPAIR (Pair in a Non-Leaf, Variable-Length-Key Page)



FNLPAIR (Pair in a Non-Leaf, Fixed-Length-Key Page)



DCE (DISPATCHER CONTROL ELEMENT)

This table is used to control the dispatching of each SQL/DS agent structure. It contains pointers to the SQL/DS control blocks used for the agent structures plus a pointer to the SQL/DS global DS2CVT control block. When an agent calls the Dispatcher to wait for completion of an event, its registers are stored in the DCE (DCEREGS). It also contains flags used to indicate the state of the associated agent.

The DCE is pointed to by:

DSCAREA	(DSCDCEP,	offset X'14')	[1]
YTABLE1	(YT1DCEP,	offset X'18')	[1]
RDAREA	(RDADCEP,	offset X'18')	[1]
DS2CVT	(DS2DCECH,	offset X'1C')	[2]
DCE	(DCENEXT,	offset X'20')	[3]
DS2CVT	(DS2CRDCE,	offset X'20')	[4]

For the following notes, also refer to the diagrams on pages 2 and 3.

[1] These pointers are only within an agent structure. That is, each of these control blocks is directly associated only with other control blocks within the same agent structure.
[2] This points only to the first in the chain of DCEs (that is, to the DCE for the Operator Agent structure).
[3] This points to the next DCE in the chain (for the next agent structure).
[4] This points to the DCE of the currently "executing" agent structure.

Dec(Hex) DCE

Dec(Hex)				
0	DCEICTCH - 'DCE' (eyecatcher)			
8	DCELENTH - DCE length	DCEPWF [5]	DCELFLAG [5]	DCEINDEX - DCE index number
16(10)	DCEDS2P - ↑ DS2CVT	DCESRAP - ↑ DSCAREA (SRAREA)		
24(18)	DCEYT1P - ↑ YTABLE1	DCERDAP - ↑ RDAREA		
32(20)	DCENEXT - ↑ next DCE in chain	DCEWTMPP - ↑ DCE's part of wait-multiple list (to I/O entry)		
40(28)	DCERECBP - ↑ DCE's RECEIVE ECB in the CPC control block (VSE) or VMCBLOCK (VM).	DCEDMECB - Dummy ECB for wait-multiple list		
48(30)	DCEDREGS - Dispatcher register save area (16 fullwords) (saved in order 0 - 15)			
112(70)	DCENXTAR - Address of next available real agent (VM only)	DCENXTIU - Address of next in-use real agent (VM only)		
120(78)	DCEFRVIU - Address of previous in-use real agent (VM only)	DCEADRPA - Address of associated pseudo-agent (VM only)		
128(80)	Reserved (4 fullwords)			143(8F)

[5] See "DCE Flags" on page 49.

DCE Flags

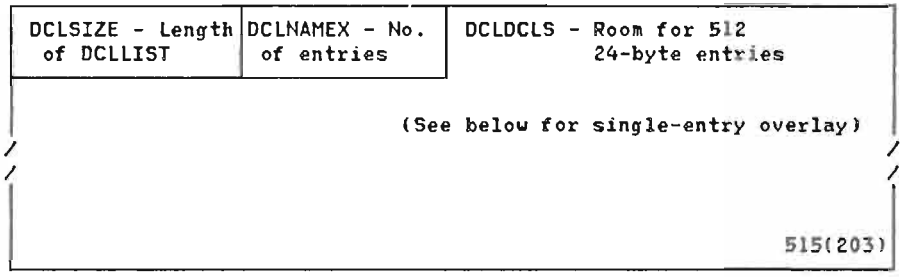
Offset	Field Name	Bits	Meanings
12(0C)	DCEPWF (Processor WAIT flag)		
	DCEGENRL	1...	Waiting for general post.
	DCEIOWAT	.1..	Request for I/O wait.
	DCELOCK	..1.	Waiting for lock post.
	DCEXPTN	...1	Waiting for communication post.
	DCEINACT	.... 1...	DCE inactive indicator.
		xxx	Reserved WAIT flags.
13(0D)	DCELFLAG (Communication link flag byte)		
	DCECNCT	1...	Cross-partition link connected.
	DCELKTRM	.1..	Link down (or disconnected).
	DCEFORCE	..1.	SQL/DS task FORCE started.
	DCERESET	...1	Communication link reset (cleared).
	DCESQLAB	.... 1...	SQL/DS severed communication link (VM only)
		xxx	Reserved communication link flags.

DCLLIST (DECLARE LIST)

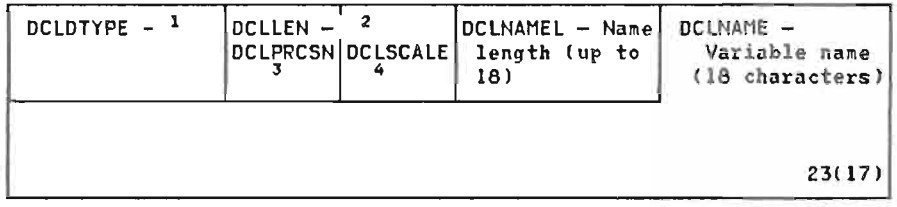
ARIBDCL is the mapping macro for the Declare List, which contains information about all host variables found in the application program. It is passed to SQL/DS by preprocessors via the RDIIN (RDIAUXPA) for a SQLCALL. The RDS Executives use it to determine host variable type information, given the list of I/O variables built by the Parser (IVNAMES/OVNAMES) for the current SQL statement. This type information is stored in the LITNODEs of the Op Tree by ARIXEPP to be used by the Code Generator in determining the proper execute-time code sequences for storing into or retrieval from host variables. (The NLST (Name List) structure is used by ARIXEPP to store the matches between the IVNAMES/OVNAMES (Parser) and the DCLLIST (PREP).

A pointer to the DCLLIST is kept in field PPODCLPT of PPOPGNST and field PDCLPTR of PREPDSCB by the RDS Executives.

Dec(Hex) DCLLIST



Overlay of one entry in DCLDCLS



1 See data type, page 151,
2 Data length
3 Decimal precision
4 Decimal scale

DSCAREA (SQL/DS CONTROL AREA)

This data area is used as a general control block for the DSC component. It contains:

- Pointers to the DS2CVT and the other agent structure control blocks.
- Control information used by the SQL/DS Operator Communication routine, by the Working Storage Manager routines, by Cross-Partition Communication or Inter-User Communication Vehicle routines, and by the Dispatcher.
- SQL/DS tracking information used by the SQL/DS abend handler to determine where an error occurred in SQL/DS.

The DSCAREA is pointed to by:

```
   DCE     (DCESRAP,  offset X'14') ┐
   YTABLE1 (YT1SRAP,  offset X'14') ├ 1
   RDAREA  (RDADSCA,  offset X'14') ┘
   DS2CVT  (DS2DSCCH, offset X'18') 2
   DSCAREA (DSCNEXT,  offset X'20') 3
```

For the following notes, also refer to the diagrams on pages 2 and 3.

[1] These pointers are only within an agent structure. That is, each of these control blocks is directly associated only with other control blocks within the same agent structure.

[2] This points only to the first in the chain of DSCAREAs (that is, to the DSCAREA for the Operator Agent structure).

[3] This points to the next DSCAREA in the chain (for the next agent structure).

Alphabetic List of Field Names

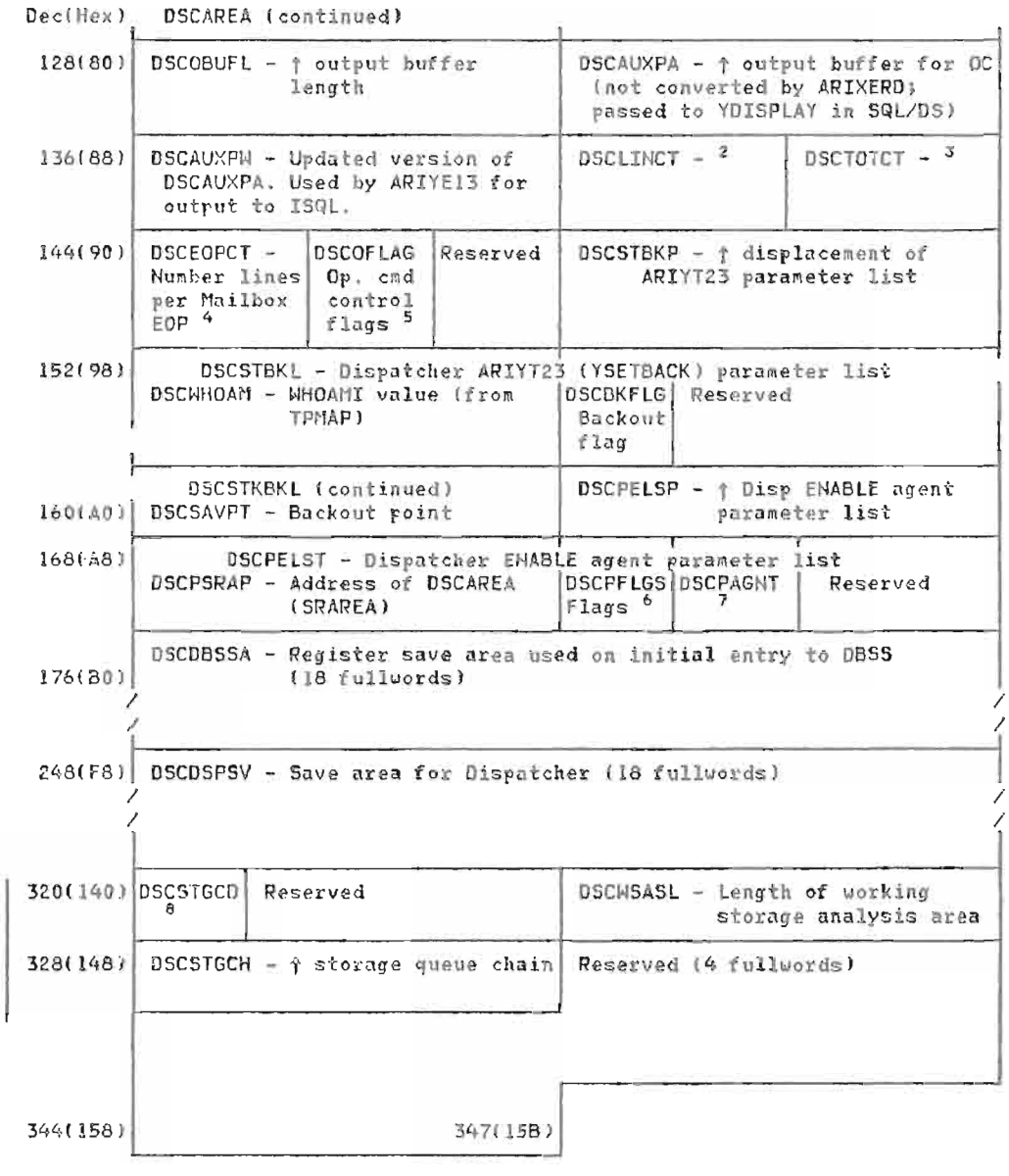
DSCALEN	8(8)	DSCHLEN	48(30)	DSCPSRAP	168(A8)
DSCAUXPA	132(84)	DSCIBUFL	120(78)	DSCRDAP	28(1C)
DSCAUXPW	136(88)	DSCIBUFP	116(74)	DSCSAVPT	160(A0)
DSCBKFLG	156(9C)	DSCLINCT	140(8C)	DSCSTBKL	152(98)
DSCCATCH	0(0)	DSCLNKNM	80(50)	DSCSTBKP	148(94)
DSCCNPLP	96(60)	DSCLNKNM1	80(50)	DSCSTGCD	320(140)
DSCCNBLK	100(64)	DSCLNKNM2	88(58)	DSCSTGCH	328(148)
DSCCVTP	16(10)	DSCNEXT	32(20)	DSCTOTCT	142(8E)
DSCDBSSA	176(B0)	DSCOBUFL	128(80)	DSCTRACK	13(D)
DSCDBUFL	112(70)	DSCOBUFP	124(7C)	DSCUSRID	72(48)
DSCDBUFP	108(6C)	DSCOFCH	52(34)	DSCVMCBP	100(64)
DSCDCEP	20(14)	DSCOFLAG	146(92)	DSCWHOAM	152(98)
DSCDSPSV	248(F8)	DSCOFLEN	56(38)	DSCWSACP	68(44)
DSCENTRY	36(24)	DSCOMBPP	104(68)	DSCWSASL	324(144)
DSCEOPCT	144(90)	DSCPAGNT	173(AD)	DSCWSASP	64(40)
DSCFLAGS	12(C)	DSCPELSP	164(A4)	DSCWSHD	40(28)
DSCFQEIP	40(28)	DSCPELST	168(A8)	DSCYTIP	24(18)
DSCHADDR	44(2C)	DSCPFLGS	172(AC)		

Dec(Hex) DSCAREA

0	DSCCATCH - "DSCAREA" (eyecatcher)			
8	DSCALEN - DSCAREA length	DSCFLAGS Flags [1]	DSCTRACK Tracking info.	Reserved
16(10)	DSCCVTP - ↑ DS2CVT	DSCDCEP - ↑ DCE for this agent structure.		
24(18)	DSCYTIP - ↑ YTABLE1 for this agent.	DSCRDAP - ↑ RDAREA for this agent		
32(20)	DSCNEXT - ↑ next DSCAREA	DSCENTRY - Entry address for this agent.		
40(28)	DSCWSHD - Header for Working Storage Pool DSCFQEIP - ↑ first FQE	DSCHADDR - Address of working storage header.		
48(30)	DSCWSHD (continued) DSCHLEN - Length of working storage pool.	DSCOFCH - ↑ next working storage pool in chain.		
56(38)	DSCWSHD (continued) DSCOFLEN - Length of next pool in chain.	DSCHDPTR - ↑ start of working storage pool.		
64(40)	DSCWSASP - ↑ start of working storage analysis area	DSCWSACP - ↑ current element in working storage analysis area		
72(48)	DSCUSRID - User ID associated with this agent.			
80(50)	DSCLNKNM DSCLNKNM1 - SQL/DS name			
88(58)	DSCLNKNM (continued) DSCLNKNM2 - User name			
96(60)	DSCCMPLP - ↑ Communication Manager parameter list	DSCCNBLK - ↑ Communication Manager control block (for VSE) DSCVMCBP - ↑ VMCBLOCK for agent (for VM)		
104(68)	DSCOMBPP - ↑ Output Mailbox parameters	DSCDBUFP - ↑ default input buffer		
112(70)	DSCDBUFL - ↑ default buffer length	DSCIBUFP - ↑ input message buffer		
120(78)	DSCIBUFL - ↑ input buffer length	DSCOBUFP - ↑ output message buffer		

Working Storage > Manager Header

[1] See "Flags" on page 54.



2 Current line number when ARIYE13 is sending mail output to ISQL.
3 Total number lines sent by ARIYE13 (used only if OCIMODE = 0).
4 Used only if OCIMODE = 0
5 See "Flags" page 54.
6 See "Flags" page 54.
7 Agent type indicator.
8 Operating system return code (storage error)

DSCAREA Flags

Offset	Field Name	Bits	Meanings
12(0C)	DSCFLAGS		
	DSCROPER	1...	Error during ISQL operator message processing
	DSCINPRC	.0..	FORCE forward allowed
		.1..	FORCE forward not allowed
	DSCACPRC	..0.	Will not accept return code
		..1.	Will accept return code
	DSCLIMER	...1	Limit error occurred
	DSCHRDER	.... 1...	Hardware error occurred
	DSCSCHED	.... .0..	SQL CONNECT required
		.... .1..	SQL CONNECT not required
	DSCWSASW	.... ..0.	LUW not started for agent
		.... ..1.	The first LUW was started
	DSCELUW	0	LUW still active
		1	LUW has terminated
146(92)	DSCOFLAG		
	DSCOMODE	1...	ISQL mode
		0...	Local mode (system console) Initialized to 1 for operator agent structure; initialized to 0 for user agent structure
	DSCORSET	.1..	Reset Mailbox line counter in ARIYE13
		.0..	No reset Initialized to zero (used only if OCIMODE = 0).
	DSCOSLCT	..x.	OCIMODE override in ARIYE13. If ISQL linkage, send message to ISQL if on, else to system console: 1 = write message to ISQL linkage 0 = write message to system console Initialize to 0, check only if OCIMODE = 0
	DSCEOPSW	...1	End-of-process switch used by ARIYM10 (operator command linkage). Tells ARIYE13 to "send" to ISQL even if the mail buffer is not full
		 xxxx	Reserved.
172(AC)	DSCPFLGS		
	DSCPENLK	1...	Enable communication link
	DSCPENRD	.1..	Enable and use RDAREA
	DSCPDSPA	..1.	Return pseudo-agent to available queue
	DSCINACT	...1	Real agent inactive (not dispatchable)
	DSCIGSTK	.... 1...	Do not attempt to free working storage
	DSCSQLAB	.... .1..	SQL/DS severed communication link (VM only)
		xx	Reserved

DS2CVT (SQL/DS COMMUNICATION VECTOR TABLE)

The DS2CVT contains pointers to the base global tables for the SQL/DS subcomponents (DBSS and RDS), the local agent structure tables for the DSC subcomponent, to various SQL/DS routines called from outside their subcomponent, and general SQL/DS global control information (primarily used by the DSC subcomponent).

The DS2CVT is pointed to by:
- DS2MODE (DS2MCVTP, offset X'14')
- YRSSCVT (YRSDSCVP, offset X'10')
- RDCVT (RDCDS2CV, offset X'10')
- DS2CVT (DS2SELFP, offset X'C4') [1]
- DCE (DCEDS2P, offset X'10')
- DSCAREA (DSCCVTPE, offset X'10')

For the following note, also refer to the diagrams on pages 2 and 3.

[1] This pointer is set up for the SQL/DS Abend Handler (ARICABE). It is located in the four bytes preceding the Abend Savearea (pointed to be register 1) and is used to get the start of the DS2CVT on abend or program check conditions.

Alphabetic List of Field Names

DS2ABSA	200(C8)	DS2ICTCH	0(0)	DS2RSTPP	116(74)
DS2ABSAP	200(C8)	DS2IDGBP	56(38)	DS2SELFP	196(C4)
DS2ABSAR	208(D0)	DS2IDIBP	64(40)	DS2SERCD	392(188)
DS2ABTRM	402(192) bit flag	DS2IDRBP	60(3C)	DS2SERFL	190(BE)
DS2AGTLP	68(44)	DS2IOFDP	48(30)	DS2SHOP	416(1A0)
DS2BRMEP	164(A4)	DS2LENTH	8(8)	DS2SHTDN	190(BE)
DS2BRMSP	160(A0)	DS2MUDP	104(68)	DS2STBKP	120(78)
DS2CECBP	64(40)	DS2NEWSM	351(15F)	DS2SYSBP	124(7C)
DS2CNBLP	404(194)	DS2OMBP	108(6C)	DS2SYSER	128(80)
DS2CRDCE	32(20)	DS2OPCSA	272(110)	DS2SYSM	13(D)
DS2CVMDP	72(48)	DS2OPSYS	348(15C)	DS2TRCUU	193(C1)
DS2DBSEP	148(94)	DS2PARMP	76(4C)	DS2TRDNR	188(BC)
DS2DBSLE	380(17C)	DS2PDMPP	132(84)	DS2TRFDB	184(B8)
DS2DBSLS	376(178)	DS2PRTEP	140(8C)	DS2TRFLG	190(BE)
DS2DBSSP	144(90)	DS2PRTSP	136(88)	DS2TRLCK	191(BF)
DS2DCECH	28(1C)	DS2RDCVP	20(14)	DS2TRMFL	402(192)
DS2DCELN	54(36)	DS2RDSEP	156(9C)	DS2TRUID	176(B0)
DS2DISPP	92(5C)	DS2RDSLE	388(184)	DS2USAG1	52(34)
DS2DOSMD	349(15D)	DS2RDSLS	384(180)	DS2USERC	190(BE)
DS2DSCCH	24(18)	DS2RDSSP	152(98)	DS2USREP	172(AC)
DS2DSPDV	14(E)	DS2REG14	80(50)	DS2USRSP	168(A8)
DS2DUMPT	15(F)	DS2RESCD	84(54)	DS2VMCBP	56(38)
DS2DWATP	96(60)	DS2RETCD	88(58)	DS2VMID	352(160)
DS2DWTP	100(64)	DS2RETRG	412(19C)	DS2VMQHP	60(3C)
DS2EVEND	44(2C)	DS2RETRM	402(192) bit flag	DS2VMSIZ	360(168)
DS2EVLST	40(28)	DS2RETSA	408(198)	DS2VMSSM	350(15E)
DS2EVTBL	36(24)	DS2RNDMS	12(0C)	DS2VMUEP	368(170)
DS2FLAGS	12(C)	DS2RMSGP	112(70)	DS2VMWMP	364(16C)
DS2GETVP	344(158)	DS2RSCVP	16(10)	DS2XERIP	372(174)

Dec(Hex) DS2CVT

0(0)	DS2ICTCH - "DS2CVT" (eyecatcher)				
8(8)	DS2LENTH - DS2CVT length	DS2FLAGS - Misc SQL/DS flags [1]	DS2SYSM [2]	DS2DSPDV [3]	DS2DUMPT [4]
16(10)	DS2RSCVP - ↑ YRSSCVT	DS2RDCVP - ↑ RDCVT			
24(18)	DS2DSCCH- ↑ DSCAREA chain (SRAREA)	DS2DCECH - ↑ DCE chain			
32(20)	DS2CRDCE- Address of running DCE	DS2EVTBL - Address of multiple wait list			
40(28)	DS2EVLST - Address of last ECB processed in table	DS2EVEND - End of table indicator			
48(30)	DS2IOFDP - ↑ SQL/DS I/O FDESC block	DS2USAG1 - Number of first user agent	DS2DCELN - Length of DCE (for Dispatcher)		
56(38)	DS2IDGBP - ↑ general CPC control block (for VSE) DS2VMCBP - ↑ VMCBLOCK for IUCV logon (for VM)	DS2IDRBP - ↑ Recovery CPC control block (for VSE) DS2VMQHP - ↑ Communication Block Queue Head (for VM)			
64(40)	DS2IDIBP - ↑ In-doubt CPC control block (for VSE) DS2CECBP - ↑ console ECB (for VM)	DS2AGTLP - ↑ Agent parameter list			
72(48)	DS2CVMDP - Address of DS2MODE	DS2PARMP - ↑ User parameters			
80(50)	DS2REG14 - Return address to operating system	DS2RESCD - Reason code passback			
88(58)	DS2RETCD - Return code passback	DS2DISPP - ↑ ARICDSP (Dispatcher)			
96(60)	DS2DWATP - Address of Dispatcher internal wait	DS2DWTP - Address of ARICDWT (DSC WAIT routine)			
104(68)	DS2MUDP - Address of Cross Partition receive routine	DS2OMBP - Address of out Mailbox procedure routine			

[1] See "DS2CVT Flags" on page 59.
[2] Single/Multiple partition or Single/Multiple Virtual Machine mode indicator (i.e. S or M)
[3] SQL/DS-directed output indicator (from DSPLYDEV parameter):
  For VSE:
   C = Operator console
   L = Printer (default)
   B = Both
  For VM:
   C = Operator console
[4] SQL/DS ABEND dump type (from DUMPTYPE parameter):
  N = None
  P = Partial
  F = Full (default)

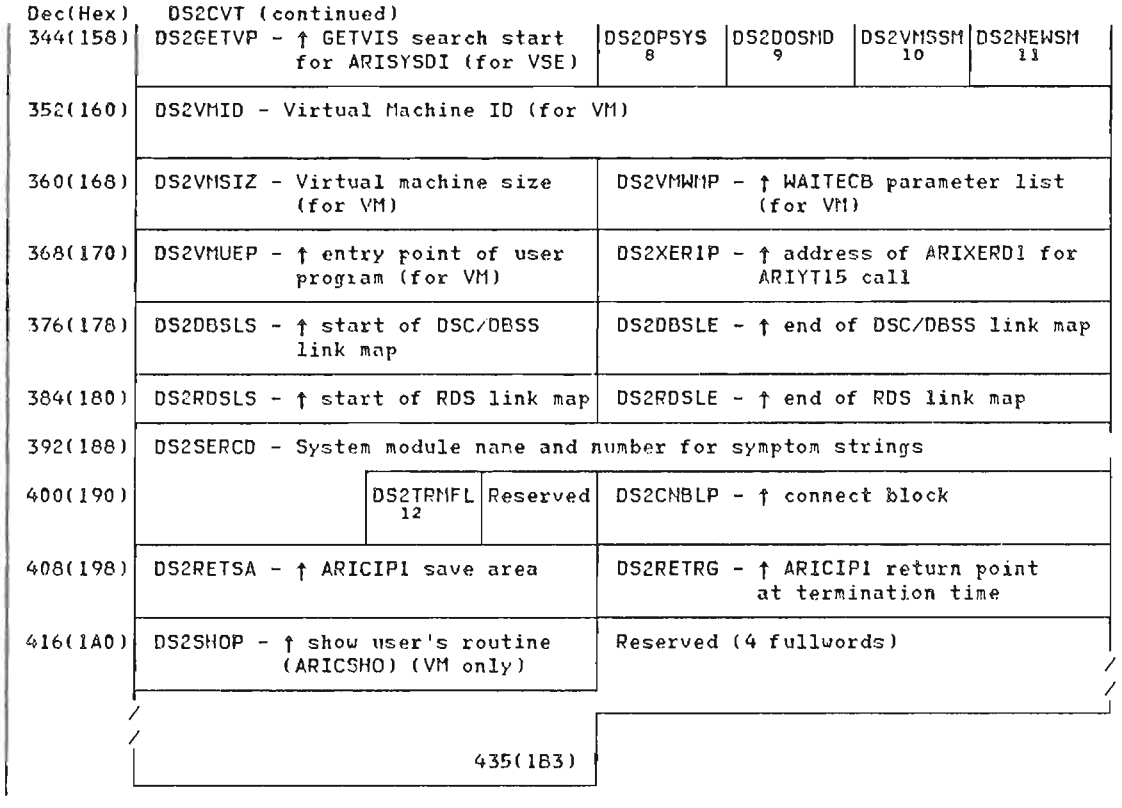
Dec(Hex) DS2CVT (continued)

112(70)	DS2RMSGP - Address of message services routine		DS2RSTPP - Address of reset process routine	
120(78)	DS2STBKP - Address of ARIYT23 (YSETBACK)		DS2SYSBP - Address of ARIYT15 (YSYSBACK)	
128(80)	DS2SYSER - Address of ARIYM02 (YSYSTERR)		DS2PDMPP - Address of snap dump routine	
136(88)	DS2PRTSP - † Start of SQL/DS partition or virtual machine		DS2PRTEP - † End of SQL/DS partition or virtual machine	
144(90)	DS2DBSSP - † Start of DBSS[5]		DS2DBSEP - † End of DBSS	
152(98)	DS2RDSSP - † Start of RDS		DS2RDSEP - † End of RDS	
160(A0)	DS2BRMSP - † Start of Batch or VM Resource Manager		DS2BRMEP - † End of Batch or VM Resource Manager	
168(A8)	DS2USRSP - † Start of user phase or load module		DS2USREP - † end of user phase or load module	
176(B0)	DS2TRUID - userid to be traced			
184(B8)	DS2TRFDB - † Trace File Descriptor Block for ARISYSD5	DS2TRDNR - Trace point number selected for partition or virtual machine snap dump	DS2TRFLG Flags for SQL/DS trace [6]	DS2TRLCK Trace 'lock' byte
192(C0)	Reserved	DS2TRCUU - cuu of trace output file (tape)	DS2SELFP - † DS2CVT (used by ARICABE)[7]	
200(C8)	DS2ABSA - ABEND save area DS2ABSAP - ABEND save area PSW			
	DS2ABSA (continued)			
208(D0)	DS2ABSAR - ABEND save area registers (16 fullwords - for VSE)			
	/ Reserved for VM /			
272(110)	DS2OPCSA - Operator communication savearea (18 fullwords - for VSE)			
	/ Reserved for VM /			

[5] ARICIP1 must be first module in the SQL/DS link book

[6] See "DS2CVT Flags" on page 59.

[7] This field must precede the SQL/DS ABEND savearea (DS2ABSA).



8 Operating system indicator - 'V' for VM, 'D' for VSE

9 Save area for VSE mode indicator (VM); Reserved (VSE)

10 System Mask setting (current) (VM); Reserved (VSE)

11 System Mask setting (new - no external interrupt) (VM); Reserved (VSE)

12 See "DS2CVT Flags" on page 59.

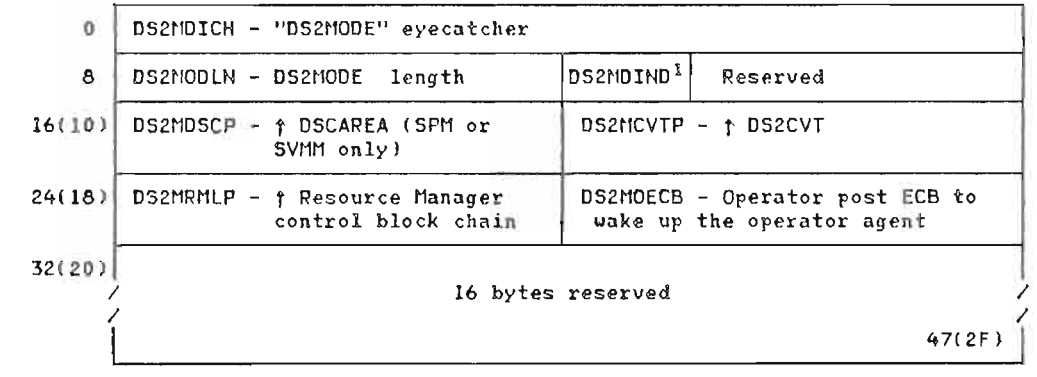
DS2CVT Flags

Offset	Field Name	Bits	Meanings
12(0C)	DS2FLAGS		
	DS2DCESC	1...	Tells Dispatcher to scan DCE chain
	DS2OACTV	.0..	Operator command processor inactive
		.1..	Operator command processor busy
	DS2WTOWT	..0.	WTO routine does implicit WAIT
		..1.	WTO routine does Dispatcher WAIT
	DS2TROPN	...0	Do not call Trace OPEN
		...1	Call Trace OPEN
	DS2DROUT	.... 0...	Output is directed to SYSTEM console (VSE)
		.... 1...	Output is directed as indicated in DS2DSPDV
	DS2RMDMS	.... .0..	Resource Manager not in DMSFREE area (SVMM)
		.... .1..	Resource Manager is in DMSFREE area (SVMM)
		xx	Reserved
190(BE)	DS2TRFLG		
	DS2TRFLA	x...	Trace active? 1 = YES; 0 = NO
	DS2TRFLU	.x..	Trace by user ID? 1 = YES; 0 = NO
	DS2TRFLD	..x.	Trace point snap dump requested? 1 = YES; 0 = NO
	DS2TRFLE	...x	Trace output disabled due to SQL/DS initialization for trace tape mounting failure or due to trace output access error? 1 = YES; 0 = NO
	DS2TRFLI	.... x...	Trace off and in closed state; DS2TRFDB points to non-open File Descriptor Block? 1 = YES; 0 = NO
	DS2TRFLS	.... .1..	Suppress working storage trace
		xx	Reserved
402(192)	DS2TRMFL		
	DS2SHTDN	0...	Shutdown has not been called
		1...	Shutdown (ARICSHT) has been called
	DS2USERC	.0..	User RC=0 or SQL/DS RC
		.1..	User return code specified
	DS2SERFL	..0.	ARICABE can clear ABEND exit
		..1.	ARIYM02 (YSYSTERR) called the ABEND routine (ARICABE), do not clear abend exit
	DS2RETRM		Termination recursion flag
		...0	Termination not started
		...1	Termination in progress
		 xxxx	Reserved

DS2MODE

DS2MODE is loaded at initialization time and the 'DS2MDIND' field, which has been set to indicate multi-partition, or multiple virtual machine mode (MPM/MVMM), is reset to indicate single-partition, or single virtual machine mode (SPM/SVMM) if that is the execution environment for SQL/DS. The Batch Resource Manager, when invoked, will load the DS2MODE control block and test the DS2MDIND field. In SPM/SVMM, it will find DS2MDIND to be an 'S'. In MPM/MVMM, the first load will bring a new copy of the control block into storage and DS2MDIND will contain 'M'. There is also a pointer to locate the DS2CVT (DS2MCVTP - only in the SQL/DS partition/virtual machine) and a pointer to locate the DSCAREA (DS2MDSCP - SPM/SVMM only). There is also a pointer to the Resource Manager control block chain (DS2MRMLP) that is set only in the application partition/virtual machine.

Dec(Hex) DS2MODE



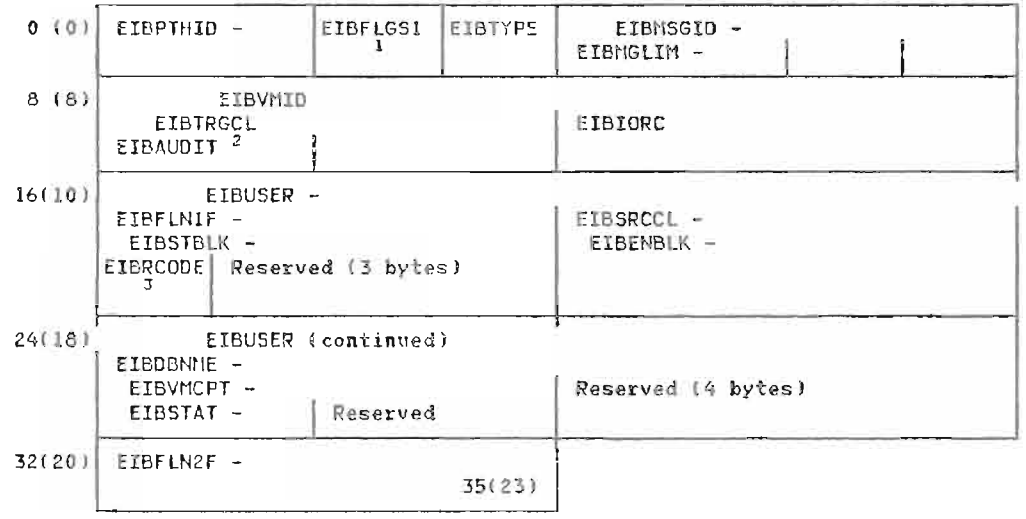
[1] SQL/DS mode indicator:
'S' = SPM/SVMM
'M' = MPM/MVMM

EIB (EXTERNAL INTERRUPT BUFFER)

The EIB is a SQL/DS mapping of the IPARML VM control block, and only the fields used by SQL/DS are shown here. ARICEIB is the mapping macro for the EIB.

The EIB is pointed to by register 2 on entry to the SQL/DS external interrupt handler.

Dec(Hex) EIB



1 EIBFLGS1

EIB1WAY	...1	One-way message.
EIBQUSCE	.... .1..	Path has been quiesced.
EIBPRTY	.... ..1.	Priority message.
	xxx. x..x	Not used by SQL/DS

2 EIBAUDIT Byte 1

EIBRPLE	1...	Reply too long for buffer
EIBSNPX	.1..	Protection exception on send buffer
EIBSNAX	..1.	Addressing exception on send buffer
EIBANPX	...1	Protection exception on answer buffer
EIBANAX	.... 1...	Addressing exception on answer buffer
EIBRJCT	.... .1..	Message was rejected
	xx	Reserved

Byte 2

EIBRCPX	1...	Protection exception on receive buffer
EIBRCAX	.1..	Addressing exception on receive buffer
EIBRPPX	..1.	Protection exception on reply buffer
EIBRPAX	...1	Addressing exception on reply buffer
EIBSEVR	.... 1...	Path was severed
	xxx	Reserved

FDESC (FILE DESCRIPTOR BLOCK)

This data area is used to pass information describing I/O requests from the caller to the system-dependent routines for I/O functions (ARISYSD5 and ARISD00D) and back to the caller.

For an I/O request, the caller calls ARISYSD5 (entry point in module ARISYSDD for VSE/Advanced Functions or entry point in ARISYSEC for VM/SP) and passes a pointer to his FDESC. If it is an 'OPEN' request, ARISYSD5 obtains automatic storage for the module (size = @SIZDATD), which is kept until 'CLOSE' processing is complete. The address of this storage is kept in the FDESC (field SYSDDWA). All I/O requests after 'OPEN' should refer to the original FDESC. Bytes 0 through 39 are in the caller's area. All necessary I/O information is stored within this area. The remainder of the data area is a save area for ARISYSD5 and (VSE only) the generalized I/O module ARISD00D.

Dec(Hex) ARIFDESC

0(0)	FDESCID - Control Block ID 'ARIFDESC'				
8(8)	FDESLEN - Control block length		DEVTYPE - Device type[1]	RECFM - Record format[2]	FLAG1[3]
16(10)	LRECL - Record length		BLKSIZE - Block size (I/O area)		
24(18)	CUUNIT - Tape cuu	EREASON - Error reason code	WORKAR - Address of DTF for VSE; DCB or FSCB for VM/SP		
32(20)	RECADDR - Record address in error (VSE only)		FLAG2[4]	Reserved	
40(28)	SYSDDWA - ↑ automatic storage for ARISYSDD (VSE) or ARISYSEC (VM/SP)		LINKADR1 - Address of general I/O module (ARISD00D) (VSE only)		
48(30)	LINKADR2 - Address of single I/O module (VSE only)		LINKADR3 - Address of I/O trace print module (VSE only)		

[1] Values:
'RD' = card reader/virtual reader
'PU' = card punch/virtual punch
'PR' = printer/virtual printer
'TA' = tape
'DI' = DASD
'TE' = terminal (VM/SP)
'DU' = dummy (VM/SP)
'UK' = unknown (VM/SP)

[2] Values:
x'40' = V
x'48' = VS
x'50' = VB
x'58' = VBS
x'80' = F
x'90' = FB
x'C0' = U

[3] Flag byte 1 (see "FDESC Flags" on page 63)

[4] Flag byte 2 (see "FDESC Flags" on page 63)

Dec(Hex)	FDB (continued)							
56(38)	WORKADDR - Address of automatic storage for ARISD00D (VSE only)				IOREG2 - DTF alternate I/O register (VSE only)			
64(40)	OPENCNT [5]	CLOSECNT [6]	LREQSWT [7]	TRCSWIT [8]	INITSW1 [9]	INITSW2 [10]	SSLSWIT [11]	Reserved
72(48)	SSLNAME - SSL member name (VSE) or CMS filename (VM/SP) for ARISYSD8							
80(50)	SSLSAVE - Save area for call to ARISYSD8							
88(58)					RECCNT - GET/PUT count			
96(60)	IOBLKAD - Address of I/O control block (VSE only)				ERRFLD - Error description (VSE only)			
104(68)					RETVAL - ARISYSD5 return code			

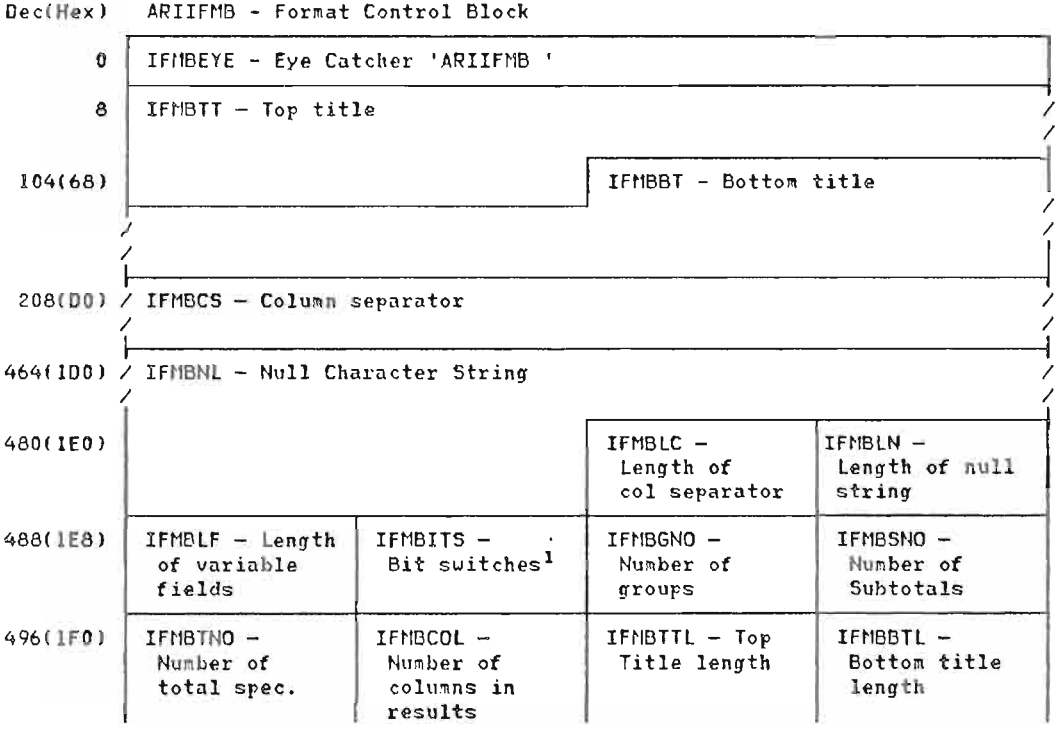
[5] Number of OPEN calls
[6] Number of CLOSE calls
[7] Last request switch:
X'01' = OPENOUT
X'02' = PUT
X'80' = OPENIN
X'C0' = GET
X'FD' = EOF
X'FE' = ERROR
X'FF' = CLOSE
[8] VSE trace I/O switches (see "FDESC Flags" on page 63)
[9] Indicator general I/O module accessed (VSE only). Indicator file is OPEN (VM/SP).
[10] Indicator single I/O module (VSE only)
[11] Flag for READ FILE or READ MEMBER statement

FDESC Flags

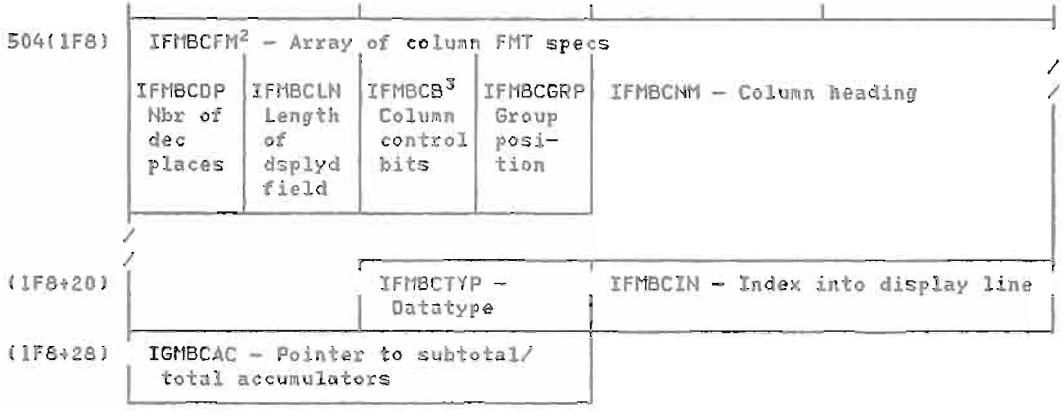
Offset	Field Name	Bits	Meanings
15(0F)	FLAG1		
	ASSIGN	1...	Dynamic tape assign (VSE).
	REWOPEN	.1..	Rewind tape before OPEN (VSE).
	REWCLOS	..1.	Rewind tape after CLOSE (VSE).
	FULLOV	...1	RECFM, LRECL, BLKSIZE to be obtained from FILEDEF or file (VM/SP only).
	BLKFMOV	.... 1...	BLKSIZE and RECFM to be obtained from FILEDEF or file (VM/SP only).
	BLKSZOV	.... .1..	BLKSIZE to be obtained from FILEDEF or file (VM/SP only).
	PRTLOV	.... ..1.	SYSPRINT LRECL value may be obtained from FILEDEF (VM/SP only).
		x	Reserved.
36(24)	FLAG2		
	ASMPREP	1...	PREP ASM write to SYSPCH (ignored in VM).
	COMPREP	.1..	PREP PL/I, COBOL, or FORTRAN to SYSPCH (ignored in VM).
		..xx xxxx	Reserved.
37(25)	TRCSWIT	(VSE only)	
	TRCLVL1	1...	I/O trace level 1 (LST).
	TRCLVL2	.1..	I/O trace level 2 (LST).
	TRCCONS	..1.	Console output device.
	TRCBOTH	...1	Console/SYSLST output.
		 xxxx	Reserved.

The current SQL statement format block (Format Control Block) pointer is field FINSTR (offset X'E4') in the GCB (Global Control Block). FINSTR is a part of the GCB's current SQL statement buffer structure INSTRING (offset X'D8').

The Format Control Block (ARIIFMB) is used to save information about a query, such as top title, bottom title, column headings, grouping, totals, and subtotals. It is based on the pointer FINSTR and is on a doubleword boundary. Storage for the block is gotten (via GETMAIN) in the module ARIIPQY. This is done after a SELECT is entered and only if there is not already a Format Control Block built for the SELECT statement.



[1]See Note 1, page 65.



[2]This array is repeated for every column created.

[3]See Note 2, page 65.

Note 1:

Offset	Field Name	Bits	Meanings
90(5A)	IFMBITS		Bit Switches
	IFMBOL	1...	Outline
	IFMBTB	.1..	Top title on
	IFMBBB	..1.	Bottom title on
	IFMBMS	...1	Master format switch
	IFMBTOT	.... 1...	Put totals out
	IFMBTSP	.... .1..	Put total separator line out
	*	.... ..xx xxxx xxxx	Reserved

Note 2:

506(1FA)	IFMBCB		Column Control Bits
	IFMBCI	1...	Column displayed if on
	IFMBCLZ	.1..	Leading zeros displayed
	IFMBCG	..1.	Column grouped by
	IFMBCST	...1	Column has subtotals
	IFMBCT	.... 1...	Column has totals
	IFMBCCB	.... .1..	Control break this column
	IFMBCSP	.... ..1.	Separator needed
	*	1	Reserved

GCB (ISQL GLOBAL CONTROL BLOCK)

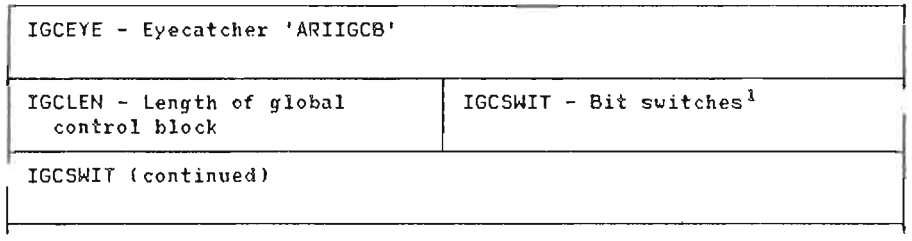
ARIIGCB macro contains the ISQL Global Control Block and other declares that are required by multiple modules. All fields declared in this macro are based on pointers. The actual storage for the fields will be obtained by the other modules. Register 6 points to the Global Control Block in all ISQL modules.

Alphabetic List of Field Names

BEGCOL	560(230)	
BEGTAB	564(234)	
BSIZE	568(238)	
CARDINAL	348(15C)	
CURRENT	584(248)	
DISPPTR	592(250)	
EOFPTR	588(24C)	
EOFREACH	14(E)	(flag bit)
FINSTR	228(E4)	
FIRST	572(23C)	
FMTINFO	560(230)	
FPREV	244(F4)	
IGCABNDQ	16(10)	(flag bit)
IGCABNDT	16(10)	(flag bit)
IGCALARM	13(D)	(flag bit)
IGCANCEL	12(C)	(flag bit)
IGCASQL	160(A0)	
IGCAUTOC	14(E)	(flag bit)
IGCBLNKP	524(20C)	
IGCCASE	417(1A1)	
IGCCDEYE	316(13C)	
IGCCHAR1	2105(839)	
IGCCERR	14(E)	(flag bit)
IGCCICDD	144(90)	
IGCCIC2D	148(94)	
IGCCISQL	15(F)	(flag bit)
IGCCLASS	411(19B)	
IGCCLEAR	16(10)	(flag bit)
IGCCLEN	280(118)	
IGCCMD	280(118)	
IGCCMS	17(11)	(flag bit)
IGCCOLSD	1836(72C)	
IGCCOLSL	344(158)	
IGCCONT	412(19C)	
IGCCOPS	408(198)	
IGCCOST	422(1A6)	
IGCCPARM	404(194)	
IGCCPTR	284(11C)	
IGCCRTN	308(134)	
IGCCSAP	164(A4)	
IGCCVLD	14(E)	(flag bit)
IGCDCPT	413(19D)	
IGCDEL1	414(19E)	
IGCDEL2	415(19F)	
IGCDEST	1832(728)	
IGCDIAG	19(13)	(flag bit)
IGCDLEN	376(178)	
IGCDOTP	332(14C)	
IGCDSPSZ	1828(724)	
IGCDS2DN	14(E)	(flag bit)
IGCDS2RB	15(F)	(flag bit)
IGCDUPCT	421(1A5)	
IGCEIR13	500(1F4)	
IGCERASE	17(11)	(flag bit)
IGCEXIT	14(E)	(flag bit)
IGCEXTRP	428(1AC)	
IGCEXTTR	16(10)	(flag bit)
IGCEYE	0(0)	
IGCFSCRP	488(1E8)	
IGCFWORD	2180(884)	
IGCFWRD	2188(88C)	
IGCGSCRP	496(1F0)	
IGCGSN	200(C8)	
IGCGSP	204(CC)	
IGCGSS	200(C8)	
IGCHELP	15(F)	(flag bit)
IGCHEXOP	520(208)	
IGCHOLD	17(11)	(flag bit)
IGCHTNS	580(244)	
IGCHWRD1	2188(88C)	
IGCHWRD2	2190(88E)	
IGCICANR	12(C)	(flag bit)
IGCICE	19(13)	(flag bit)
IGCICICD	440(1B8)	
IGCIDBS	12(C)	(flag bit)
IGCIDUSW	17(11)	(flag bit)
IGCIECB	100(64)	
IGCIFULD	444(1BC)	
IGCIMMED	14(E)	(flag bit)
IGCINDEX	304(130)	
IGCINEOF	13(D)	(flag bit)
IGCINEX	13(D)	(flag bit)
IGCINL	208(D0)	
IGCINP	212(D4)	
IGCINPUT	13(D)	(flag bit)
IGCINS	208(D0)	
IGCINTL	400(190)	
IGCIOREQ	15(F)	(flag bit)
IGCIPATD	448(1C0)	
IGCIPFKD	456(1C8)	
IGCIRETD	460(1CC)	
IGCISCMD	464(1D0)	
IGCITRM	12(C)	(flag bit)
IGCITRMD	468(1D4)	
IGCIVFYD	472(1D8)	
IGCIWATD	476(1DC)	
IGCIWRTD	480(1E0)	
IGCLBNO	380(17C)	
IGCLBPTR	388(184)	
IGCLBSZ	384(180)	
IGCLBUF	380(17C)	
IGCLEN	8(8)	
IGCLINEP	504(1F8)	
IGCLINSZ	558(22E)	
IGCLNUM	392(188)	
IGCLPTR	1624(658)	
IGCMAPP	436(1B4)	
IGCMSG	18(12)	(flag bit)
IGCMSGI	18(12)	(flag bit)
IGCNEND	328(148)	
IGCNFR	15(F)	(flag bit)
IGCNLD	356(164)	
IGCNLL	352(160)	
IGCNODSP	16(10)	(flag bit)
IGCNOTD	18(12)	(flag bit)
IGCNOTR	18(12)	(flag bit)
IGCNSCRL	556(22C)	
IGCNSCRP	300(12C)	
IGCNTR	534(216)	
IGCNULLM	19(13)	(flag bit)
IGCNUMR	2106(83A)	
IGCOCE	2100(834)	
IGCOLEN	288(120)	
IGCONTUE	18(12)	(flag bit)
IGCOPCNT	312(138)	
IGCOPR	288(120)	
IGCOPTR	292(124)	
IGCORTN	324(144)	
IGCOUTLN	16(10)	(flag bit)
IGCOUTP	512(200)	
IGCPATCH	452(1C4)	
IGCPFK	420(1A4)	

```
IGCPF9      15(F)   (flag bit)
IGCPGSZ     396(18C)
IGCPINIT    13(D)   (flag bit)
IGCPLEN     396(18C)
IGCPRCPR    13(D)   (flag bit)
IGCPREOF    13(D)   (flag bit)
IGCPROC     12(C)   (flag bit)
IGCPROCQ    336(150)
IGCPRTR     2091(82B)
IGCPRTR1    2091(82B)
IGCPRTR2    2091(82B)
IGCPTR      2184(888)
IGCPWID     398(18E)
IGCRBDON    13(D)   (flag bit)
IGCRC       538(21A)
IGCREAD     12(C)   (flag bit)
IGCREADL    176(B0)
IGCREADP    180(B4)
IGCREADS    176(B0)
IGCREEND    19(13)  (flag bit)
IGCRETRN    15(F)   (flag bit)
IGCRFWRD    2104(838)
IGCRMARP    432(1B0)
IGCRMXP     2096(830)
IGCRPLYP    516(204)
IGCRPTAP    1824(720)
IGCRSAVE    32(20)
IGCRSTRT    18(12)  (flag bit)
IGCRSWA1    21(15)  (flag bit)
IGCRSWA2    21(15)  (flag bit)
IGCRSWA3    21(15)  (flag bit)
IGCRSWA4    21(15)  (flag bit)
IGCRSWA5    21(15)  (flag bit)
IGCRSWA6    21(15)  (flag bit)
IGCRSWA7    21(15)  (flag bit)
IGCRSWA8    21(15)  (flag bit)
IGCRSWB1    22(16)  (flag bit)
IGCRSWB2    22(16)  (flag bit)
IGCRSWB3    22(16)  (flag bit)
IGCRSWB4    22(16)  (flag bit)
IGCRSWB5    22(16)  (flag bit)
IGCRSWB6    22(16)  (flag bit)
IGCRSWB7    22(16)  (flag bit)
IGCRSWB8    22(16)  (flag bit)
IGCRSWC1    23(17)  (flag bit)
IGCRSWC2    23(17)  (flag bit)
IGCRSWC3    23(17)  (flag bit)
IGCRSWC4    23(17)  (flag bit)
IGCRSWC5    23(17)  (flag bit)
IGCRSWC6    23(17)  (flag bit)
IGCRSWC7    23(17)  (flag bit)
IGCRSWC8    23(17)  (flag bit)
IGCRSW91    20(14)  (flag bit)
IGCRSW92    20(14)  (flag bit)
IGCRSW93    20(14)  (flag bit)
IGCRSW94    20(14)  (flag bit)
IGCRSW95    20(14)  (flag bit)
IGCRSW96    20(14)  (flag bit)
IGCRSW97    20(14)  (flag bit)
IGCRSW98    20(14)  (flag bit)
IGCRTRP     424(1A8)
IGCRUNM     2091(82B)
IGCSAVED    19(13)  (flag bit)
IGCSCRHT    112(70)
IGCSCRWD    114(72)
IGCSQDSL    552(228)
IGCSQDSP    548(224)
IGCSQL      12(C)   (flag bit)
IGCSQLCA    152(98)
IGCSQLDA    156(9C)
IGCSQLRC    402(192)
IGCSSCRP    492(1EC)
IGCSTACK    116(74)
IGCSTATL    184(B8)
IGCSTATP    188(BC)
IGCSTATS    184(B8)
IGCSTATX    484(1E4)
IGCSTEND    19(13)  (flag bit)
IGCSTKID    116(74)
IGCSTKL     124(7C)
IGCSTKNO    540(21C)
IGCSTKP     128(80)
IGCSTKSV    2092(82C)
IGCSTRCD    418(1A2)
IGCSWIT     12(C)   (flag bit)
IGCSYSCL    544(220)
IGCTASKN    136(88)
IGCTBL      320(140)
IGCTECB     96(60)
IGCTERM     12(C)   (flag bit)
IGCTINDX    532(214)
IGCTKNO     17(11)  (flag bit)
IGCTRCUR    2172(87C)
IGCTREND    2176(880)
IGCTRL      536(218)
IGCTRMID    104(68)
IGCTRNID    140(8C)
IGCTRON     18(12)  (flag bit)
IGCTRPTR    528(210)
IGCTRWP     18(12)  (flag bit)
IGCTRWRT    19(13)  (flag bit)
IGCTSD      16(10)  (flag bit)
IGCTSK      132(84)
IGCTSKID    132(84)
IGCTSL      168(A8)
IGCTSP      172(AC)
IGCTSR      16(10)  (flag bit)
IGCTSS      168(A8)
IGCTSSAV    2108(83C)
IGCTSW      15(F)   (flag bit)
IGCUID      24(18)
IGCVFYC     19(13)  (flag bit)
IGCWORKP    508(1FC)
IGCWRIT     416(1A0)
IGCWRITL    192(C0)
IGCWRITP    196(C4)
IGCWRITS    192(C0)
IGCXTRC1    17(11)  (flag bit)
IGCXTRC2    17(11)  (flag bit)
IGCXTRC3    17(11)  (flag bit)
INSTRING    216(D8)
IPOS        296(128)
LAST        576(240)
LINSTR      216(D8)
LPREV       232(E8)
LTOKEN      248(F8)
LTOKEN2     256(100)
LTOKEN3     264(108)
LTOKEN4     272(110)
NLBLANKS    1620(654)
OFFSETS     596(254)
PINSTR      220(DC)
PPREV       236(EC)
PREVIOUS    232(E8)
PTOKEN      252(FC)
PTOKEN2     260(104)
PTOKEN3     268(10C)
PTOKEN4     276(114)
QINPROG     14(E)   (flag bit)
TINSTR      224(E0)
TOKEN       248(F8)
TOKEN2      256(100)
TOKEN3      264(108)
TOKEN4      272(110)
TPREV       240(F0)
```

Dec(Hex) IGCB - ISQL Global Control Block



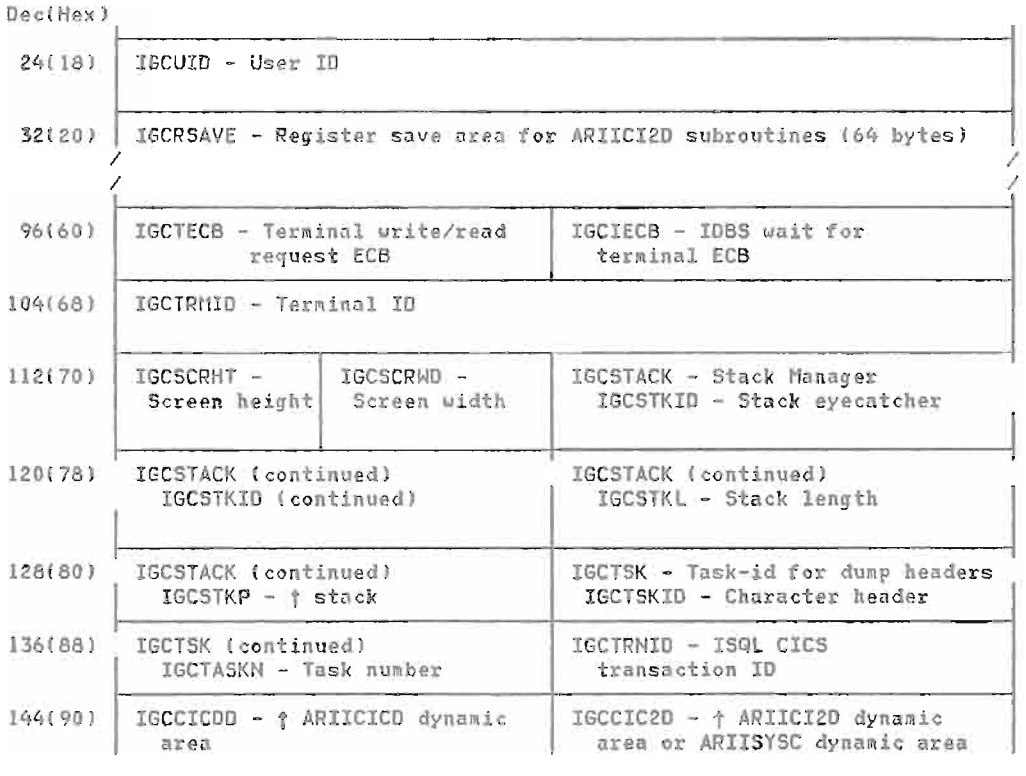
Note 1:

Offset	Field Name	Bits	Meaning
12(C)	IGCSWIT		
	Byte 0		
12(C)	IGCITRM	1...	ITRM transaction is up
	IGCPROC	.1..	Read from procedure
	IGCTERM	..1.	Forced terminal I/O
	IGCICANR	...1	ARIICAN was called
	IGCSQL	.... 1...	SQL statement input
	IGCANCEL	.... .1..	CANCEL entered
	IGCIDBS	.... ..1.	ARIIRWI called by ARIIDBS
	IGCREAD	1	READ after WRITE
	Byte 2		
14(E)	IGCAUTOC	1...	Autocommit On mode
	IGCIMMED	.1..	Immediate On mode
	IGCCVLD	..1.	Valid command
	IGCERR	...1	CICS error detected
	QINPROG	.... 1...	Query in progress
	IGCEXIT	.... .1..	Exit to CICS
	EOFREACH	.... ..1.	End of query rows
	IGCDS2DN	1	SQL/DS down
	Byte 4		
16(10)	IGCTSR	1...	Read from temporary storage
	IGCTSD	.1..	Delete temporary storage
	IGCEXTTR	..1.	Extract trace on
	IGCOUTLN	...1	Display all fields
	IGCCLEAR	.... 1...	Clear before continuing
	IGCABNDT	.... .1..	ARIT abended
	IGCABNDQ	.... ..1.	ARIQ abended
	IGCNODSP	1	Input not displayed
	Byte 6		
18(12)	IGCONTUE	1...	Continuation
	IGCTRWP	.1..	Wrap trace table
	IGCTRON	..1.	Display trace
	IGCRSTRT	...1	Cancel verify restart
	IGCNOTR	.... 1...	No trace entry
	IGCNOTD	.... .1..	No trace display
	IGCMSGI	.... ..1.	Special message reply
	IGCMSG	1	Normal message reply

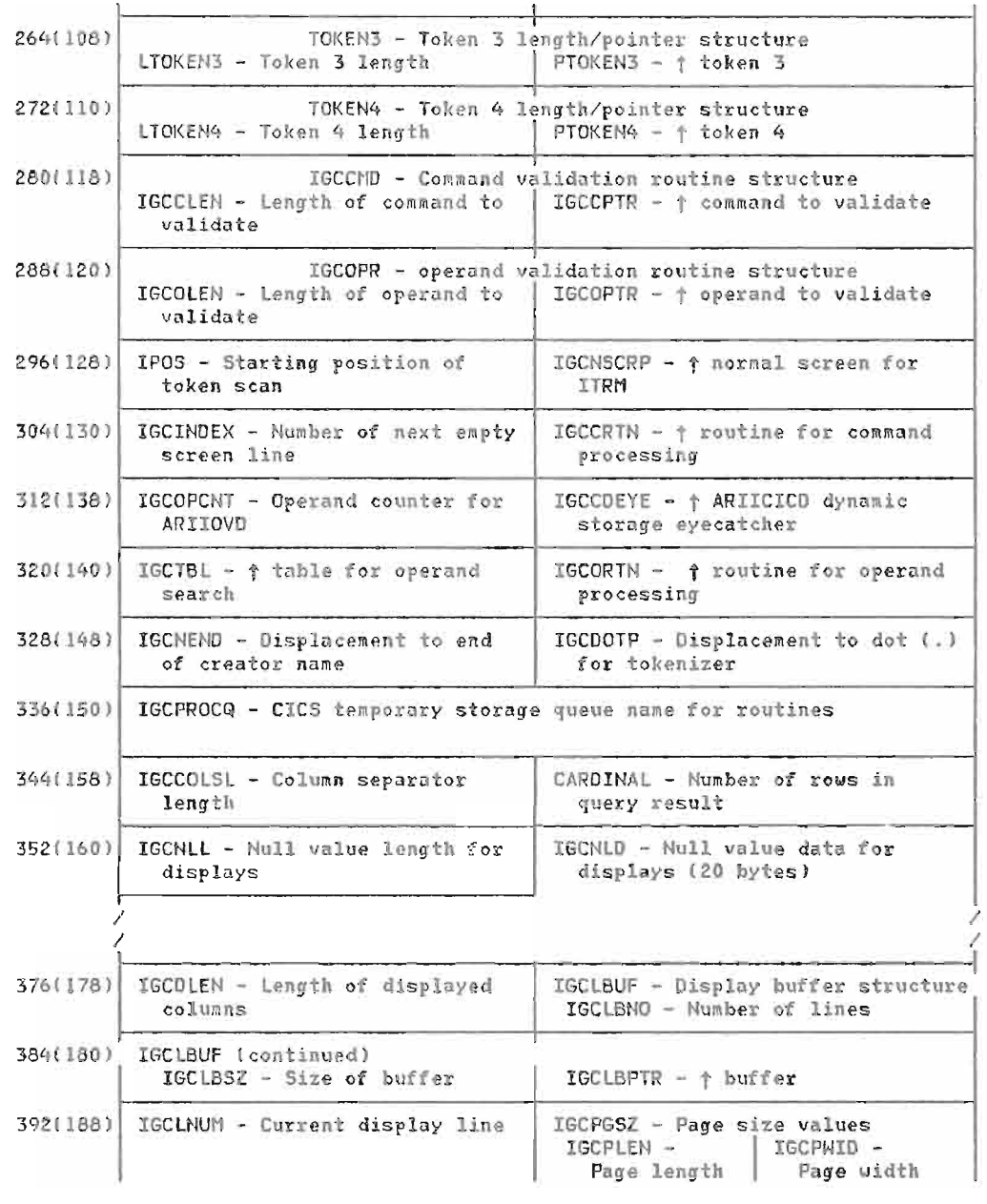
Offset	Field Name	Bits	Meaning
	Byte 1		
13(D)	IGCINPUT	1...	In bulk input data mode
	IGCINEX	.1..	Exit from build input data mode
	IGCINEOF	..1.	End-of-file in a routine for INPUT
	IGCALARM	...1	Sound terminal alarm
	IGCRBDON	.... 1...	Rollback done, cancel
	IGCPINIT	.... .1..	PROC initialization processing
	IGCPRCPR	.... ..1.	PROC line printed
	IGCPREOF	1	End-of-file on a procedure
	Byte 3		
15(F)	IGCDS2RB	1...	Indicates SQL/DS caused a ROLLBACK
	IGCPF9	.1..	PF Key 9 was pressed
	IGCCISQL	..1.	ARIIVFYC called from ISQL
	IGCHELP	...1	HELP command is being processed
	IGCNFR	.... 1...	No first record in buffer
	IGCRETRN	.... .1..	ITRM return to CICS
	IGCIOREQ	.... ..1.	ITRM I/O request
	IGCTSW	1	Write to temporary storage
	Byte 5		
17(11)	IGCXTRC1	1...	Extract trace on
	IGCXTRC2	.1..	Extract JCL trace on
	IGCXTRC3	..1.	Extract message trace on
	IGCIDUSW	...1	Do a COMMIT
	IGCTKNO	.... 1...	Use zeros for token delimiters
	IGCCMS	.... .1..	VM/CMS
	IGCERASE	.... ..1.	Erase screen
	IGCHOLD	1	HOLD (PF 9)
	Byte 7		
19(13)	IGCICE	1...	Improved command entry
	IGCTRWRT	.1..	Write trace entry
	IGCREEND	..1.	Routine error simulates end for input and query display
	IGCSTEND	...1	Set error
	IGCVFYC	.... 1...	ARIIREAC called by ARIIVFYC
	IGCDIAG	.... .1..	Diagnostic write done
	IGCSAVED	.... ..1.	Duplicate count saved
	IGCNULLM	1	Null message line

Offset	Field Name	Bits	Meaning
	Byte 8		
20(14)	IGCRSW91	1...	Switch 91 (Reserved)
	IGCRSW92	.1..	Switch 92 (Reserved)
	IGCRSW93	..1.	Switch 93 (Reserved)
	IGCRSW94	...1	Switch 94 (Reserved)
	IGCRSW95	.... 1...	Switch 95 (Reserved)
	IGCRSW96	.... .1..	Switch 96 (Reserved)
	IGCRSW97	.... ..1.	Switch 97 (Reserved)
	IGCRSW98	1	Switch 98 (Reserved)
	Byte 10		
22(16)	IGCRSWB1	1...	Switch B1 (Reserved)
	IGCRSWB2	.1..	Switch B2 (Reserved)
	IGCRSWB3	..1.	Switch B3 (Reserved)
	IGCRSWB4	...1	Switch B4 (Reserved)
	IGCRSWB5	.... 1...	Switch B5 (Reserved)
	IGCRSWB6	.... .1..	Switch B6 (Reserved)
	IGCRSWB7	.... ..1.	Switch B7 (Reserved)
	IGCRSWB8	1	Switch B8 (Reserved)

Offset	Field Name	Bits	Meaning
	Byte 9		
21(15)	IGCRSWA1	1...	Switch A1 (Reserved)
	IGCRSWA2	.1..	Switch A2 (Reserved)
	IGCRSWA3	..1.	Switch A3 (Reserved)
	IGCRSWA4	...1	Switch A4 (Reserved)
	IGCRSWA5	.... 1...	Switch A5 (Reserved)
	IGCRSWA6	.... .1..	Switch A6 (Reserved)
	IGCRSWA7	.... ..1.	Switch A7 (Reserved)
	IGCRSWA8	1	Switch A8 (Reserved)
	Byte 11		
23(17)	IGCRSWC1	1...	Switch C1 (Reserved)
	IGCRSWC2	.1..	Switch C2 (Reserved)
	IGCRSWC3	..1.	Switch C3 (Reserved)
	IGCRSWC4	...1	Switch C4 (Reserved)
	IGCRSWC5	.... 1...	Switch C5 (Reserved)
	IGCRSWC6	.... .1..	Switch C6 (Reserved)
	IGCRSWC7	.... ..1.	Switch C7 (Reserved)
	IGCRSWC8	1	Switch C8 (Reserved)

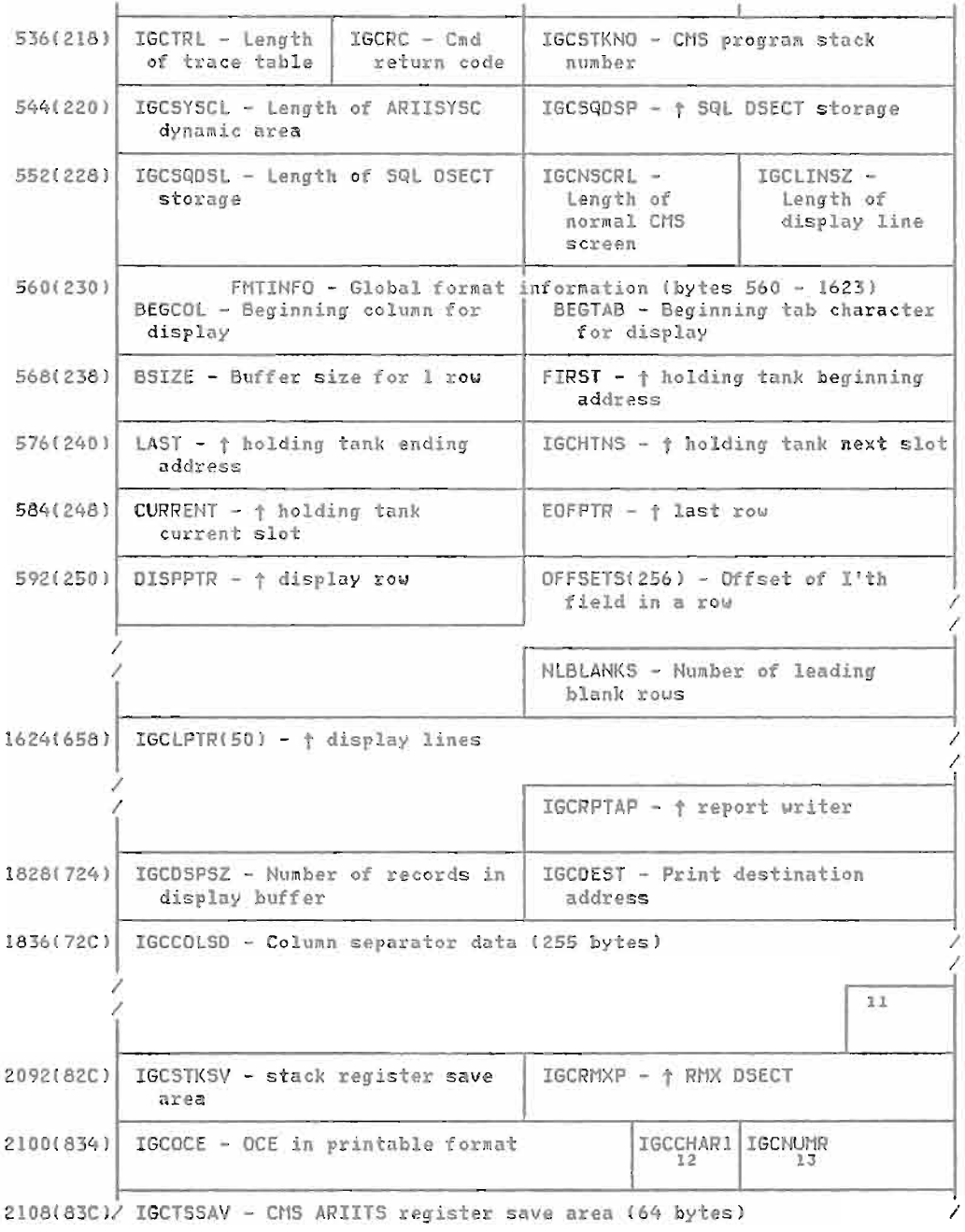


152(98)	IGCSQLCA - ↑ SQLCA	IGCSQLDA - ↑ SQLDA
160(A0)	IGCASQL - ↑ active SQL statement	IGCCSAP - ↑ CICS CSA
168(A8)	IGCTSS - Temporary storage write structure	
	IGCTSL - Temporary storage write length	IGCTSP - Temporary storage write pointer
176(B0)	IGCREADS - Input length/pointer structure	
	IGCREADL - Length of input area used	IGCREADP - ↑ 2K input area
184(B8)	IGCSTATS - Status area length/pointer structure	
	IGCSTATL - Length of status area used	IGCSTATP - ↑ status area
192(C0)	IGCWRITS - Output area length/pointer structure	
	IGCWRITL - Length of output area used	IGCWRITP - ↑ output area
200(C8)	IGCGSS - General screen length/pointer structure	
	IGCGSN - Number of general screen lines	IGCGSP - ↑ general screen output area
208(D0)	IGCINS - Terminal/procedure input area structure	
	IGCINL - Length of terminal/procedure IPT area used	IGCINP - ↑ terminal/procedure input area
216(D8)	INSTRING - Current SQL statement buffer structure	
	LINSTR - Current SQL statement actual length	PINSTR - ↑ current SQL statement
	INSTRING (continued)	
224(E0)	TINSTR - Current SQL statement total used length	FINSTR - ↑ current SQL statement format block
232(E8)	PREVIOUS - Previous SQL statement buffer structure	
	LPREV - Previous SQL statement actual length	PPREV - ↑ previous SQL statement
	PREVIOUS (continued)	
240(F0)	TPREV - Previous SQL statement total used length	FPREV - ↑ previous SQL statement format block
248(F8)	TOKEN - Token length/pointer structure	
	LTOKEN - Token length	PTOKEN - ↑ token
256(100)	TOKEN2 - Token 2 length/pointer structure	
	LTOKEN2 - Token 2 length	PTOKEN2 - ↑ token 2

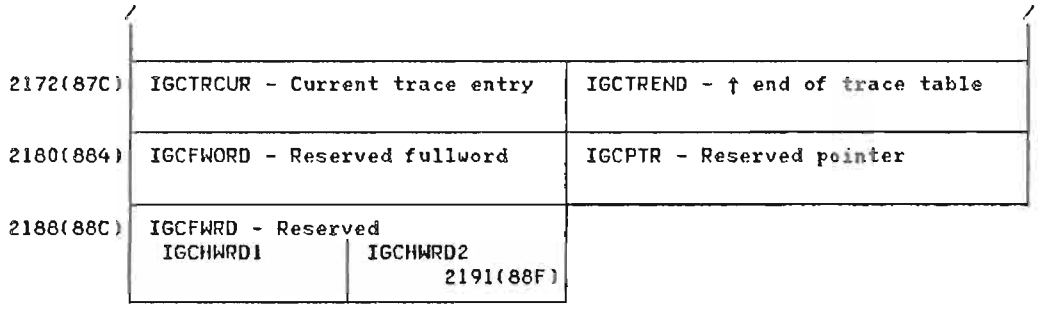

264(108)
TOKEN3 - Token 3 length/pointer structure
LTOKEN3 - Token 3 length
PTOKEN3 - ↑ token 3
272(110)
TOKEN4 - Token 4 length/pointer structure
LTOKEN4 - Token 4 length
PTOKEN4 - ↑ token 4
280(118)
IGCCMD - Command validation routine structure
IGCCLEN - Length of command to validate
IGCCPTR - ↑ command to validate
288(120)
IGCOPR - operand validation routine structure
IGCOLEN - Length of operand to validate
IGCOPTR - ↑ operand to validate
296(128)
IPOS - Starting position of token scan
IGCNSCRP - ↑ normal screen for ITRM
304(130)
IGCINDEX - Number of next empty screen line
IGCCRTN - ↑ routine for command processing
312(138)
IGCOPCNT - Operand counter for ARIIOVD
IGCCDEYE - ↑ ARIICICD dynamic storage eyecatcher
320(140)
IGCTBL - ↑ table for operand search
IGCORTN - ↑ routine for operand processing
328(148)
IGCNEND - Displacement to end of creator name
IGCDOTP - Displacement to dot (.) for tokenizer
336(150)
IGCPROCQ - CICS temporary storage queue name for routines
344(158)
IGCCOLSL - Column separator length
CARDINAL - Number of rows in query result
352(160)
IGCNLL - Null value length for displays
IGCNLD - Null value data for displays (20 bytes)
376(178)
IGCDLEN - Length of displayed columns
IGCLBUF - Display buffer structure
IGCLBNO - Number of lines
384(180)
IGCLBUF (continued)
IGCLBSZ - Size of buffer
IGCLBPTR - ↑ buffer
392(188)
IGCLNUM - Current display line
IGCPGSZ - Page size values
IGCPLEN - Page length
IGCPWID - Page width

400(190)	IGCINTL - Total input area length available		IGCSQLRC - SQL return code	IGCCPARM - ↑ CICS parameter list			
408(198)	IGCCOPS - Number of copies		IGCCLASS [2]	IGCCONT [3]	IGCDCPT [4]	IGCDEL1 [5]	IGCDEL2 [6]
416(1A0)	IGCWRIT [7]	IGCCASE [8]	IGCSTRCD - ARIICICD CICS start code	IGCPFK [9]	IGCDUPCT [10]	IGCCOST - ISQL optimizer cost estimate	
424(1A8)	IGCRTRP - ↑ retrieve buffer			IGCEXTRP - ↑ extract work area			
432(1B0)	IGCRMARP - ↑ RMAR control block			IGCMAPP - ↑ map system-dependent table			
440(1B8)	IGCICICD - ↑ ARIICICD for map			IGCIFULD - ↑ ARIIFULD for map			
448(1C0)	IGCIPATD - ↑ ARIIPATD for map			IGCPATCH - ↑ ARIIPAT for map			
456(1C8)	IGCIPFKD - ↑ ARIIPFKD for map			IGCIRETD - ↑ ARIIRETD for map			
464(1D0)	IGCISCMD - ↑ ARIISCMD for map			IGCITRMD - ↑ ARIITRM for map			
472(1D8)	IGCIVFYD - ↑ ARIIVFYD for map			IGCIWATD - ↑ ARIIWATD for map			
480(1E0)	IGCIWRTD - ↑ ARIIWRTD for map			IGCSTATX - ↑ status messages			
488(1E8)	IGCFSCRP - ↑ full screen buffer			IGCSSCRP - ↑ 3270 status area			
496(1F0)	IGCGSCRP - ↑ 3270 general screen work area			IGCEIR13 - ↑ CICS EIB for ARIICICD			
504(1F8)	IGCLINEP - ↑ 3270 buffer address matrix			IGCWORKP - ↑ 3270 total area			
512(200)	IGCOUTP - ↑ 3270 output message			IGCRPLYP - ↑ 3270 reply area			
520(208)	IGCHEXOP - ↑ 3270 hex 0 area			IGCBLNKP - ↑ 3270 blank area			
528(210)	IGCTRPTR - ↑ trace table			IGCTINDX - Index into trace table		IGCNTR - Number of trace entries	

[2] Class value
[3] Continuation character
[4] DECOPT value
[5] Tokenizer first delimiter
[6] Tokenizer second delimiter
[7] Read/Write flag
    L - Single line write
    G - General screen write
    S - Status area write
[8] Set case operand (upper/lower)
[9] PF key number
[10] Retrieve duplicate count



11 IGCRUNM - Set run-mode codes
11.. .... '00'B - CONTINUE
'01'B - STOP
'10'B - CANCEL
IGCPRTR - Print route code
IGCPRTR1 - ARIIPRTD uses to save/restore
IGCPRTR2 - ARIIPRTD uses to save/restore
..11 .... '00'B - SYSTEM
'01'B - CICS/VS
'10'B - POWER/VS
.... xxxx Reserved

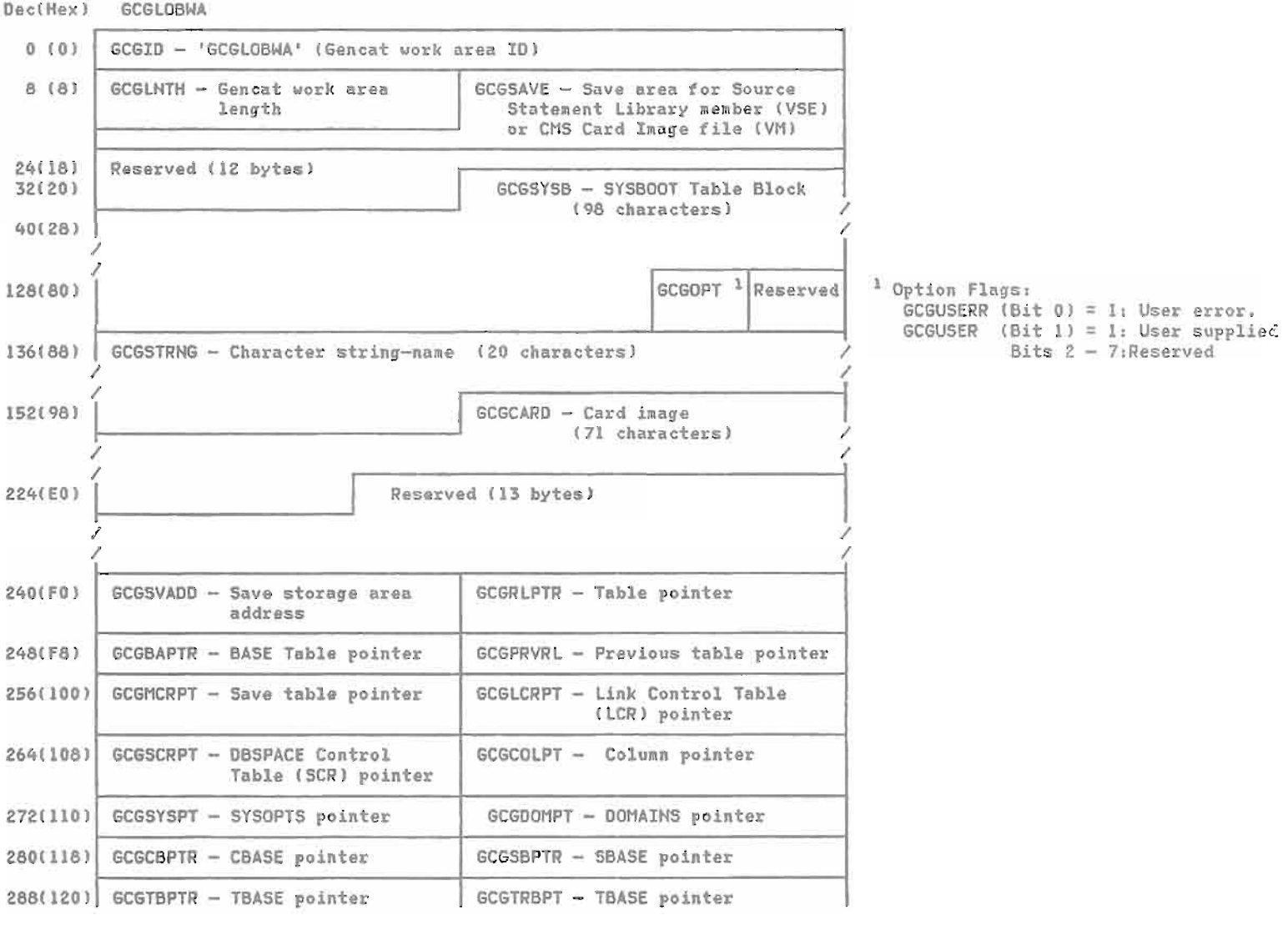


[12] Reserved character

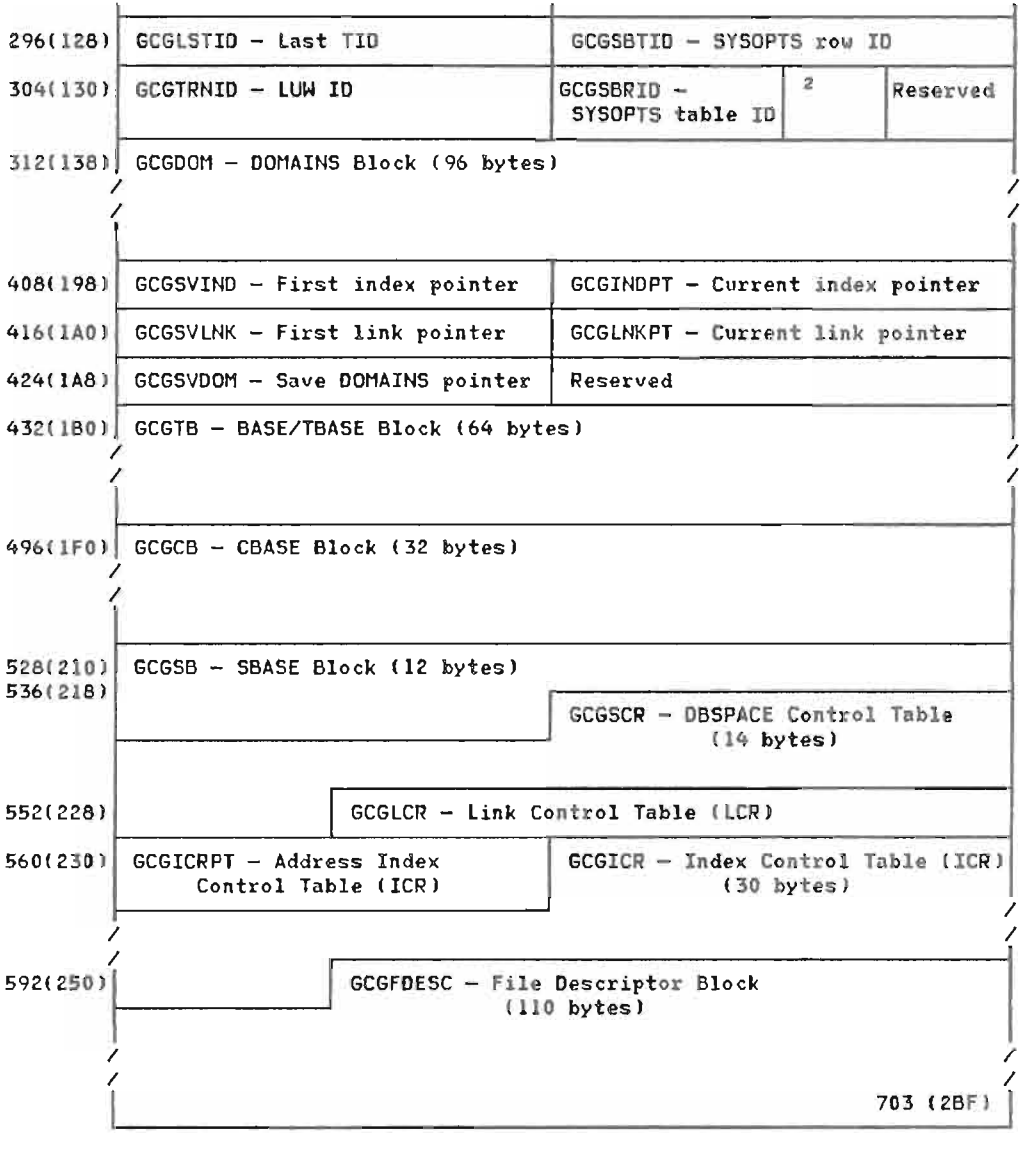
[13] Number of records in program stack

GCGLOBWA (CATALOG GENERATION GLOBAL WORK AREA)

The GCGLOBWA contains base structure blocks, a file descriptor block, a record area for the source statement library member (ARISCAT) reads, pointers, and other control information necessary for catalog generation. ARISMAI gets the storage for the work area and sets up the addressability. Register 11 points to the work area in all catalog generation modules. The GCGLOBWA is mapped by ARIBGCG.



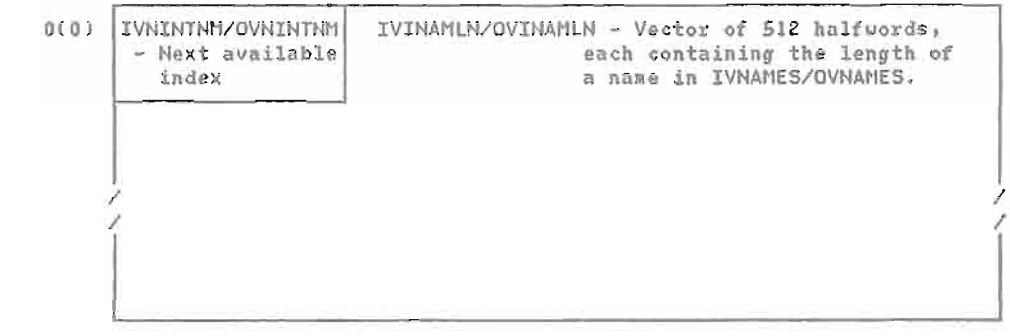
Dec(Hex) GCGLOBWA (continued)



2 GCGOPRT (Bit 0) = 1: Open print data set
GCGLSTCL (Bit 1) = 1: Last column in DEFINE table
GCGEXDFM (Bit 2) = 1: Exit loop in DEFINE index
GCGENDC (Bit 3) = 1: Exit loop in read card
Bits 4 - 7: Reserved

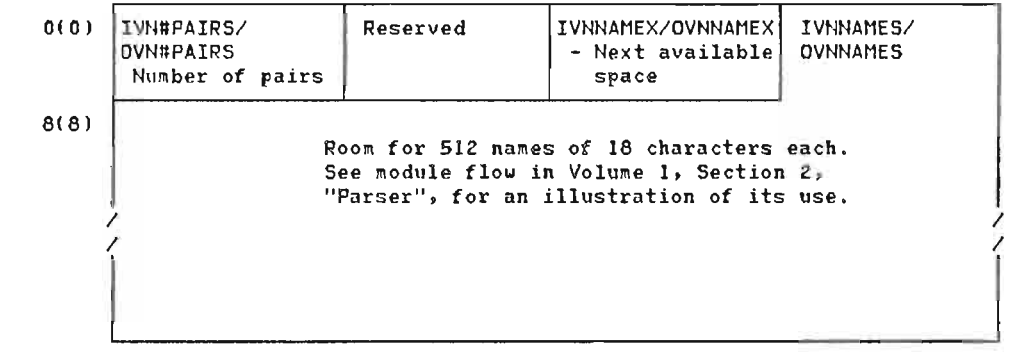
IVIND/OVIND (INPUT OR OUTPUT VARIABLE INDEX)

ARIBIVI and ARIBOVI map the host variable index list. This list contains the length of each name in IVNAMES and OVNAMES.



IVNAMES/OVNAMES (INPUT OR OUTPUT VARIABLE NAMES)

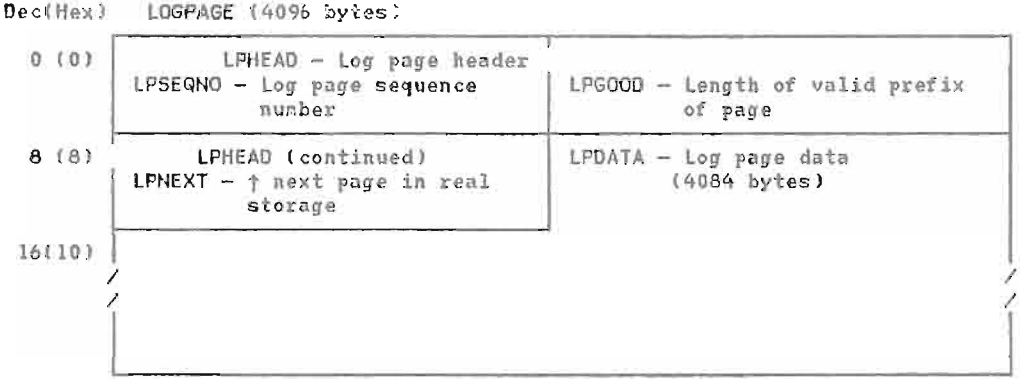
ARIBIVN and ARIBOVN map the input and output variable name list, which contain input or output host variables found in a SQL statement by the Parser.



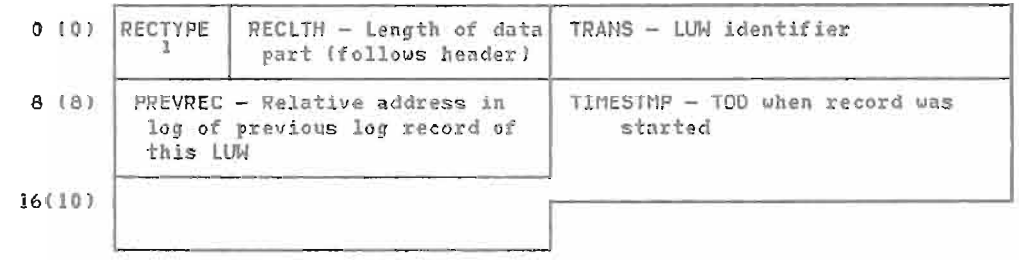
LOG PAGES AND RECORDS

This subsection consists of templates for log pages and for those log records that are used by the Log subsystem itself. These areas are mapped by YLOGTMPL.

LOG SEGMENT PAGE TEMPLATE

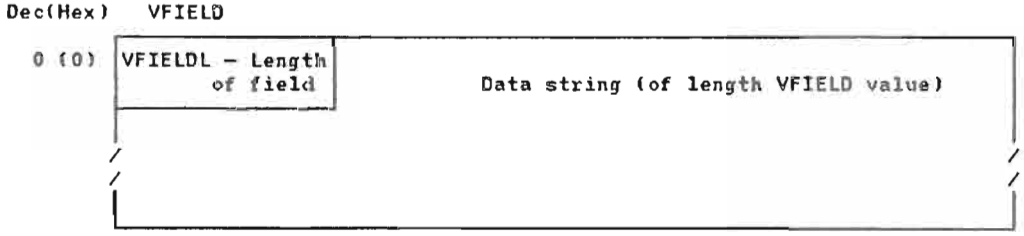


LOG RECORD HEADER

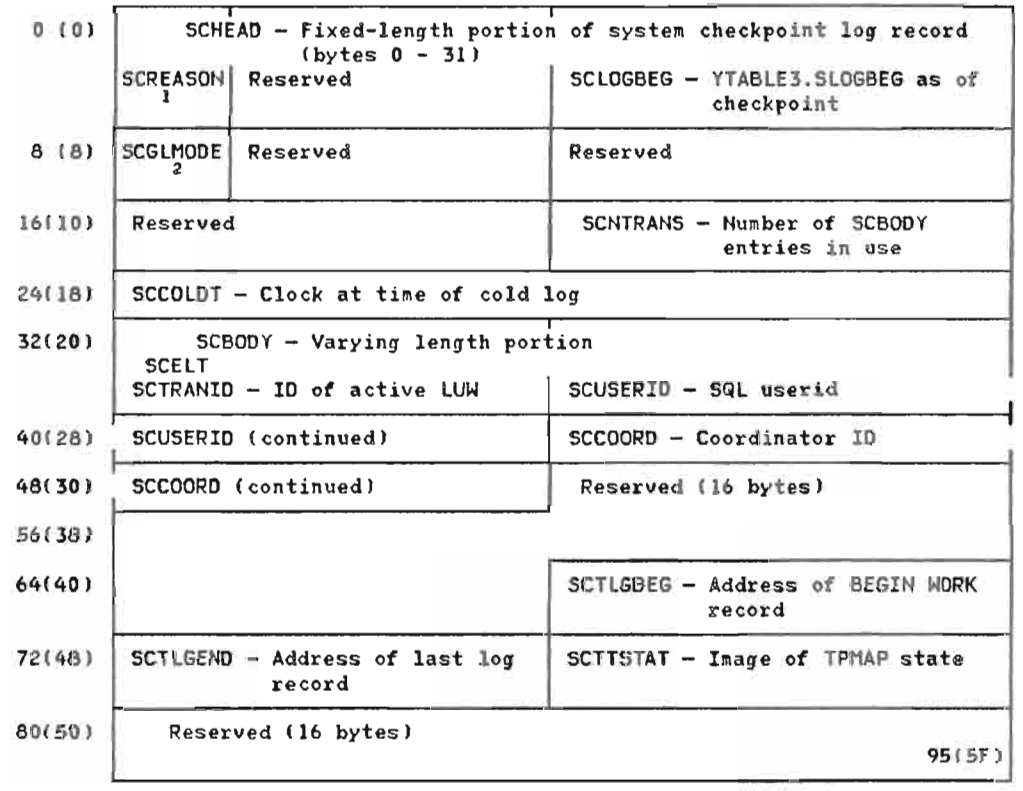


[1] Record type:
- 0 - Not-yet-written log record
- 1 - Smallest code of UPDATE-type record
- 16 - Largest code of UPDATE-type record
- 20 - Begin LUW
- 21 - LUW savepoint
- 22 - End LUW
- 23 - Abort LUW
- 24 - System checkpoint
- 25 - Prepared-to-Commit

VARIABLE-LENGTH FIELD TEMPLATE



SYSTEM CHECKPOINT RECORD



[1] Reason for checkpoint:
- 'I' - Log just initialized (formatted (cold start)
- 'R' - System just recovered (warm start)
- 'N' - No logging this session
- 'C' - Periodic system checkpoint
- 'S' - System shutdown
- 'B' - Archive checkpoint (just before archive begins)
- 'A' - Archive checkpoint (after archive complete)

[2] Log mode as of checkpoint

EXTENSION OF CHECKPOINT RECORD

Dec(Hex) SCHKDAT1

0 (0)	SCARCHIV – Address of most recent archive checkpoint (or 0 if no archive)	SMAXTRAN – Largest LUW sequence number used to date

BEGIN WORK / SAVE WORK RECORD DATA PART

Dec(Hex) TSAVDATA

0 (0)	NEXTREC – † next record. Set during recovery to bypass invalid records	USERID – Identifier of user (8 bytes)	
8 (8)		TSCOORID – Coordinator ID (8 bytes)	
16(10)		TSRCVTOK – Recovery token (8 bytes)	
24(18)		Reserved (16 bytes)	
32(20)			
40(28)		SAVEDNO – Number of savepoint	
48(30)	LASTLOCK – † last lock held at savepoint	SCANS SCANLTH – Length of SCANDATA	SCANDATA – Scan data (length in field SCANLTH)
56(38)			
/	/	/	/
/	/	/	/

End SCANDATA + 0 (0)	USERDATA UDATALTH – UDATA length	UDATA – User data (length = value in UDATALTH)
8 (8)	/ /	/ /

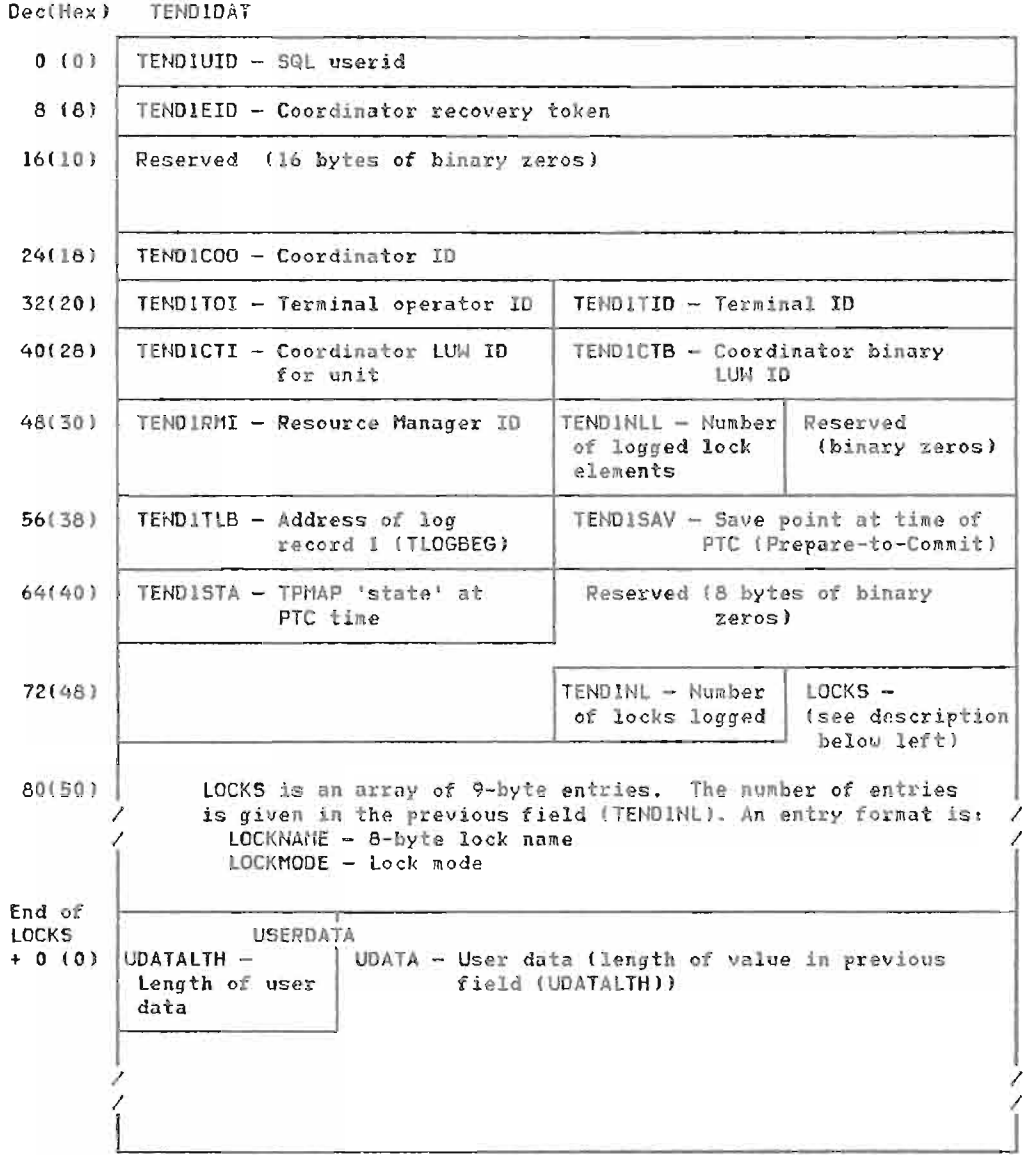
ABORT LUW RECORD

Dec(Hex)	TABTDATA
0 (0)	TAFUSER - User ID
8 (8)	TAFCOORD - Coordinator ID
16(10)	TAFRCVTK - Recovery token
24(18)	Reserved (16 bytes)
32(20)	

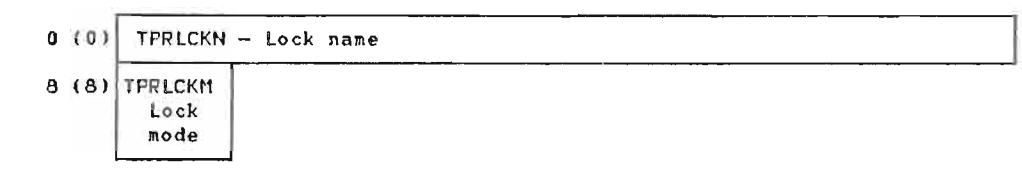
END LUW LOG RECORD

Dec(Hex)	TENDDATA
	This is an array of 1-byte fields containing the length and value of user data.

PREPARED-TO-COMMIT LOG RECORD



LOCK ELEMENT (FROM PTC RECORD)



LOG DATA FOR A DBSI (ARIYM00) CALL AS PASSED TO/FROM THE LOG LINKAGE

The following structures are mapped by the YLOG macro.

OVERLAY FOR FIRST BYTES OF ALL LOG RECORDS

LCOMHEAD

Dec (Hex)				
0 (0)	LCOMFLG[1] Log Record Flags	LCOMSEG[2]	LCOMTID[2]	LCOMRID RID for calls: INSERT, DELETE, UPDATE
8 (8)	LCOMRID cont'd			

[1] Log record flags:
```
1... .... If child entity, page lock
          in backup
.1.. .... If parent entity, page lock
          in backup
..1. .... If index, page lock in
          backup
...1 .... If any index has to be updated
          the meaning of the other bit
          depends on the type of log
          record
.... xxxx Reserved
```

[2] DBSPACE and TID of base row

LINSERT - Log Data for INSERT

Dec (Hex)							
0 (0)	LINSHEAD - Header of logged data						
	LINSFLG[1] Log flags	LINSSEG[2]	LINSTID[2]				LINSRID[2]
8 (8)	LINSHEAD (continued)				LINSPREF[3] - Link prefix information		
	LINSRID (cont'd)	LINSHD4 - Bytes 3 - 6 of prefix of inserted row			LINSCP Copy of YLCTIDP for LID	LINSPP Copy of YLPTIDP, PSEGMENT	LINSPSEG Parent DBSPACE number (Don't care if parent = 0)
		LINSLTH Total length of row	LINSRCD Relation code	LINSPRL Length of link prefix			
16 (10)	LINSPREF (continued)						
	LINSPSEG (cont'd)	LINSTIDS - TIDs of R-twin, L-twin, parent of row 1					
		LINSRT			LINSLT		
24 (18)	LINSPREF (continued)						
	LINSTIDS (continued)				LINST2 [4]		
	LINSLT (cont'd)	LINSPA			LINST201 - TID of old overflow row (0 if none)		
32 (20)	LINSPREF (continued)						
	LINST2 (continued)						LINSTOD
	LINST201 (cont'd)	LINST202 - TID of new overflow row (0 if none)			LINST2L1 Length [5]	LINST2L2 Length[6]	TID of data row of row 0
40 (28)	LINSPREF (continued)						
	LINSTOD (continued)	LINSVAL - Field values of inserted row					
		/ /					

[1] Log Flags for row:
1... .... If child entity, page lock in backup
.1.. .... If parent entity, page lock in backup
..1. .... If index, page lock in backup
...1 .... If any index exists
.... 1... If row also being connected
.... .1.. If row 2 is a parent
.... ..1. If row 2 is a left twin
....x Reserved

[2] ID (DBSPACE, TID, RID) of inserted row

[3] Present if fifth bit of LINSFLG = 1.

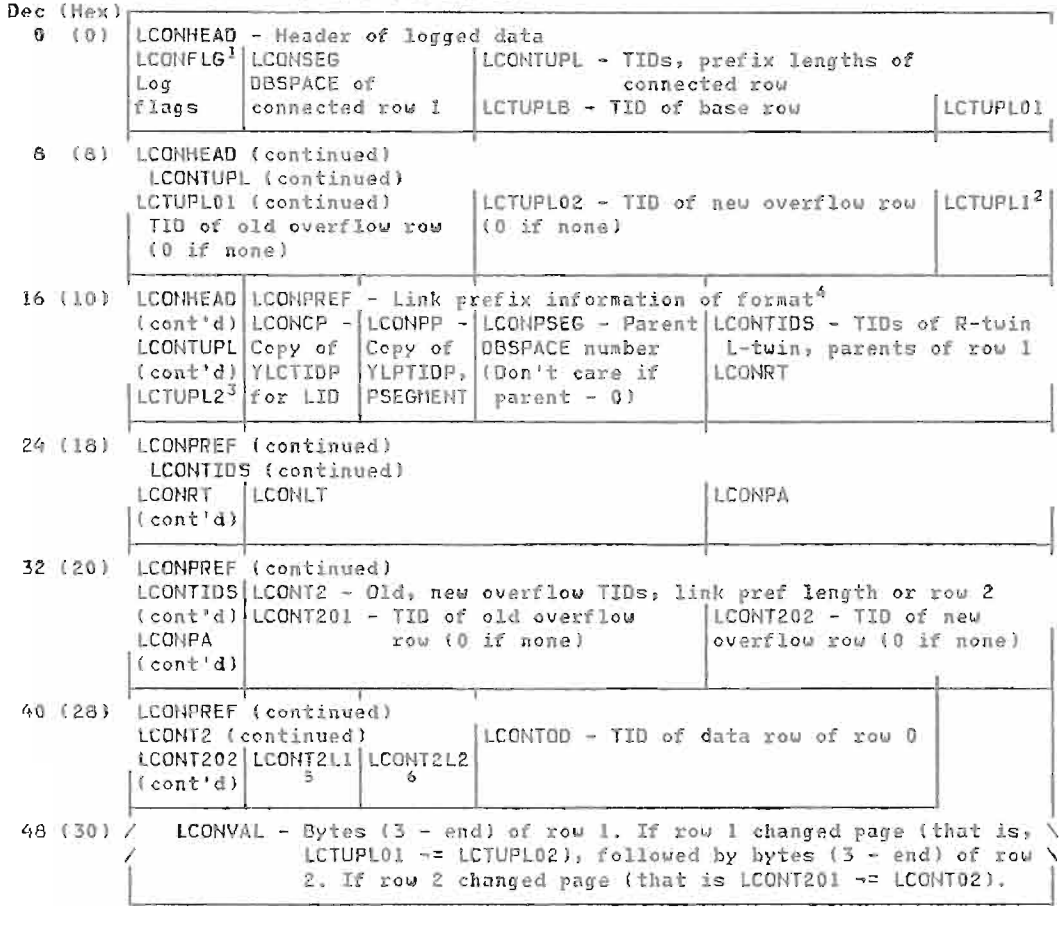
[4] Old, new overflow TIDs, link pref lengths of row 2.

[5] Length of link prefix of old row

[6] Length of link prefix of new row

DATA FOR CONNECT

LCONNECT - Log data for CONNECT



[1] Log flags for row:
1... .... If child entity, page lock in backup
.1.. .... If parent entity, page lock in backup
..1. .... Don't care
...1 .... Don't care
.... 1... If row 1 expanded but not connected
.... .1.. If row 2 is a parent
.... ..1. If row 2 us a left twin
....x Reserved

[2] Length of link prefix of old row
[3] Length of link prefix of new row
[4] Present if fifth bit of LCONFLG = 1
[5] Length of link prefix of old row
[6] Length of link prefix of new row

LDELETE - Log data for DELETE

Dec (Hex)					
0 (0)	LDELHEAD - Header of logged data				
	LDELFLG[1] Log flags	LDELSEG DBSPACE of deleted row	LDELTIDB - Base TID of deleted row		LDELRID RID of deleted row
8 (8)	LDELHEAD (continued)				
	LDELRID (cont'd)	LDELTIDO - Overflow (or 0 if none) TID of row		LDELHD4 - Bytes (3 - 6) of prefix of data row	
				LDELLTH - Total length of deleted data row	LDELRCD Relation code of data row
16 (10)	LDELHEAD (cont'd)	LDELPREF - Link prefix information similar to LDISPREF. (Present if fifth bit of LDELFLG = 1)			
	LDELPRL Length of row link prefix	LDELCP Copy of YLCTIDP for LID	LDELPP Copy of YLPTIDP, PSEGMENT	LDELPSEG - Parent DBSPACE number (Don't care if LDISPA = 0)	LDELTIDS - TIDs of R-twin L-twin, parent of row LDELRT (R-twin)
24 (18)	LDELPREF (continued) LDELTIDS (continued)				
	LDELRT (cont'd)	LDELLT (L-twin)			LDELPA (parent)
32 (20)	LDELPREF (continued)				
	LDELTIDS (cont'd) LDELPA (cont'd)	LDELTIDR - TID of data row of right twin			LDELTIDL - TID of data row or left twin or parent
40 (28)	LDELPREF (cont'd) LDELTIDL (cont'd)	LDELVAL - Field values of deleted row			

[1] Log flags:

```
1... .... If child entity page
          lock in backup
.1.. .... If parent entity page
          lock in backup
          lock in backup
..1. .... If index page lock in
          backup
...1 .... If any index exists
.... 1... If row being disconnected
.... .xxx Reserved
```

DATA FOR DISCONNECT

LDISCONN - Log data for DISCONNECT

Dec (Hex)					
0 (0)	LDISHEAD - Header of logged data				
	LOISFLG[1] Lock protocol flags	LDISSEG - DBSPACE number of child	LDISTID - Base TID of disconnected row		LDISTIDD TID of data and disconn row
8 (8)	LDISHEAD (continued) LDISTIDD (continued)	LDISPREF - Link prefix information of format Similar to LDELPREF			
		LDISCP Copy of YLCTIDP for LID	LDISPP Copy of YLPTIDP, PSEGMENT	LDISPSEG - Parent DBSPACE number (Don't care if LDISPA = 0)	LDISTIDS LDISRT (R-twin)
16 (10)	LDISPREF (continued) LDISTIDS - TIDs of R-twin, L-twin and parent of row (continued)				
	LDISRT (continued)	LDISLT (L-twin)			LDISPA (parent)
24 (18)	LDISPREF (continued) LDISTIDS (continued) LDISPA (continued)	LDISTDR - TID of data row of right twin			LDISTDIL (cont'd)
32 (20)	LDISPREF (continued) LDISTDIL - TID of data row of left twin or parent				

[1] Lock Protocol flags:
1... .... If page lock needed for child
.1.. .... If page lock needed for parent
.... 1... Link prefix data is present
..xx .xxx Reserved

DATA FOR UPDATE

LUPDATE - Log data for UPDATE

Dec (Hex)					
0 (0)	LUPDHEAD - Header of logged data				
	LUPDFLG[1] Log flags	LUPDSEG DBSPACE of updated row	LUPDTID - Base TID of updated row		LUPDRID RID of updated row
8 (8)	LUPDHEAD (continued)				
	LUPDRID (cont'd)	LUPDTID1 - Old overflow TID. 0 if none		LUPDTID2 - New overflow TID. 0 if none	
16 (10)	LUPDHEAD (continued)				
	LUPDTID2 (cont'd)	LUPDLTH1 - Length of old subrow	LUPDLTH2 - Length of new subrow	LUPDBEG - Displ. in row of 1st updated byte	LUPDIND[2]
24 (18)	LUPDDOMS - Old, new subrows				

[1] Log flags:
1... .... If entity page locks in backup
.1.. .... Don't care
..1. .... If index page lock in backup
...1 .... If any index has to be updated
.... 1... If any index after 8 is updated
.... .1.. If any domain changed length
.... ..xx Reserved

[2] Flags describing whether any of the first 8 indexes were updated. Bit I = 1 if the the Ith index was updated.

DATA FOR LIST

LLISTINS - Log data for LIST INSERT

Dec (Hex)				
0 (0)	LLSTHEAD - Header of logged data			
	LLSTSEGM - DBSPACE of list row	LLSTPAGE - Page of list row	LLSTDISP - Displacement of inserted list row	LLSTTLTH Length of inserted list row
8 (8)	LLSTHEAD (cont'd) LLSTTLTH (cont'd)	LLSTDOMS - Inserted field values of list row		

DATA FOR CINSERT OF AN INDEX CONTROL RECORD (ICR)

CINSERTI - Log data for CINSERT

Dec (Hex)						
0 (0)	CIHEADI - Header of logged data					
	CILFLGSI Log [1] flags	CISEGNI - DBSPACE number	CILASTI - Previous index ID	CINEWIDI - New index ID		CIRIDI ↑ to master control record
8 (8)	CIHEADI (continued)					
	CIRIDI (cont'd)	CIDOMIDI - ↑ to domain control record	CILTHNI - Length of new index control record	CILTHDI - Length of domain control record		CINEWI - CIDOMNI Number of domains in index
16 (10)	CINEWI - New index record					
	CIRELI - Back ↑ to master control record	CINEXTI - ↑ to next index record	CICODEI Ordinal number of index	Reserved	CIFLAGSI [2]	CIFREEI Percent. of free space at load
24 (18)	CINEWI (continued)					
	CILENI Length for fixed length case	CIROOTPI - Index root page	CIDOMSI - Domain Number with sign: + For ascending order - For descending order			

[1] Log flags for index row:
xxx. .... Unused
...1 .... If indexed
.... xxxx Unused

[2] Flags:
(CIUNIQI) 1... .... If unique key
(CIFIXLI) .1.. .... If fixed length keys
(CITYPEI) ..00 .... If type 1 table
..01 .... If type 2, unary
..11 .... If type 2, binary
(CILOCKI) .... 1... Fine key interval locking
.... 0... Page locking at leaf
(CIORDERI) For variable length keys
.... .0.. Short keys are high
.... .1.. Short keys are low
(CIASCDI) .... ..0. If all ascending domains
Reserved x

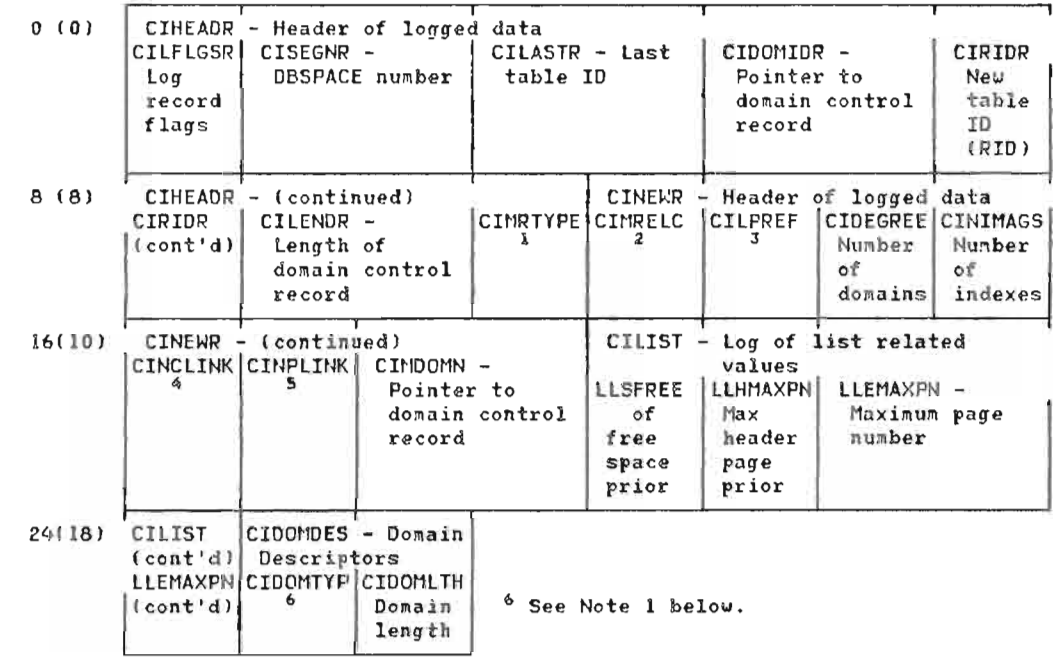
CINSDOMI - Log data describing domains

Dec (Hex)		
0 (0)	CIDOMDSI - Domain Descriptors	
	CIDOMTYI [1]	CIDOMLI Domain length

[1] Domain type:
CIDFIXLI 0... .... If variable length domain
CIDINDXI .0.. .... If indexed domain
..xx xxxx Unused

DATA FOR CINSERT OF A TYPE 1 TABLE CONTROL RECORD OR LIST CONTROL RECORD (MCR)

Dec(Hex) CINSERTR - Log data for CINSERT of a table



1 Record type:
'R' for table or 'L' for list
2 Table ordinal (stored in rows)
3 Size in bytes of link prefix in table rows
4 Number of links in which table has child rows
5 Number of links in which table has parent rows

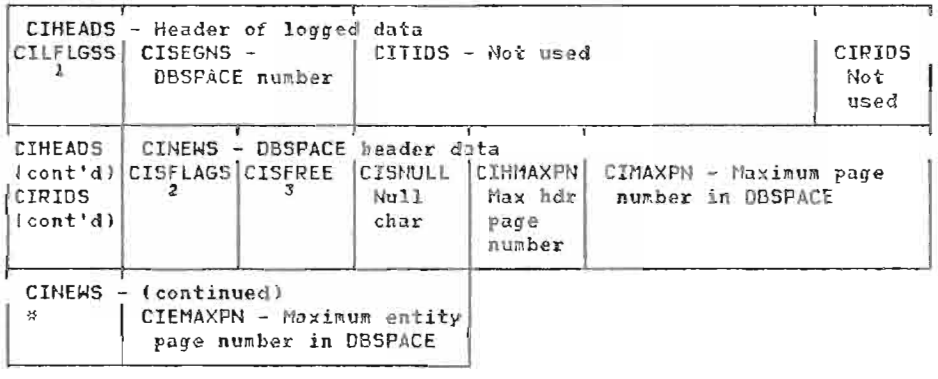
Note 1:

CIDOMTYP - Domain type

CIDFIXLT	0...	If variable length domain
CIDINDEX	.0..	If indexed domain
*	..xx xxxx	Reserved

DATA FOR CINSERT OF DBSPACE HEADER CONTROL RECORD (SCR)

Dec(Hex) CINSERTS - Log data for CINSERT of a DBSPACE header control record



[1] Log flags for header row

[2] See Note 1 below.

[3] Percentage of free entity space at load

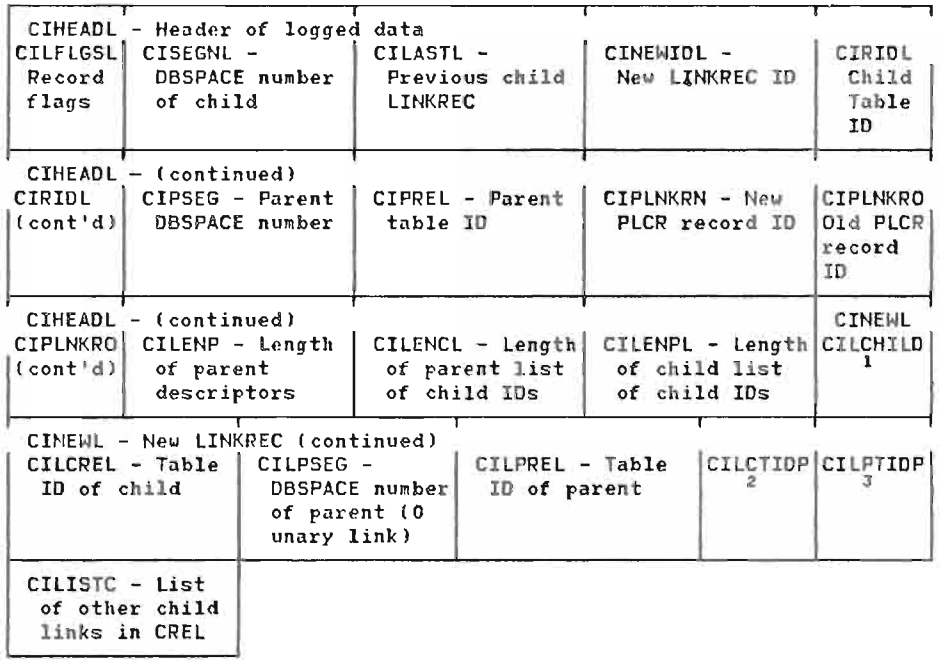
Note 1

CISFLAGS:

CITLOCK	0...	If entity page log lock
CISGLOCK	..1.	If DBSPACE locking
	.x.x xxxx	Reserved

DATA FOR CINSERT OF A LINK CONTROL RECORD (PLCR OR LCR)

Dec(Hex) CINSERTL - Data for CINSERT of a link control record

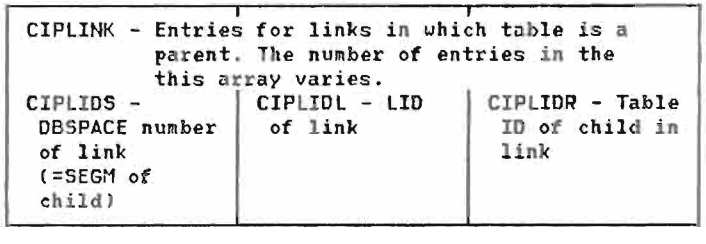


[1] Length prefix of child used in open of a parent to allocate SCB space

[2] Index of first TID pointer in child vector of TID pointers. TID pointers are consecutive in this order: right twin, left twin, parent.

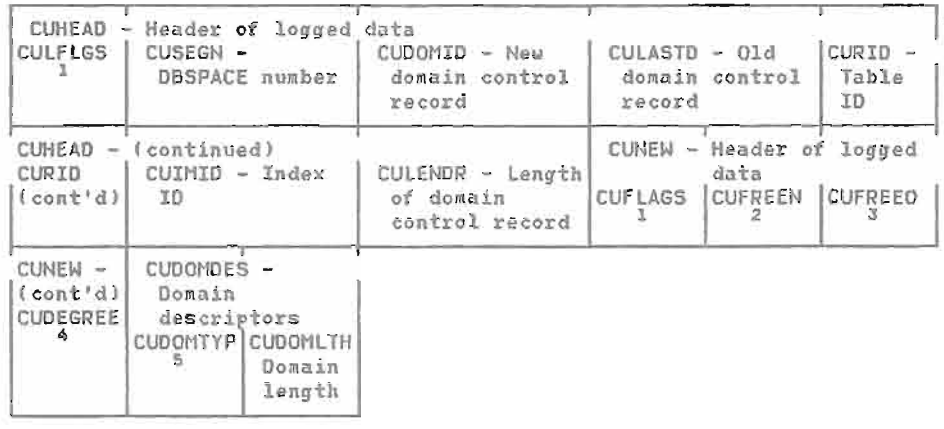
[3] Index of first TID pointer in parent vector of TID pointers

Dec(Hex) CIPLINKS



DATA FOR UPDATE OF HEADER CONTROL RECORD

Dec(Hex) CUPDATE - Log data for UPDATE of header control record



1 See Note 1 below.

2 Percentage of free entity space - new value

3 Percentage of free entity space - updated value

4 Number of domains before updating

5 See Note 1 below.

Note 1:

CULFLGS: Log record flags

CUSEG	.... 1...	If DBSPACE header control update
CUREL	.... .1..	If table control record update
CUIMAG	.... ..1.	If index control record update
*	xxxx ...x	Reserved

CUFLAGS: DBSPACE header and index control record flags

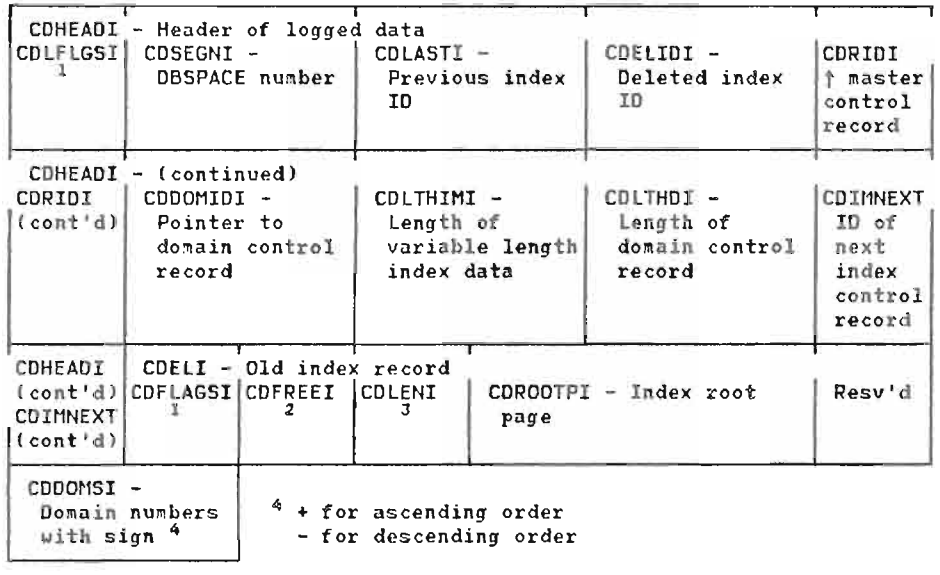
CUTLOCKN	0...	If entity page logical locking
CUTLOCKO	.1..	TLOCK before update
CUILOCKN	..1.	If key locking
	..0.	If page locking
CUILOCKO	...1	ILOCKING before update
CUSGLOKN	.... 1...	If DBSPACE locking
CUSGLOKO	.... .1..	SGLOCK before update
*	xx	Reserved

CUDOMTYP: Domain type

CUDFIXLT	0...	If variable length domain
CUDINDEX	.0..	If DINDEXED domain
*	..xx xxxx	Reserved

DATA FOR CDELETE OF AN INDEX CONTROL RECORD (ICR)

Dec(Hex) CDELETEI - Log data for CDELETE of index control record



[1] See Note 1 below.

[2] Percentage of free space at load

[3] Length for fixed length case

[4] + for ascending order
- for descending order

Note 1:

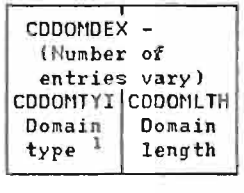
CDLFLGSI: Log flags for index row

	...1	If indexed
	xxx. xxxx	Unused

CDFLAGSI:

CDUNIQI	1...	If unique key
CDFIXLI	.1..	If fixed length keys
CDTYPEI	..00	If Type 1 table
	..01	If Type 2, unary
	..11	If Type 2, binary
CDLOCKI	.... 1...	Fine key interval locking
	.... 0...	Page locking at leaf
CDORDERI	.... .0..	Short keys are high
	.... .1..	Short keys are low
CDASCDI	.... ..0.	If all ascending domains
*	x	Reserved

Dec(Hex) CDESCRPT

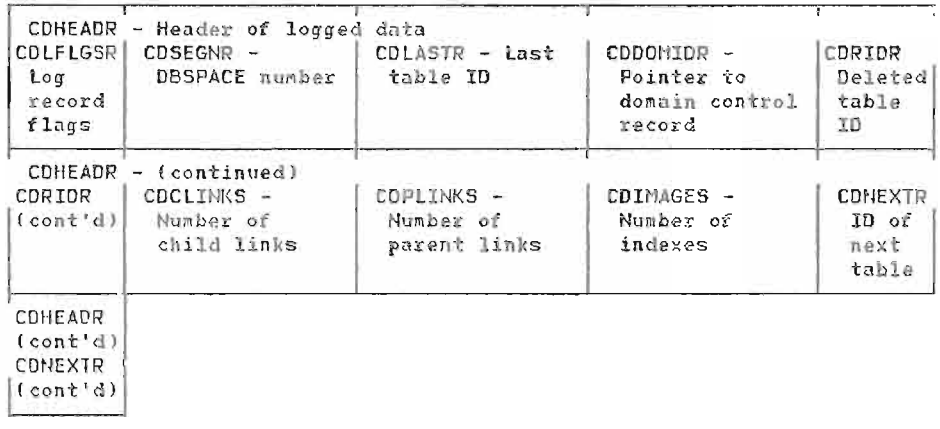


[1] CDDOMTYI:

CDDFIXLI	0...	If variable length domain
CDDINDXI	.0..	If indexed domain

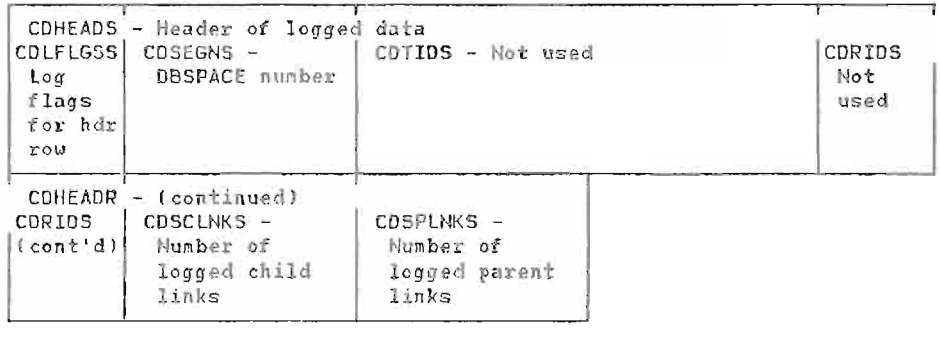
DATA FOR CDELETE OF A TYPE 1 TABLE RECORD (MCR)

Dec(Hex) CDELETER - Log data for CDELETE of a table record



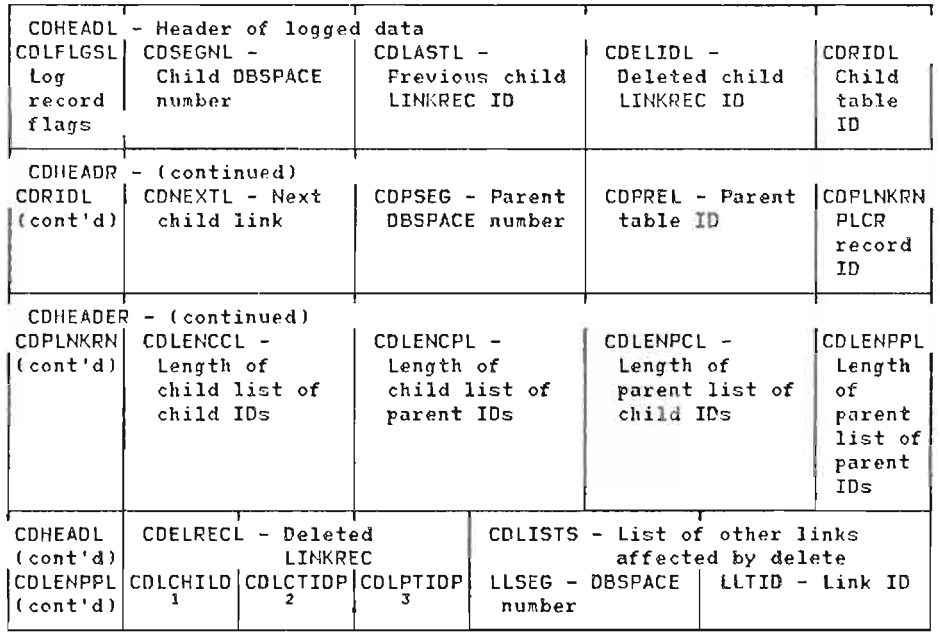
DATA FOR CDELETE OF A DBSPACE HEADER CONTROL RECORD (SCR)

Dec(Hex) CDELETES - Log data for CDELETE DBSPACE header control record



DATA FOR CDELETE OF LINK CONTROL RECORD (LCR, PLCR)

Dec(Hex) CDELETEL - Log data for CDELETE of a Link Control Record



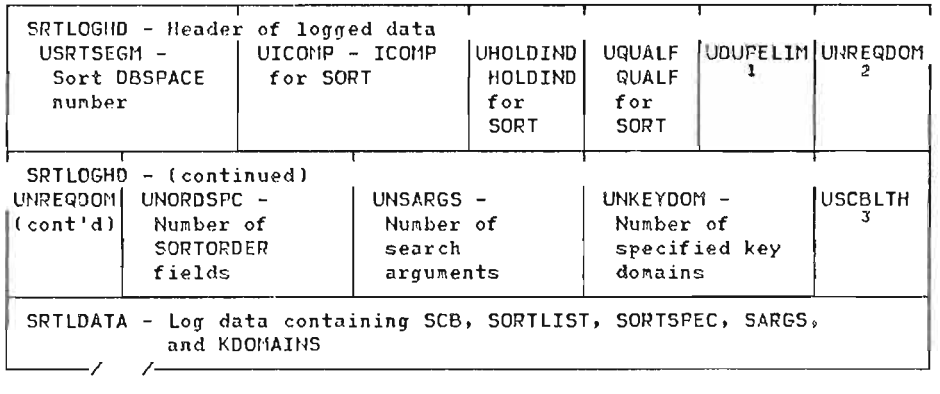
[1] Length of prefix of child used in open of a parent to allocate SCB space

[2] Index of first TID pointer in child vector of TID pointers. TID pointers are consecutive in this order: right twin, left twin, parent.

[3] Index of the first TID pointer pointer in the parent vector of the TID pointers.

DATA FOR UNDO/REDO OF A SORT

Dec(Hex) SRTLOGTP - Log data for SORT

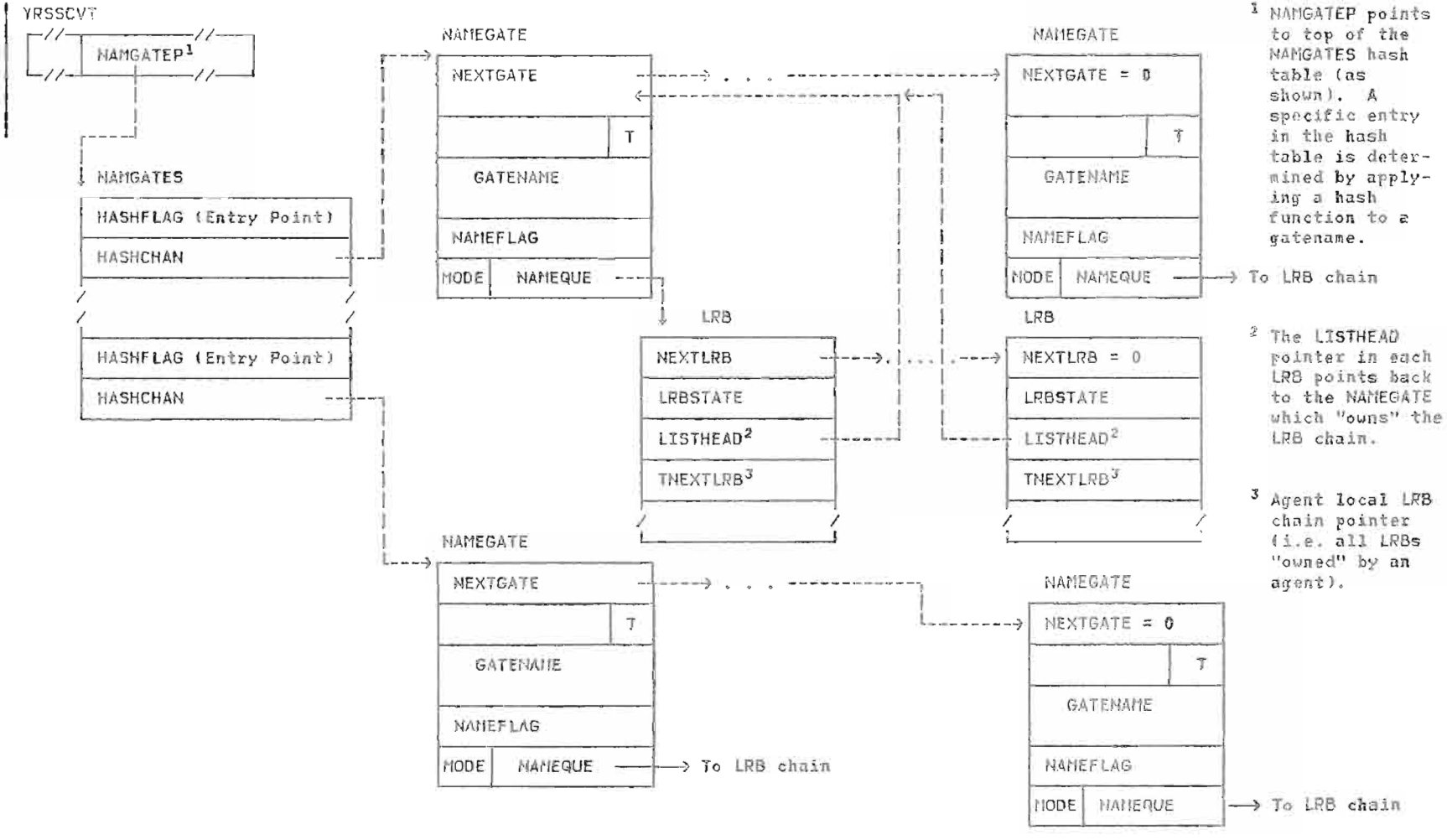


[1] Duplicate elimination option

[2] Number of SORTLIST domains

[3] Length of logged Scan Control Block

LOCK CONTROL BLOCK INTERCONNECTIONS



[1] NAMGATEP points to top of the NAMGATES hash table (as shown). A specific entry in the hash table is determined by applying a hash function to a gatename.

[2] The LISTHEAD pointer in each LRB points back to the NAMEGATE which "owns" the LRB chain.

[3] Agent local LRB chain pointer (i.e. all LRBs "owned" by an agent).

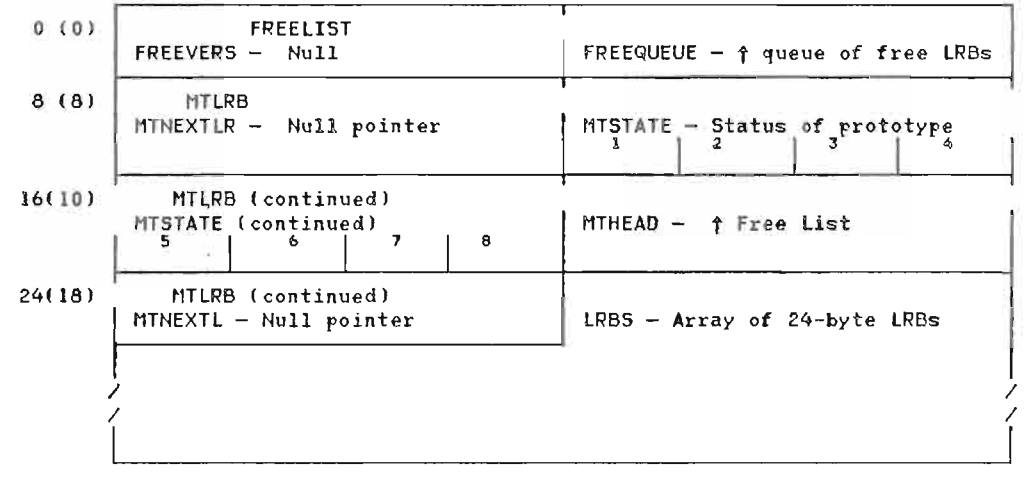
Note: See page 100 for format of an LRB.
See page 101 for the Named Gate.

LRBS (LOCK REQUEST BLOCKS)

The LRB chain consists of a header and an array of LRBs. It is generated by the YLRBS macro. An LRB in the chain may be either a regular LRB or a Named Gate Header. The Named Gate Header and overlay are generated by the YGATEDEC macro.

This subsection shows the header, an LRB, a Named Gate Header, and a Named Gate. The HEADLRB points to the queue of free LRBs. Each Named Gate Header points to a queue of Named Gates, each of which points to a queue of requests for a resource (LRBs).

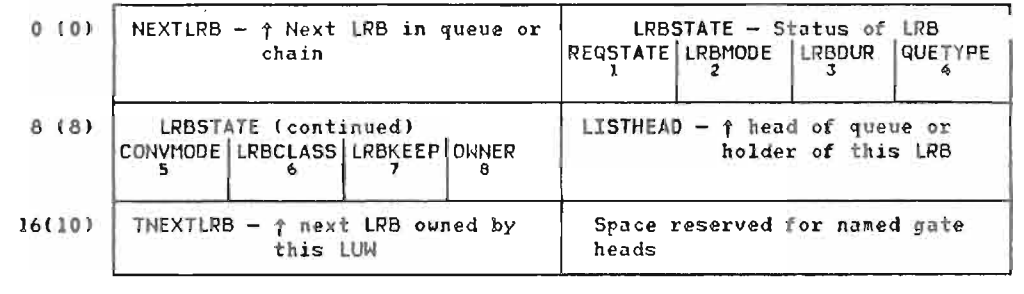
Dec(Hex) HEADLRB (The Prototype (Empty) Header for LRB chain)



1 Free state (= 0)
2 No access mode (= 1)
3 Instant duration (= 1)
4 Type free queue (= 1)
5 Not convert mode (= 1)
6 Class (= 0)
7 Keep (= 0)
8 Owner (= 0)

LRB

Dec(Hex) LRB



1 to 8: See Notes on page 101.

Notes:

[1] REQSTATE:
- 0 = LRB is free.
- 1 = Lock waiting to be granted.
- 2 = Waiting to be converted.
- 3 = Lock request granted.
- 4 = Lock request denied.

[2] LRBMODE:
- 1 = No access
- 2 = Intention share (IS) mode
- 3 = Intention exclusive (IX) mode
- 4 = Share (S) mode
- 5 = Share and intention exclusive (SIX) mode
- 6 = Exclusive (X) mode

[3] LRBDUR:
- 1 = Instant (cancel request when granted).
- 2 = Short (has duration of DBSI call).
- 3 = Duration under user control.
- 255 = Long (lock held to end of LUW).

[4] QUETYPE:
- 1 = In Free List.
- 4 = In Named Gate Queue.
- 5 = Named Gate Header in Hash Chain.

[5] CONVMODE - If converting, the new mode.
[6] LRBCLASS - Count of ARIYKOI calls by the requestor.
[7] LRBKEEP - If "1", keep LRB beyond COMMIT.
[8] OWNER - Index of LRB owner in TPMAP.

Named Gate Header

Dec(Hex) NAMGATES

0 (0)	HASHFLAG - Latch for Hash Chain	HASHCHAN - ↑ Named Gate Queue

Named Gate Overlay for LRBs Used as Named Gate

Dec(Hex) NAMEGATE

0 (0)	NEXTGATE - ↑ next gate in hash chain	Reserved	ISNAMED [9]
8 (8)	GATENAME - [10]		
16(10)	GATEGATE - Queue description NAMEFLAG - Latch on gate queue	NAMEMODE [11]	NAMEQUE - ↑ queue of requests for gate

[9] Type field of LRB = 5 (Gate Header)

[10] Name of Gate. See field GATE in YTABLE1, page 248 for more information.
- Bits (0-15) Lock type.
- Bits (16-31) DBSPACE number.

[11] Mode of granted group

IIFPARM (PARAMETER LIST TO BUILD INPUT MAILBOX)

IIFPARM is a parameter list structure mapped by ARICIIF and is the linkage to ARICIMB to build the Input Mailbox.
The pointer to IIFPARM is passed by ARICMMB.

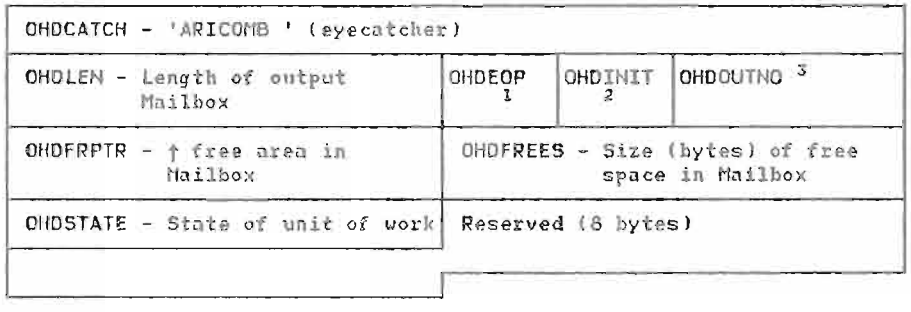
Dec(Hex) IIFPARM

0 (0)	IIFRDINP - ↑ RDIIN area	IIFDBUFP - ↑ Mailbox (default)
8 (8)	IIFDBUFL - Length of Mailbox (default)	IIFOBUFP - ↑ new Mailbox
16(10)	IIFOBUFL - Length of new Mailbox	IIFMSGLN - Length of message in Mailbox
24(18)	IIFPABEG - Partition or virtual machine beginning address	IIFPAEND - Partition or virtual machine ending address

OHDHEAD (OUTPUT MAILBOX HEADER) AND OUTPUT ELEMENTS

An output Mailbox consists of the header (OHDHEAD) and a number of output elements. The output Mailbox is pointed to by OIFDSPTR in the OIFPARM parameter list. Module *ARICOMB* builds the output Mailbox (MPM) for sending data to the user partition or virtual machine

Dec(Hex) OHDHEAD

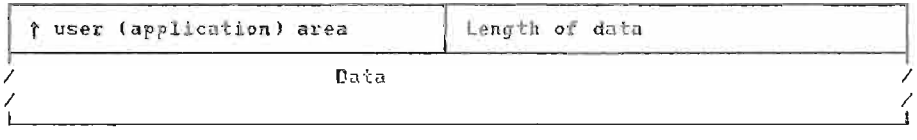


1 End-of-Process flag:
'Y' = End of process
'N' = Not end of process

2 First-time out Mailbox header flag.
'Y' = Initialize; 'N' = Do not.

3 Number of output elements.

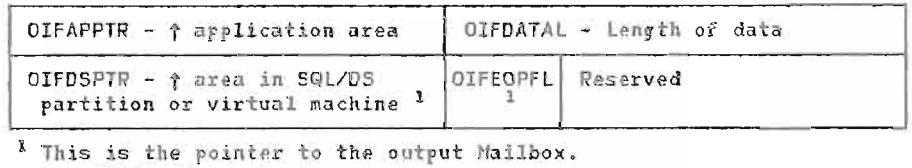
The output elements immediately follow OHDHEAD in the output Mailbox. The format of an output element is:



OIFPARM

OIFPARM is a parameter list structure mapped by ARICOIF. It is pointed to by field DSCOMBPP in the DSCAREA.

Dec(Hex) OIFPARM

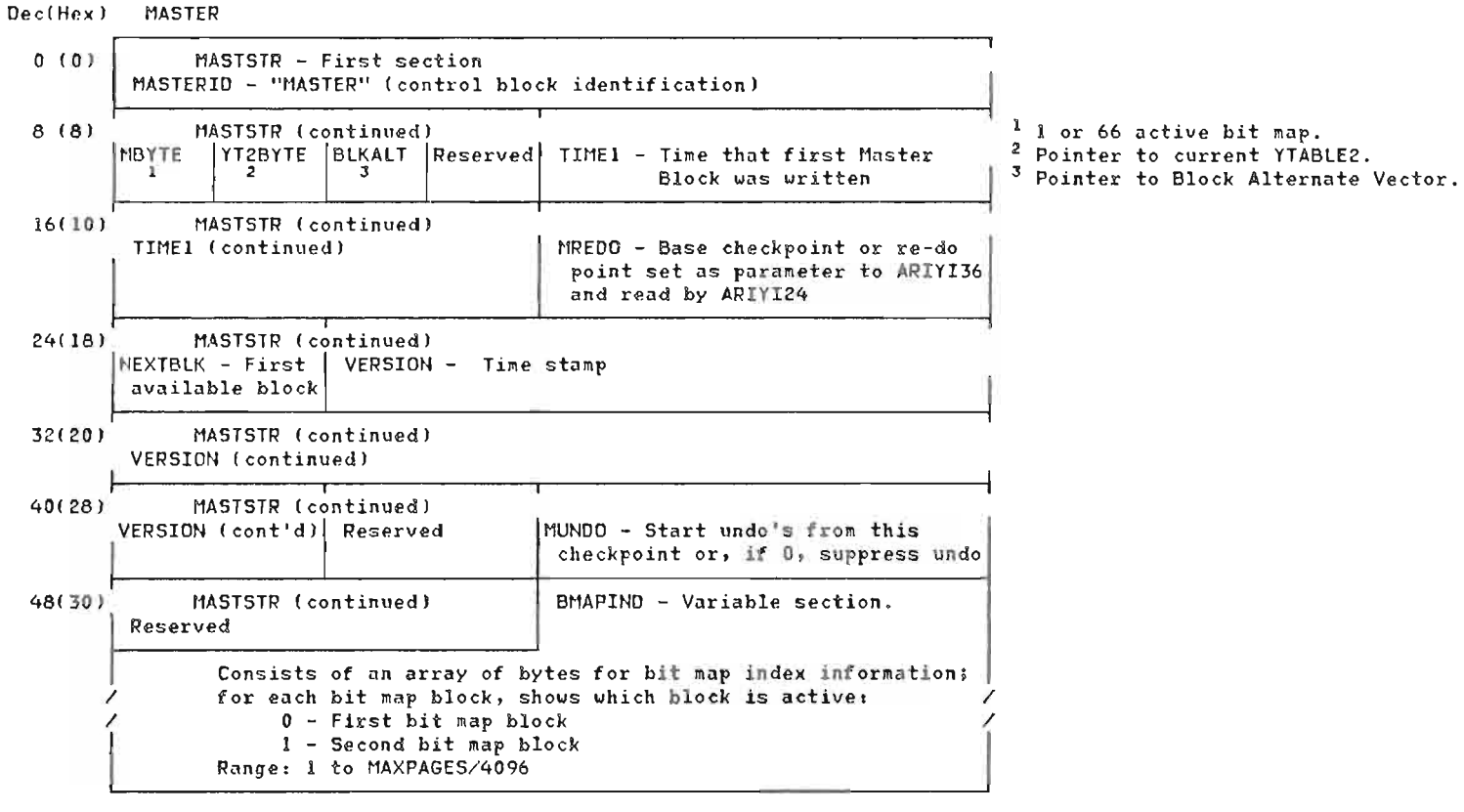


1 This is the pointer to the output Mailbox. See "OHDHEAD ... " above.

1 End-of-Process flag:
'Y' = End of process
'N' = Not end of process

The Directory Master Record is a composite of a fixed section called MASTER and two variable-size sections, BMAPIND and SEGAT. Portions of this table vary in size according to the values of "MAXPAGES" and "MAXSEGS".

Addressability to the various sections is set during DBSS initialization. The anchor pointers are in YRSSCVT (fields MASTERP and MASTEREP).



[1] 1 or 66 active bit map.
[2] Pointer to current YTABLE2.
[3] Pointer to Block Alternate Vector.

MESSAGE FORMATTER DATA AREAS

INPUT PARAMETERS AND THEIR STRUCTURES

The structures MSGID, VARLIST, and BUFFER are the input to and the output from calls to the three message formatter modules:

- ARIYM05 - SQL/DS partition or virtual machine and Resource Managers
- ARIIFOR - ISQL and Extract
- ARIMFMT - Other SQL/DS components, such as, DBSU, PREP, etc.

In these structures, the following abbreviations apply:

(I) = Input data to formatter
(O) = Output data from formatter
(I/O) = Both input and output data

MSGID

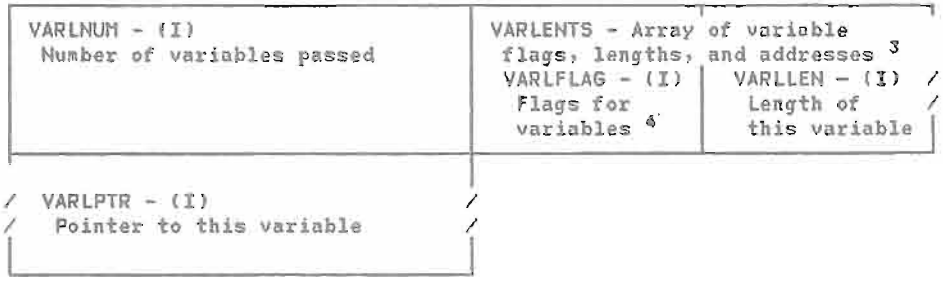
Dec(Hex) MSGID - MSGID structure

[1] SQLTEXT: (I)
1... .... MSGNO is SQL return code
.xxx xxxx Unused

[2] Multi-line sequence number updated by message formatter and used to determine current line to be retrieved and formatted.

VARLIST

Dec(Hex) VARLIST - Structure containing variables for message text insertion

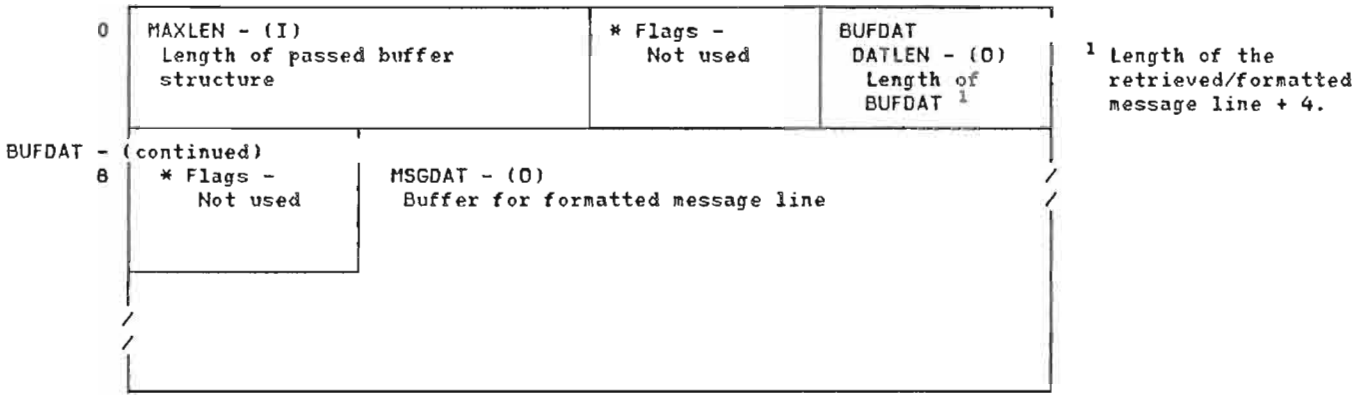


[3] This structure varies in length depending on the number of variables.

[4] VARLNOVR: (I)
NOVAR flag
1... .... Null variable
.xxx xxxx Unused

BUFFER

Dec(Hex) BUFFER - Message buffer structure



MAPPINGS FOR MESSAGE MODULES STRUCTURES

DIRENT

DIRENT is the message directory structure containing all message numbers in the module. Each entry points to either a message sequence structure (multi-line message) or to a message text structure (single line message). ML means multi-line.

DIRENT is generated by the MSGDIR macro: ?MSGDIR(message#|SQLCODE#) [ML], or ?MSGDIR EOF. There is one substructure for each MSGDIR macro and one MSGDIR macro for each message. MSGDIR EOF produces the last substructure terminating the structure.

Dec(Hex) DIRENT - Directory entry array

0	DIRFLG - Entry flags [5]	DIRMSG - Entry message or SQLCODE number or EOF (end of directory indicator)	DIRMPT - Pointer to sequence structure (ML) or to text structure (non-ML)

[5] DIRML:
1... .... ML message
0... .... non-ML message
.xxx xxxx Unused

SEQENT

SEQENT is the message sequence structure containing an entry for each line of a ML (multi-line) message. It is for multi-line messages only. Each entry has a flag for the last line, the line sequence number, and the pointer to the message text structure for that line.

SEQENT is generated by the MSGSEQ macro. This macro creates n repetitions of the shown substructure, where n is the number of lines in the ML message.

Dec(Hex) SEQENT - Message sequence entry array

0	SEQFLG - Entry flags [6]	SEQNR - Line sequence number	SEQMPT - Pointer to text structure for line

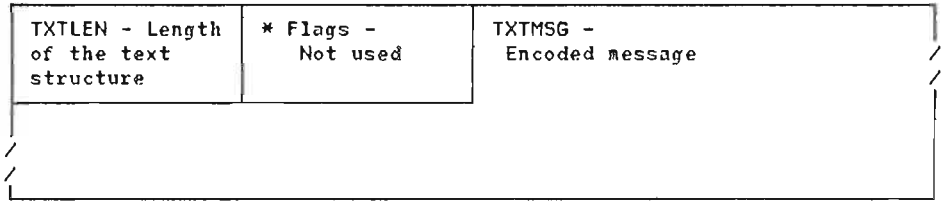
[6] SEQLST:
1... .... Last line of message
.xxx xxxx Unused

TXTENT

TXTENT is the message text structure. It contains the structure length, the flags (unused), and the encoded message text with 0 to n imbedded input variable (from VARLIST) substitution requests in the form &n. or &nalpha., where 'n' specifies the nth variable from VARLIST and 'alpha', if present, indicates conversion (A = binary to decimal, B = binary to hexadecimal).

TXTENT is generated by the MSGFORM macro. The structure is used for each message line. TXTMSG is coded by the message implementor as a level 2 DCL immediately after the MSGFORM macro.

Dec(Hex) TXTENT - Message text structure

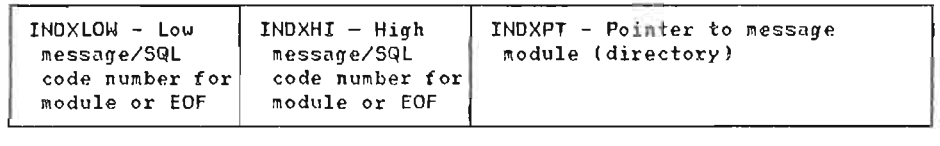


INDXENT

INDXENT is the mapping description of the message index and the SQL index structures. It tells what message numbers (MSGID) or SQL codes (MSGID with SQLTEXT = 1) are in what message text module. INDXENT is an array of low and high message numbers or SQL code numbers paired with text module entry points. An 'EOF' message number ends the array.

Given a message number or a SQL code number, the INDXENT array substructure lets the message formatter find the message module containing the corresponding message. This substructure is repeated (n level array) times, once for each message module. The arrays are imbedded in the message formatter modules.

Dec(Hex) INDXENT - Message or SQL Index entry



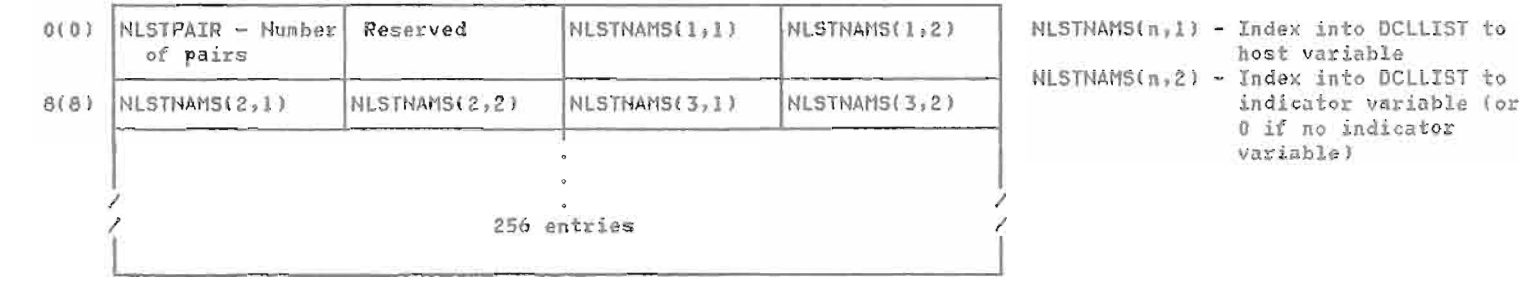
NLST (NAME LIST)

There is one of these lists for input host variables and another for output host variables. They are built by ARIXEPP from IVNAMES/OVNAMES upon return from Parser (ARIXPA1). ARIBNLS maps the Name List. It maps each name in IVNAMES or OVNAMES with an entry in the DCLLIST. The lists are returned to the originating preprocessor PREPDSCB (RETIVARS/RETOVARS and RETSLTPT[1]) where they can be used to indicate which host variables are used for the current SQL statement. They are used by the preprocessor in building SQLDA structures for dynamic statements.

See Volume 1, Section 2, "Parser", for its relationship to the IVNAMES/OVNAMES built by the Parser.

[1] These are pointed to from the AUX SLT entry that is returned to the preprocessor.

Dec(Hex) NLST

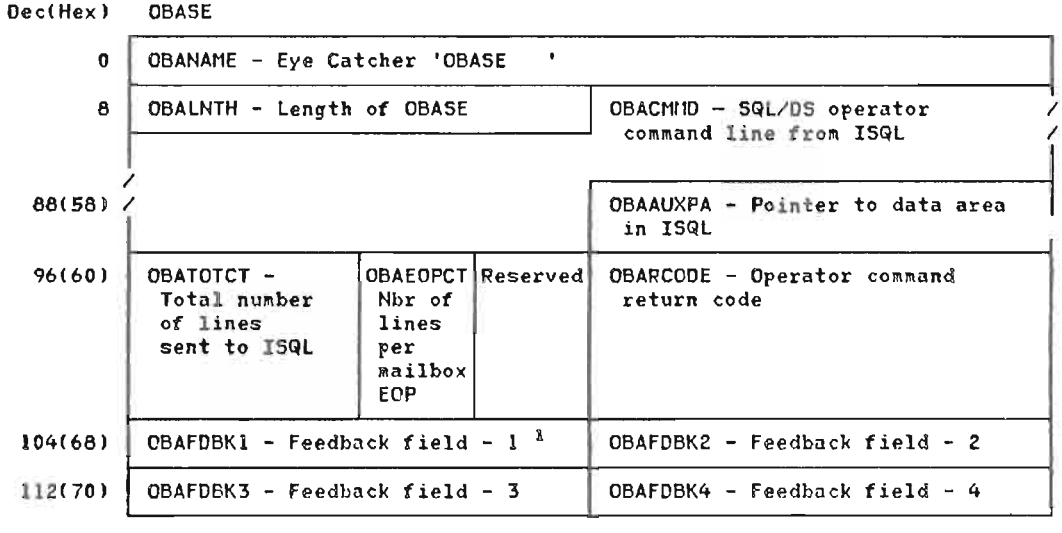


NLSTNAMS(n,1) - Index into DCLLIST to host variable

NLSTNAMS(n,2) - Index into DCLLIST to indicator variable (or 0 if no indicator variable)

OBASE

OBASE is the parameter list for the DBSS operation (operation code 32) process operator command (SQL/DS operator command entered via ISQL). It is the RDS to DBSS linkage to execute the operator commands. On entry to ARIYM00, parameter PARMS points to OBASE. On entry to ARIYM101, parameter OBAPTR points to OBASE.

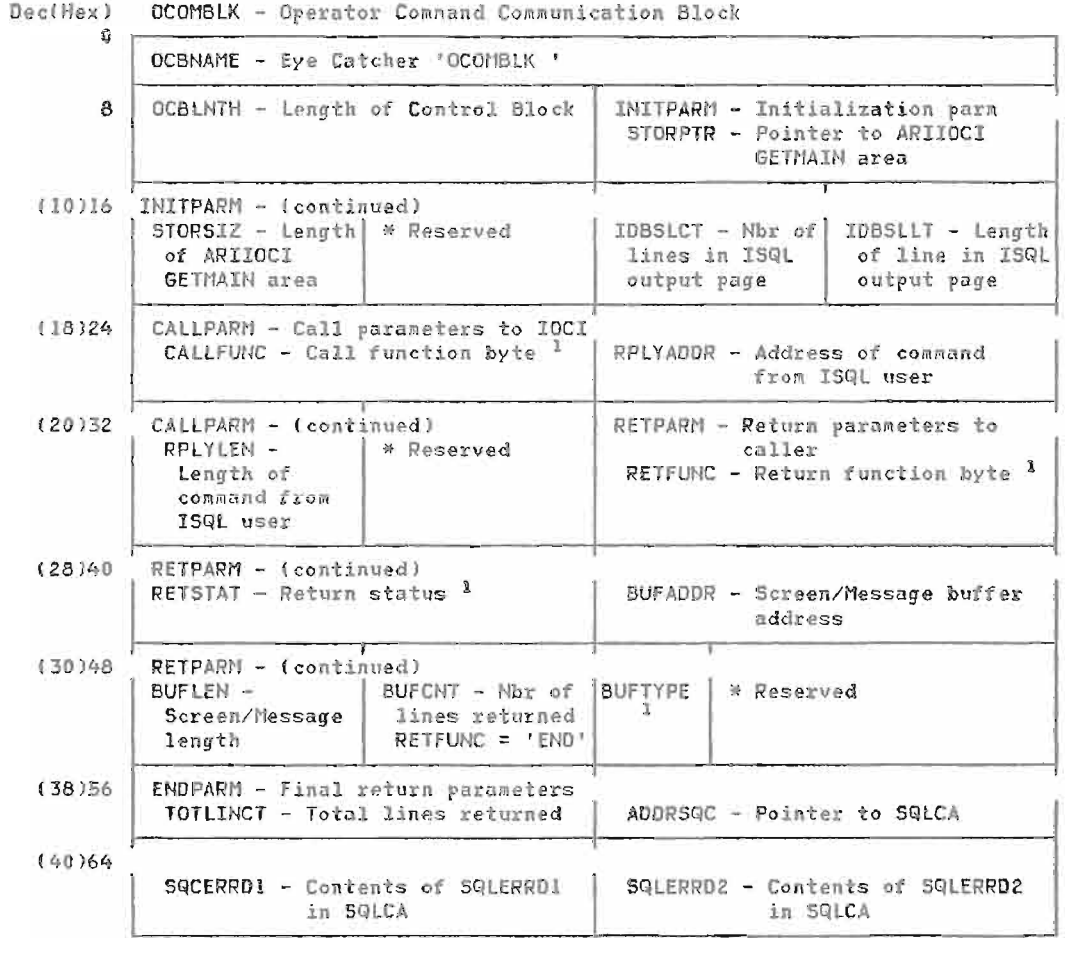


[1] OBAFDBK1 contains a return code if it is an error return code, or it contains a message number if an error message was issued; else 0.

OCOMBLK (OPERATOR COMMAND COMMUNICATION BLOCK)

ARIIOCB is the communication block between ARIICMD and ARIIOCI. It is used to maintain pointers to data needed on subsequent calls to ARIIOCI by ARIICMD, determine what type of processing is to be done by ARIIOCI, and return pointers to operator command information and statuses to ARIICMD.

OCOMBLK exists only during ISQL operator command processing. It is based on the pointer OCCBPTR contained in ARIICMD.



1 See Note 1 on page 112.

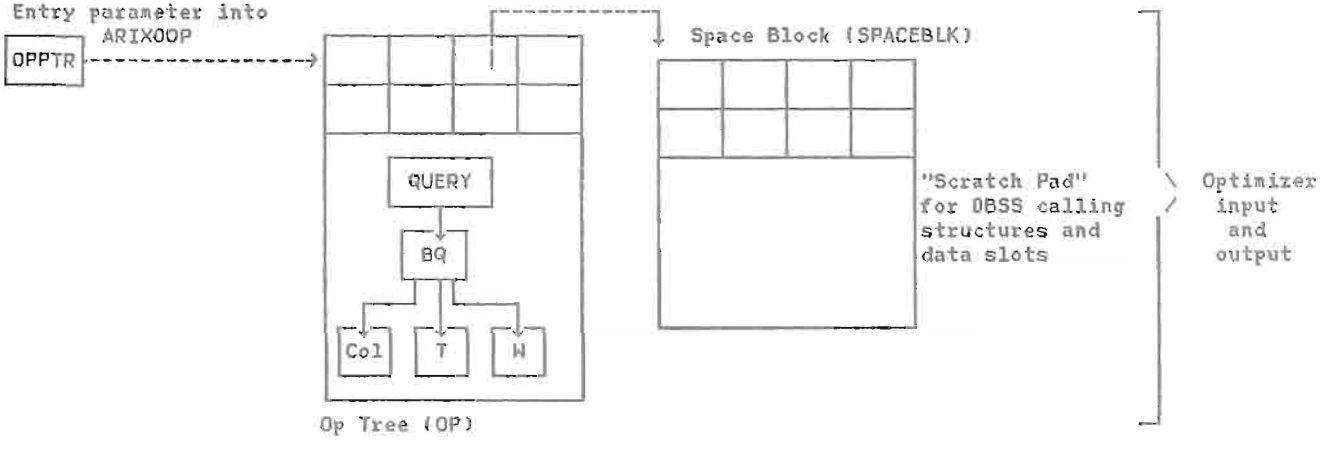
Note 1:

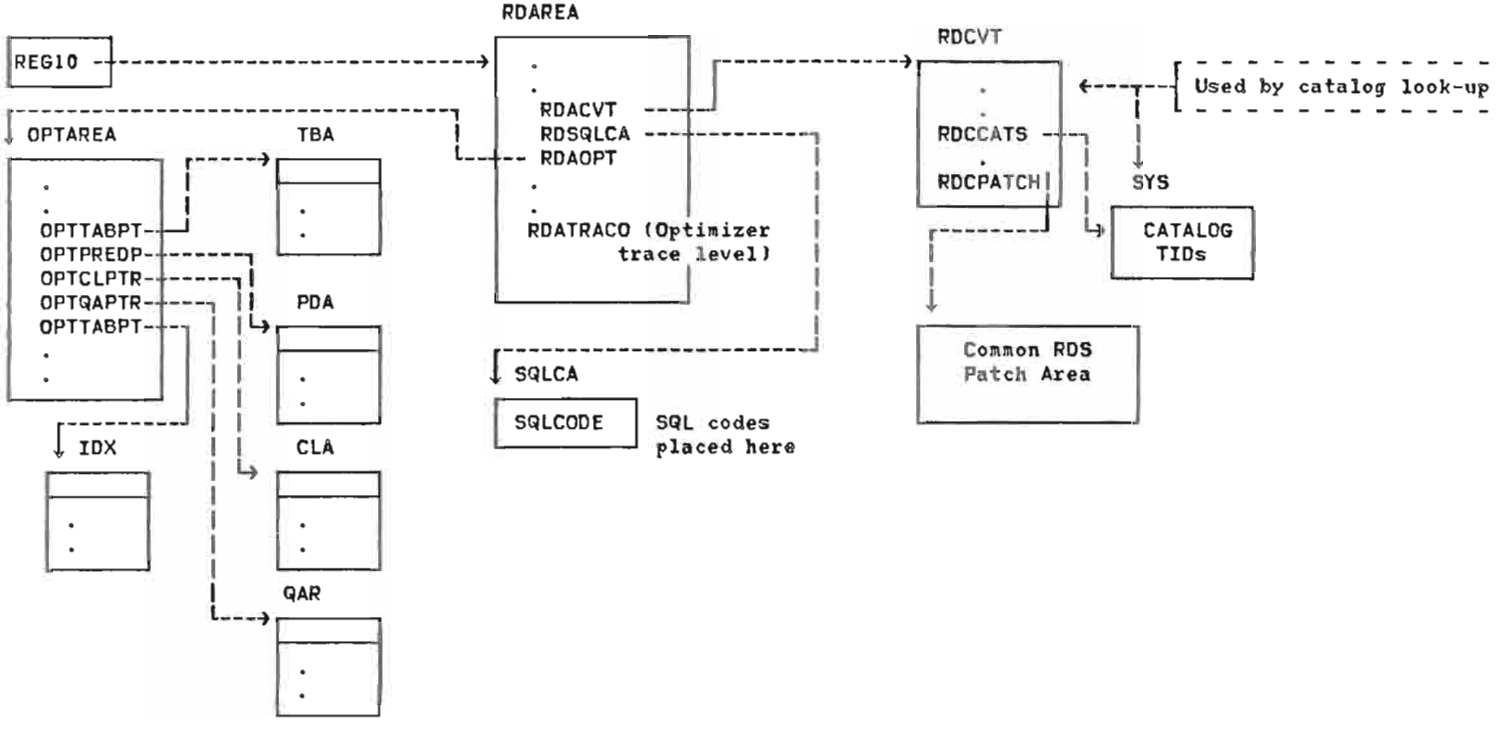
```
CALLFUNC:
   'ONI '  '.OP ON' with init option entered
   'ON  '  '.OP ON' entered
   'CMD '  Command entered, get first page
   'CONT'   Get next page of data
   'CAN '  Cancel ('END') request
   'OFF '  '.OP OFF' entered, command display complete
   'OFFA'  '.OP OFF' entered, command display incomplete
           (implicit 'END')

RETFUNC:
   'DM  '  Display message line
   'DS  '  Display screen
   'DSE '  Display screen (last page of output)
   'NOP '  No screen or message
   'ERR '  Caller error
```

```
RETSTAT:
    0  ok
   -1  SQL/DS error
   -2  CPC error (VSE);
       IUCV error (VM)
   -3  Storage allocation error
   -4  Storage deallocation error
   -5  LUW in progress
   -6  Resource Manager error
   -7  Operator command error, message included
   -8  Operator command error, no message

BUFTYPE:
   'S'  Screen
   'L'  Last screen of data
   'M'  Message
   'R'  Message with reply
   'N'  Null
```

OPTIMIZER DATA AREAS

MAJOR DATA AREAS AND CONTROL BLOCKS USED BY THE OPTIMIZER



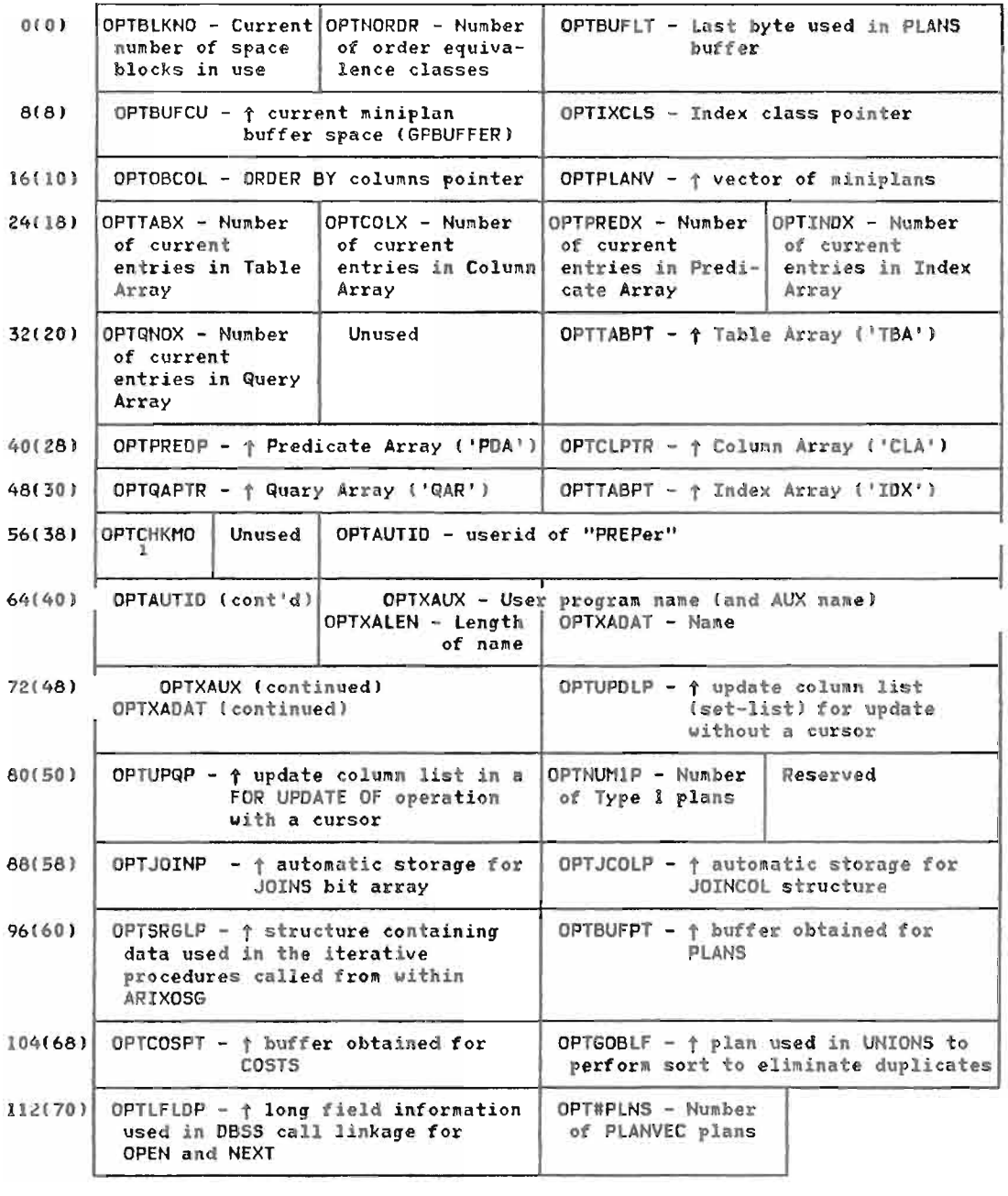


OPTAREA (OPTIMIZER AREA)

OPTAREA is a global data area used throughout an execution thread of the Optimizer.
It is a convenient mechanism to make available information common to the needs of the Optimizer modules.
OPTAREA is established as automatic storage by the Optimizer entry module ARIXOOP.
The OPTAREA is pointed to by field RDAOPT of the RDAREA.

Alphabetic List of Field Names

OPT#PLNS	16(74)	OPTIXCLS	12(C)	OPTRUNMO	56(38) (flag bit)
OPTAUTID	58(3A)	OPTJCOLP	92(5C)	OPTSRGLP	96(60)
OPTBLKNO	0(0)	OPTJOINP	88(58)	OPTTABPT	36(24)
OPTBUFCU	8(8)	OPTLFLDP	12(70)	OPTTABPT	52(34)
OPTBUFLT	4(4)	OPTMKDSF	56(38) (flag bit)	OPTTABX	24(18)
OPTBUFPT	00(64)	OPTNORDR	2(2)	OPTUNION	56(38) (flag bit)
OPTCEXMO	56(38) (flag bit)	OPTNUMIP	84(54)	OPTUPDLP	76(4C)
OPTCHKMO	56(38) (flag bit)	OPTOBCOL	16(10)	OPTUPQP	80(50)
OPTCHKMO	56(38)	OPTPLANV	20(14)	OPTVCMMO	56(38) (flag bit)
OPTCLPTR	44(2C)	OPTPREDP	40(28)	OPTVDFMO	56(38) (flag bit)
OPTCOLX	26(1A)	OPTPREDX	28(1C)	OPTXADAT	68(44)
OPTCOSPT	104(68)	OPTQAPTR	48(30)	OPTXALEN	66(42)
OPTGOBLF	108(6C)	OPTQNOX	32(20)	OPTXAUX	66(42)
OPTINDX	30(1E)				



[1] See OPTAREA flags on next page

OPTAREA Flags

Offset	Field Name	Bits	Meanings
56(38)			
	OPTCHKMO	x...	If set on, error was encountered in a previous SQL statement during PREP, so only do catalog lookup and semantic check.
	OPTRUNMO	.x..	Set on if run (execution) time call to Optimizer.
	OPTCEXMO	..x.	Set on if the AUX (access module) section is an INDEFSEC or if the call type is a SETUPCAL.
	OPTVDFMO	...x	Set on for VIEW definition (or creation).
	OPTVCMMO	.... x...	Set on if VIEW composition is necessary.
	OPTMKDSF	.... .x..	Set on to indicate that this is a UNION and therefore only one call is made to ARIXOMD to build slots for the answer set columns.
	OPTUNION	.... ..x.	Set on if processing a UNION.
	OPTNOFRE	x	Set on so that ARIXOCU will not free storage on calls from the Interpreter.

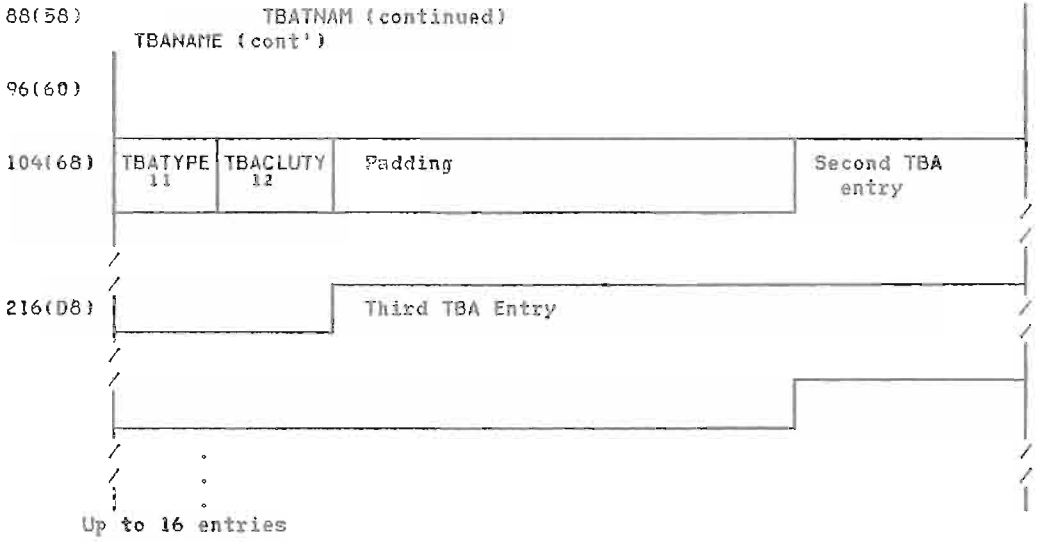
TABLE ARRAY ('TBA')

The Table Array is pointed to by OPTTABPT in OPTAREA. There is a maximum of sixteen table entries. The current number of entries is specified in OPTTABX.

Dec(Hex) TBA

0(0)	TBAQPTR – BQUERY node that contains this table		TBACCPTR – ↑ CATCOL ARRAY for this table	
8(8)	TBACCLLN – CATCOL size in bytes	TBAITBOC – Index in TBA of first occurrence of this table	TBATABIN – Index in Parse Tree of Table node for this table	TBACRATR – Table creator
16(10)	TBACRATR (continued)			Padding
24(18)	TBATID – DBSS Table ID (TID) of SYSCATALOG		TBACLUST – Highest DBSS TID of any row in this table	
32(20)	TBASEG [1]	TBARID – TID of this table	TBANRELS [2]	TBAAVTPL [3]
40(28)	TBACARD – Total number of rows in this table		TBANPAGS – Approximate number of entity pages on which rows of this table appear	
48(30)	TBANCOLS – Number of columns in this table	TBAFINDX [4]	TBALINDX – Index of last entry for this table	TBAPCTPG [5]
56(38)	TBANBLKS – Number of usable pages in this table's DBSPACE		TBAPCTIX [6]	Padding
64(40)	TBASITE – Reserved		TBANPEER – Reserved	
72(48)	TBALFRID [7]	TBALFLID [8]	TBALFSEG [9]	Padding
80(50)	TBATNAM TBAQNO [10]	TBANMLN – Length of table name	TBANAME – Table name	

[1] DBSPACE number in which this table is contained.
[2] Number of tables contained in the DBSPACE in which this table resides.
[3] Average length of rows in this table
[4] Index into IDX of first entry for this table. The remaining index entries follow in order.
[5] Percentage of total active pages in this table's DBSPACE that have rows from this table in them.
[6] Percentage of pages in this table's DBSPACE used for indexes.
[7] DBSS TID of relation (table) containing long-field data. Zero if no long fields.
[8] Link ID (LID) of unary link for long field values. Zero if no long fields.
[9] DBSPACE number for long-field data relation. Zero if no long fields.
[10] Index into QAR (query array) of query to which this table entry belongs.

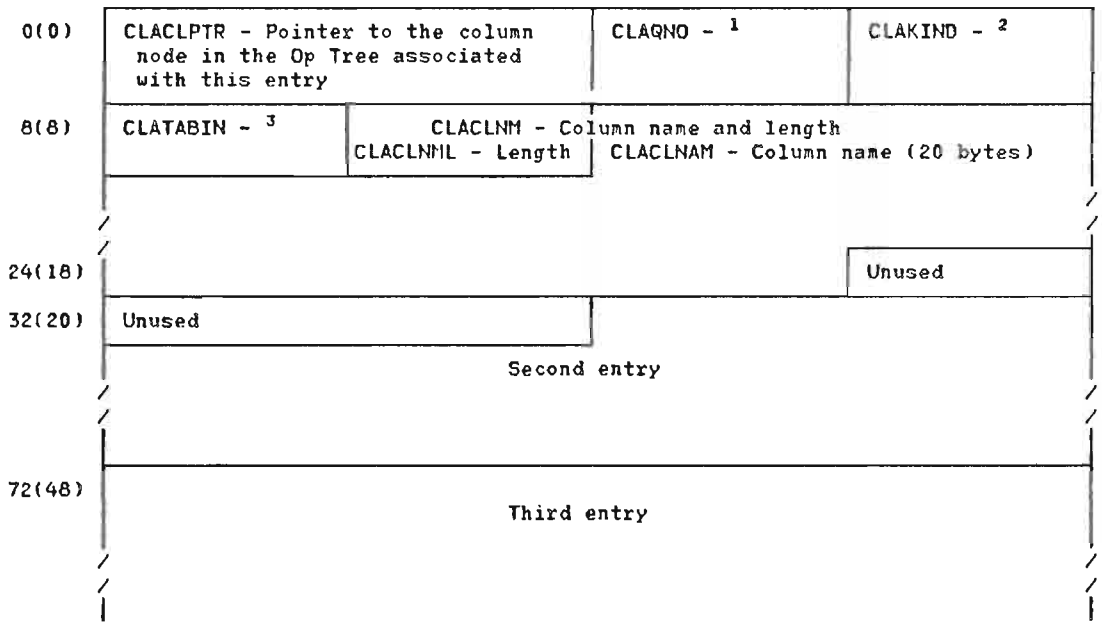


11 'R' if real table; 'V' if view

12 'I' if clustering object is an index
'D' if clustering is to be done by
' ' default

COLUMN ARRAY ('CLA')

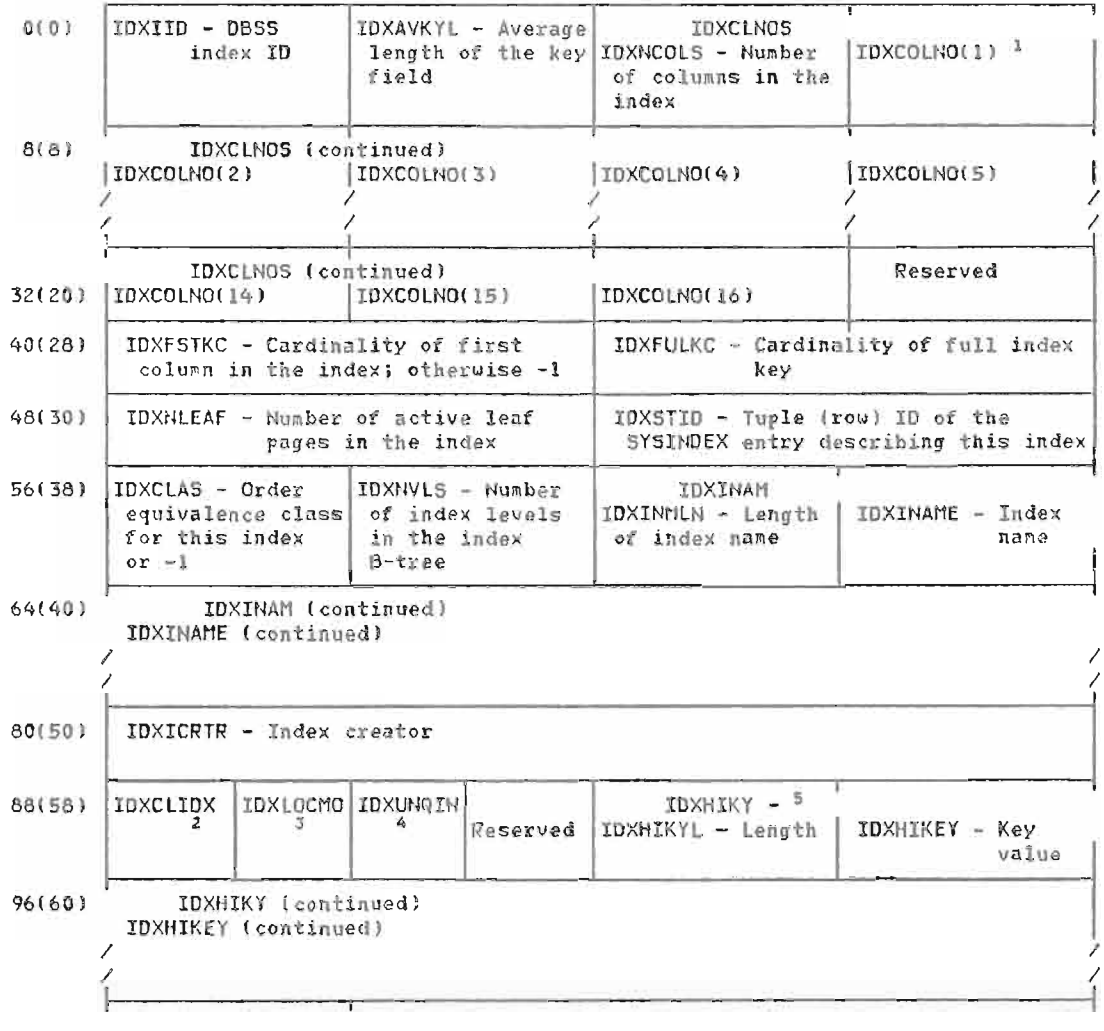
The Column Array is pointed to by field OPTCLPTR in OPTAREA. There is a maximum of 255 table entries. The current number of entries is held in field OPTCOLX of OPTAREA. Each entry is 36 bytes long. Only the first entry is shown here. The second starts at offset 36(24), the third at 72(48), and so on.



1 Index of the query entry in QAR in which this column's table appears.
2 How the column is used: output, predicate, object of built-in function, input, update, insert.
3 Index of the table entry in TBA to which this column belongs or is associated with by qualified name.

INDEX ARRAY ('IDX')

The Index Array is pointed to by field OPTINPTR in OPTAREA. There is a maximum of 150 entries. The current number of entries is shown in field OPTINDX of OPTAREA. Each entry is 128 bytes long. Only the first entry is shown here. The second starts at offset 128(80), the third at 256(100), and so on.



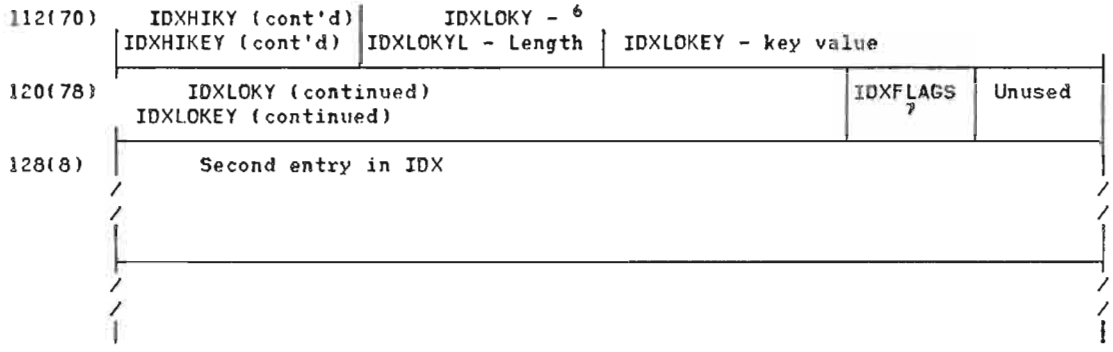
1 The column number of the column contained in the key of the index. These columns appear in left-to-right order.

2 'Y' if clustered; otherwise 'N'

3 'K' if key-interval locking is performed on this index; 'P' if it is pages of the index that are locked

4 'D' = Duplicates allowed
'U' = Unique

5 Second highest column value for the index key



6 Second lowest column value for the index key

7 IDXDELET (bit 0) set "off" by ARIXOCA; set on by ARIXOGP if this index is not used in this access path.

QUERY ARRAY ('QAR')

The Query Array is pointed to by field OPTQAPTR in OPTAREA. There is a maximum of 8 table entries. The current number of entries is shown in field OPTQNOX of OPTAREA. Each entry is 56 bytes long. Only the first entry is shown here. The second starts at offset 56(38), the third at 112(70), and so on.

0(0)	QARBQIND - Parse tree index of BQUERY node for this query	QARPAQNO - Index in query array of the parent query	QARPABTX - Index in predicate array of the predicate that contains this query	QARPAPAX - [1]
8(8)	QARPRED1 - Index of the first entry in predicate array for this entry	QARNPRED - Number of entries in predicate array for this query	QARTBNO1 - Index in table array of the first table entry for this query	QAR#TBLS - Number of table array entries for this query
16(10)	QARASLX - [2]	QARNOPEN - [3]	QARLPTPT - † end of an SQ sequence if done at open of another query	Reserved
24(18)	QAROCLPT - † open check list chain		QARESCOS - Estimated cost of query (in 4-byte internal 370 float representation)	
32(20)	QARESTMP - Estimated cost to retrieve query result from a temp (in 4-byte internal 370 float representation)		QARESCAR - Estimated cardinality of the query result (in internal 370 float representation)	
40(28)	QARONOPN [4] / Flags [5]	QARPRDTR [6]	Unused / QARUNIQ [7] / QARQRYRF [8]	
48(30)	QARCOSTP - Cost of row update or delete		QARCOSIX - Cost of index maintenance under updates or deletes	
56(38)	Second entry			
112(70)	Third entry			

[1] Index in predicate array of the predicate that directly contains this query. This is the same as QARPABTX if the boolean type is not an 'OR' tree.

[2] Index in the parse tree of start of root sequence of ASL tree for this query if done at open of

[3] Number of query blocks that are not to be done at open of this query block. Equals number of 1-bits in the QARONOPN bit string. Set by ARIXOGP.

[4] 8-bit string. Bit position is 1 to signify a corresponding query is to be done at open time of this query. Leftmost bit represents the first query in the query array.

[5] See QAR flags on next page.

[6] 16-bit string containing table reference information used for correlations to next higher query. Copied to the predicate entries for PDATBLRF.

[7] K = duplicates permitted; N = not premitted. Used for sorting.

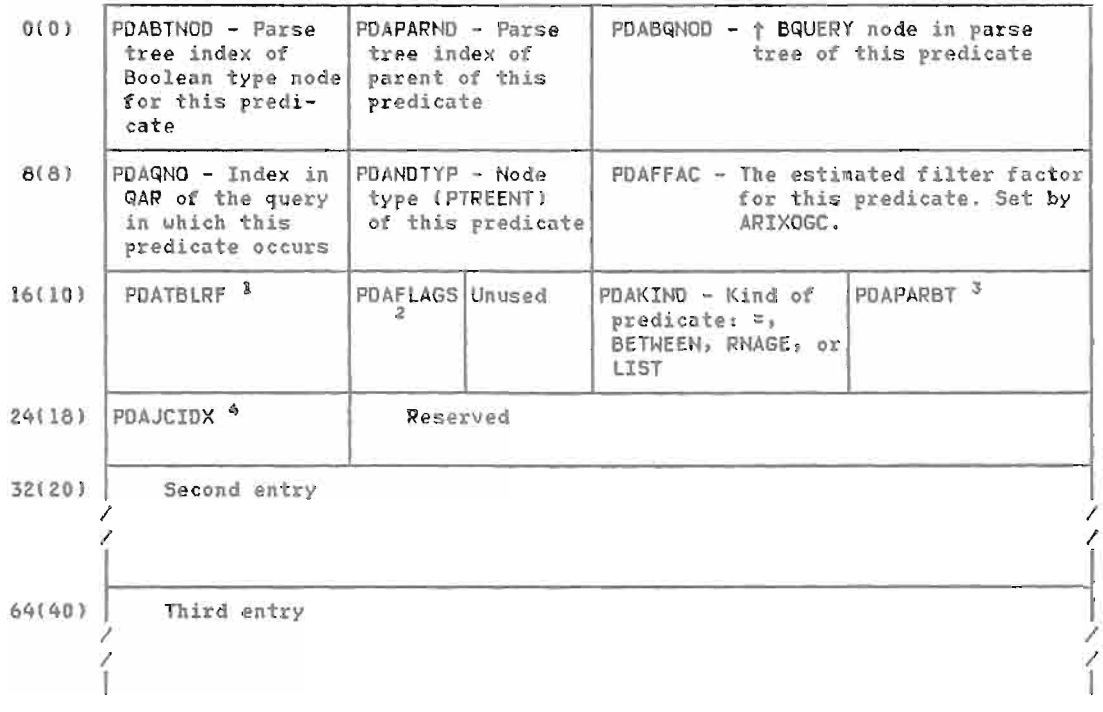
[8] 8-bit array. QAR(I).QARQRYRF(J) = '1'B if the subtree rooted at Query I references Query J with a correlation predicate. For all I<J QARQRYRF=0B.

QAR Flags

Offset	Field Name	Bits	Meanings
41(29)			
	QARCOPAR	1...	Bit set on if this query or its descendants contain a correlation to the immediate parent of this query. Set by ARIXOGP.
	QARDONOP	.1..	Set on by ARIXOGP if this query is done at open time of another query. Set by ARIXOGP.
	QARDUHAV	..1.	Set on by ARIXOEX if this query is directly under a HAVING tree.
		...x xxxx	Reserved.

PREDICATE ARRAY ('PDA')

The Predicate Array is pointed to by field OPTPREDP in OPTAREA. There is a maximum of 200 table entries. The current number of entries is shown in field OPTPREDX of OPTAREA. Each entry is 32 bytes long. Only the first entry is shown here. The second starts at offset 32(20), the third at 64(40), and so on.



1 16-bit string. Each bit position set on represents the TBA index of a table in which a column from this predicate is contained. A predicate must reference at least one table of the current BQUERY.

2 See PDA Flags following this figure.

3 Index in PDA of the root node of the Boolean type containing this predicate, or zero.

4 If this is a join predicate, index into JOINCOL of the left-hand side (LHS) join column.

PDA Flags

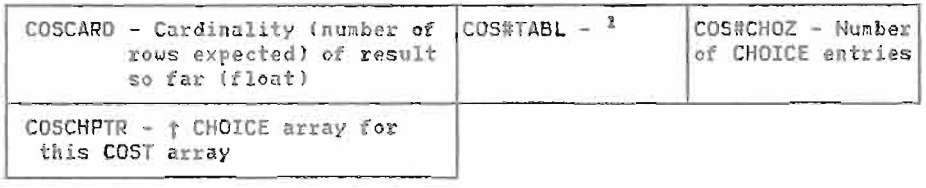
Offset	Field Name	Bits	Meanings
18(12)	PDAFLAGS		
	PDADELET	x...	Set to 0 by ARIXOFP. Set to 1 by ARIXORM if this predicate is removed from the WHERE tree.
	PDASARG	.1..	This predicate can be used as a search argument; that is, it is of the form col op value.
	PDAJOIN	..1.	This predicate is a join predicate.
	PDABT	...1	This predicate is a Boolean type; it is connected to the top of the WHERE tree only by ANDs.
	PDAKEEP	.... 1...	This predicate is to be kept in the WHERE tree regardless of being a SARG since it contains a SUBQUERY that returns a non-null type result
	PDAHPRED	.... .1..	This predicate is in a HAVING tree.
	PDAKDOM	.... ..1.	This predicate may be converted into a key specification. Set in ARIXOGC.
		x	Reserved.

COSTS

COSTS is an array of possibilitites for best-cost paths. Space for COSTS is gotten in ARIXOGP and is a variable-length array. There is an entry in COSTS for each table and each join combination of tables. So for an n-way join, there will will be 2**n-1 entries in the COSTS array. The index into the COSTS array is based on the binary representation of the tables joined together if each table is given a value of a bit position. That is, table 1 has value 1, table 2 has value 2, table 3 has value 4, table 4 has value 8, etc. The plan to join tables 3 and 2 would be found at index 6 of COSTS (value 4 + value 2). The plans described by COSTS(2**n-1) is the join of all tables requested. The format of one entry is shown below.

COSTS is pointed to by field OPTCOSPT of OPTAREA.

Dec(Hex) COSTS



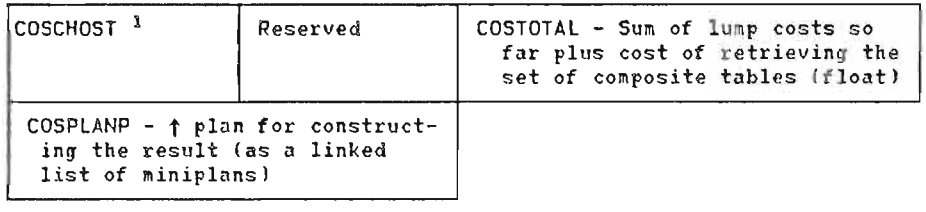
[1] Number of tables in this plan.

CHOICES

CHOICES is an array of entries relating to the COSTS data area. Single table entries are filled in by ARIXOTS. ARIXOCS fills in COSTS for each join possibility with information relating to each possible access paln stored in the CHOICES entries.

An entry in the CHOICES array is pointed to by the COSCHPTR field of the associated COSTS entry.

Dec(Hex) CHOICES



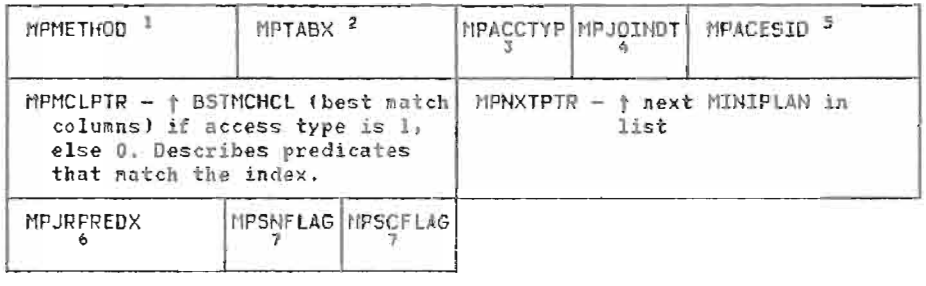
[1] Order resulting from join done with this choice (order class of result or -1 if unordered).

MINIPLAN

In calculating access path costs, Path Selection records pertinent information in a series of entries called MINIPLANs.

Initially, COSTS field COSPLANP points to MINIPLAN. When ARIXOGP calls ARIXOGA, it passes the pointer to the best MINIPLAN entry to generate ASL.

Dec(Hex) MINIPLAN



1 A value representing how the tables are joined together.
Value is:
4*(access method for this join Table, or zero if this plan is first table) + (access method of next table, or zero if this plan is last join Table)
Where: Access method = 1 for Type 1 join (nested loops),
2 for Type 2 join (sort merge),
3 for GROUPBY scan

2 Index in Table Array (TBA) of table to be added, or zero for extra plan(s) if needed for GROUPBY or ORDERBY after a join.

3 R or I: R=Relation (DBSPACE) scan, I=Index scan
4 A or D: Ascending/Descending. Filled in for Type 2 joins only.
5 Index in index array (IDX...) for access Type 1, else 0.
6 If Type 2 join, then index in Predicate Array (PDA) of join predicate; else zero.
7 See "MINIPLAN Flags" on next page.

MINIPLAN Flags

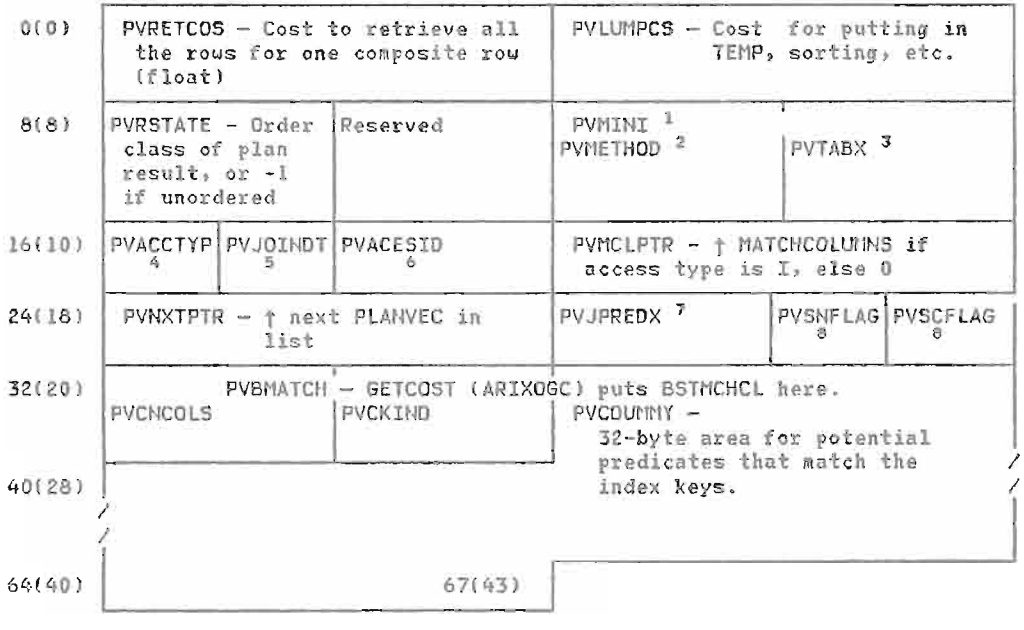
Offset	Field Name	Bits	Meanings
18(12)			
	MPSNFLAG		
	MPSNTABL	1...	Set if table being joined needs to be sorted before joining (sort new table). Indicates one of the following bits is set:
	MPSNJOIN	.1..	Sort key is join predicate.
	MPSNORDB	..1.	Sort key is ORDER BY list.
	MPSNGRPB	...1	Sort key is GROUP BY list.
	MPSNUNIQ	.... 1...	Sort should eliminate duplicates.
	MPSNFNUN	.... .1..	Sort key is SET FUNCTION UNIQUE column.
		xx	Reserved.
19(13)			
	MPSCFLAG		
	MPSORTCM	1...	Set if result so far needs to be sorted (sort composite). Indicates one of the following bits is set.
	MPSCJOIN	.1..	Sort key is join predicate.
	MPSCORDB	..1.	Sort key is ORDER BY list.
	MPSCGRPB	...1	Sort key is GROUP BY list.
	MPSCUNIQ	.... 1...	Sort should eliminate duplicates.
	MPCNFNUN	.... .1..	Sort key is SET FUNCTION UNIQUE column.
		xx	Reserved.

PLANVEC

PLANVEC is a vector of multiple miniplans. It is passed back from ARIXOGC to ARIXOTS or ARIXOCS. These two modules fill in COSTS and CHOICES and point to the associated MINIPLAN. Plans are returned for the cheapest cost plan and for each interesting ordering that matches a join column, and ORDER BY, or a GROUP BY.

The PLANVEC is pointed to by field OPTPLANV in OPTAREA.

Dec(Hex) PLANVEC
PVPLAN (An array of plans, the dimension of which is determined and and gotten by ARIXOGP)



[1] Bytes 12(0C) through 31(1F) are a miniplan for adding a new table. This area has the same format as the MINIPLAN data area.

[2–8]: See next page.

[2] A value representing how the tables are joined together.
Value is:
4*(access method for this join Table, or zero if this plan is first table) + (access method of next table, or zero if this plan is last join Table)
Where: Access method = 1 for Type 1 join (nested loops),
2 for Type 2 join (sort merge),
3 for GROUPBY scan

[3] Index in Table Array (TBA) of table to be added, or zero for extra plan(s) if needed for GROUPBY or ORDERBY after a join.

[4] R or I: R=Relation (DBSPACE) scan, I=Index scan

[5] A or D: Ascending/Descending. Filled in for Type 2 joins only.

[6] Index in Index Array (IDX...) access Type 1, else 0.

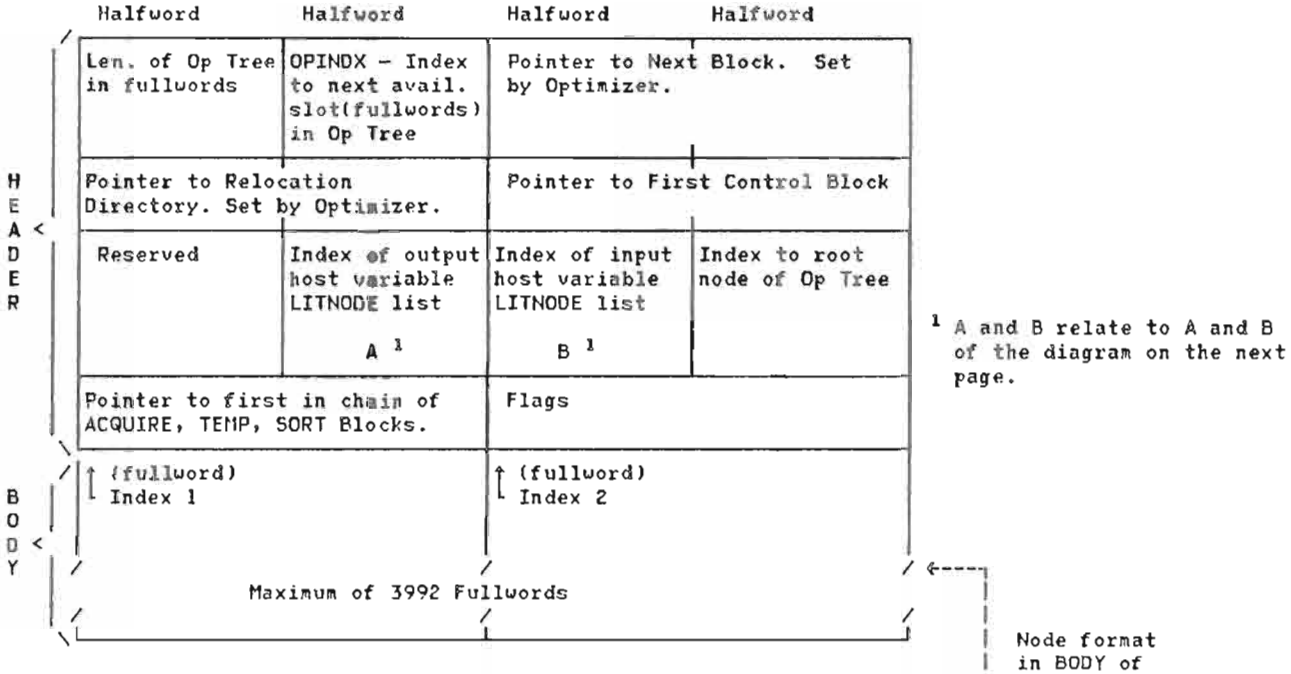
[7] If Type 2 join, then index in Predicate Array (PDA) of join predicate; else zero.

[8] See "PLANVEC Flags" on next page.

PLANVEC Flags

Offset	Field Name	Bits	Meanings
18(12)			
	PVSNFLAG		
	PVSNTABL	1...	Set if table being joined needs to be sorted before joining (sort new table). Indicates one of the following bits is set:
	PVSNJOIN	.1..	Sort key is join predicate.
	PVSNORDB	..1.	Sort key is ORDER BY list.
	PVSNGRPB	...1	Sort key is GROUP BY list.
	PVSNUNIQ	.... 1...	Sort should eliminate duplicates.
	PVSNFNUN	.... .1..	Sort key is SET FUNCTION UNIQUE column.
		xx	Reserved.
19(13)			
	PVSCFLAG		
	PVSORTCM	1...	Set if result so far needs to be sorted (sort composite). Indicates one of the following bits is set.
	PVSCJOIN	.1..	Sort key is join predicate.
	PVSCORDB	..1.	Sort key is ORDER BY list.
	PVSCGRPB	...1	Sort key is GROUP BY list.
	PVSCUNIQ	.... 1...	Sort should eliminate duplicates.
	PVCNFNUN	.... .1..	Sort key is SET FUNCTION UNIQUE column.
		xx	Reserved.

OP TREE FORMAT



One node of Op Tree = three fullwords (six halfwords) as follows:

Node Type (PTREENT)	P1 (PTREEP1)	P2 (PTREEP2)	P3 (PTREEP3)
P4 (PTREEP4)	P5 (PTREEP5)		

Use of P1-P5 varies according to node type. P1-P5 usually contain codes or indexes to other nodes in the Op Tree. They may also point to descriptors in the Op Tree. Descriptors are usually strings of characters, etc, that do not fit in the halfword format of the node (described above). The Op Tree nodes and descriptors are shown further in this section under "Op Tree Node Encodings" and "Descriptor Records".

- LISTNOD ARITHMETIC LIST node
- COLMNNOD COLUMN node
- LITNODE LITERAL node
- MINUSNOD MINUS node
- SETFNNOD SET FUNCTION node
- STARNODE STAR (ASTERISK) node

An expression-type node may be any of these:

- ARITHNOD ARITHMETIC node
- COLMNNOD COLUMN node
- LITNODE LITERAL node
- MINUSNOD MINUS node
- SETFNNOD SET FUNCTION node

Use of P1 - P5 in Various Node Types

ACQUIRNOD (ACQUIRE DBSPACE node)
P1 DBSPACE type: integer from 1 to 5.
P2 Index to ENTDESCR (ENTITY DESCRIPTOR) for DBSPACE name.
P4 Requested DBSPACE size in blocks (each block = 128 pages. Parser fills in a default of 1 block).
P5 Index of SEGSPNOD (DBSPACE SPECIFICATION node).

ALISTNOD (ARITHMETIC LIST node) (Used whenever an ARITHNODE would have appeared on the SELECT-list, including when it appears as the final element on the list)
P3 Index of next node in SELECT list, or zero.
P5 Index of ARITHNOD

AQRTMNOD (ACQUIRE TEMPORARY (DBSPACE) node)
(Created by Optimizer in ASL tree only. For acquiring a temporary Type-5 DBSPACE.)
P1 Index of BINDTABLE, or zero.
P3 Index of next node on query chain.
P5 Index of LOCDESC (LOCATION DESCRIPTOR) for ACQBLK (ACQUIRE BLOCK).

ARITHNOD
P1 Zero after parsing. Filled by Optimizer with decimal data-type if result of expression is decimal.
P2 Zero after parsing. Filled in by Optimizer with spec-length (precision and scale) if result of expression is decimal.
P3 Index of left-hand expression.
P4 ARITHOP code: PLUSCODE, MINSCODE (MINUS CODE), MULTCODE, DIVDCODE (DIVIDE code).
P5 Index of right-hand expression.

AUTHNODE
P1 Operation code: One of SELCTAUT, INSRTAUT, DELETAUT, UPDTEAUT, EXPNDAUT (ALTER authority), INDEXAUT, ALLAUTH, RUNAUTH, DBAAUTH, RESRCAUT, CONNAUT, SCHEDAUT.
P3 Index to COLMNNOD if operation is UPDATE and a column-list is specified; else zero.
P5 Index to next AUTHNODE or zero.

BLDIXNOD (BUILD INDEX node) (Created by Optimizer in ASL tree only.)
P1 Index of BINDTAB, or zero.
P3 Index of next node on query chain.
P5 Index of LOCATION DESCRIPTOR for CBASE for creating index.

BLDVLNOD (BUILD VALUE node) (Created by Optimizer in ASL tree only.)
- P1 Index of BINDTAB, or zero.
- P3 Index of next node on query chain.
- P5 Index of SQUERNOD (SQUERY node), which returns a single value. The place to hold the value is denoted by a LITUPNOD on P1 of the SQUERNOD.

BOLENNOD (BOOLEAN node)
- P1 Reserved for internal use by Optimizer.
- P2 Reserved for internal use by Optimizer.
- P3 Index of left-hand node -- BOOLEAN node or any predicate.
- P4 ANDCODE or ORCODE.
- P5 Index of right-hand node -- BOOLEAN node or any predicate.

BQURYNOD (BQUERY node)
- P1 Index of BQUERY DESCRIPTOR allocated by Parser.
- P2 Zero after parsing. After optimization, contains index of LITUPNOD for output of query-block, if any. (A BQUERY node created by the Optimizer inside a join may have P2 = 0.)
- P3 Index of first node of SELECT-list (see page 137 for permissible node types). After parsing, P3=0 denotes SELECT *, which is expanded by the Optimizer into a list of column-nodes. After optimization, P3=0 denotes no SELECT-list, as in the case of a BQUERY node created by the Optimizer inside a join.
- P4 Index of first TABLENOD of FROM list.
- P5 Index of GROUP BY node, else index of root of WHERE tree, else zero. After optimization, residual WHERE-tree is moved to TABLENOD.

BTWENNOD (BETWEEN node)
- P1 Reserved for internal use by Optimizer.
- P2 Reserved for internal use by Optimizer.
- P3 Index of lower bound expression.
- P4 Index of expression that is alleged to be ≥ lower bound and ≤ upper bound.
- P5 Index of upper bound expression.

CHSEGNOD (CHANGE DBSPACE node)
- P1 Index of ENTITY DESCRIPTOR for DBSPACE name.
- P2 New IDXLOCMO (INDEX LOCK mode): SEGTCODE (DBSPACE) or PAGECODE or TUPLCODE (ROW code) or zero if no change.
- P4 Positive number or zero denotes new PCTFREE for this DBSPACE. Negative number denotes no change.

COLMNNOD (COLUMN node)
- P1 Zero, then column number within TABLE or VIEW. (Filled in by Optimizer.)
- P2 Index of COLUMN DESCRIPTOR.
- P3 Index of next node in column-spec or select-list, else zero.
- P4 In case of a CREATE or ALTER TABLE command, Parser fills in type: CHARTYPE, INTTYPE, HWTYPE, DECTYPE, FLOATYPE, VRCHRTYPE (VARYING CHAR type), LVARCTYP (long char), GRAPH, VGRAPH, LVGRAPH or any of these prefixed by 'N'. For other commands, Parser leaves P4=0; Optimizer later fills in type code. Optimizer always chooses an RS_TYPE or NRS_TYPE if node occurs in select-list or in a predicate.
- P5 Length specification, not including any NULL bytes. For fixed-length CHAR columns, the length; for varying-length CHAR or GRAPH columns, max length; for DECIMAL, the m,n spec; else the actual length. Filled in by Parser for CREATE and ALTER TABLE commands, otherwise by Optimizer.

COMNTNOD (COMMENT node)
- P1 Typecode -- TABLECOD, VIEWCODE, or CLMNCODE (COLUMN code).
- P2 Index to ENTITY DESCRIPTOR for object to be commented on. If object is a column, this ENTITY DESCRIPTOR contains the table name.
- P3 If P1 is CLMNCODE, index to ENTITY DESCRIPTOR for column name; else zero.
- P4 Length of comment in bytes.
- P5 Index to OPCHAR representation of comment.

CRTIMNOD (CREATE INDEX node)
- P2 Index to ENTITY DESCRIPTOR for index name.
- P3 Index to COLMNNOD in ORDER specification. Ascending/descending information is contained in COLUMN DESCRIPTOR for each column.
- P4 Index to ENTITY DESCRIPTOR for table name.
- P5 Index to IMGOPNOD (INDEX OP node).

CRTMPNOD (CREATE TEMPORARY node) (Created by Optimizer in ASL tree only. For creating a temporary list or TABLE.)
- P1 Index of BINDTABL, or zero.
- P3 Index of next node on query chain.
- P5 Index of LOCATION DESCRIPTOR for TEMPBLK.

CRTTBNOD (CREATE TABLE node)
- P1 Index of USER DESCRIPTOR for authorization checking.
- P2 Index of ENTITY DESCRIPTOR for table name.
- P3 Index to ENTITY DESCRIPTOR for DBSPACE name, or zero.
- P5 Index to COLMNNOD of column definition list.

CURNTNOD (CURRENT node)
- P4 After optimization, indicates block number of CURSOR DESCRIPTOR.
- P5 Index of CURSOR DESCRIPTOR. (After parsing, index within Block 0; after optimization, index within block indicated by P4.)

DEFVUNOD (CREATE VIEW node)
- P2 Zero after parsing. Executive fills in index of SQL statement creating the view, in PL/I VARCHAR format.
- P3 Index of TABLENOD indicating names for view and its columns.
- P4 Zero after parsing. Optimizer sets to index of first TABLENOD in a chain of the views and tables that are depended on by this view.
- P5 Index of QUERYNOD creating the view.

DELETNOD (DELETE node)
- P4 Index of TABLENOD.
- P5 Index of root of WHERE tree (see above for types), or zero.

DFSYNNOD (CREATE SYNONYM node)
- P1 Index of ENTITY DESCRIPTOR for synonym.
- P2 Index of ENTITY DESCRIPTOR for table on which synonym is created.

DQUERNOD (DQUERY node) (Created by Optimizer in ASL tree only.)
- P1 Index of BINDTABL, or zero.
- P3 Index of FORCHNOD (FOR-EACH node), or zero.
- P5 Index of SQUERNOD (SQUERY node) or BQURYNOD (BQUERY node) or SETOPNOD to be executed at this point in the query chain.

DROPNODE (DROP node)
- P1 Type code: TABLECOD, VIEWCODE, INDXCODE, LINKCODE, SEGTCODE DBSPACE code), PROGCODE (PROGRAM code).
- P2 Index to ENTITY DESCRIPTOR for name of dropped object. 0 if object specified as host variable.

ENDNODE (COMMIT WORK node)

EXISTNOD (EXISTS node)
- P5 Index of BQURYNOD for subquery.

EXPNDNOD (ALTER node)
- P1 Index to USER DESCRIPTOR for authorization checking.
- P2 Index to ENTITY DESCRIPTOR.
- P3 Index to COLUMN node

FORCHNOD (FOR-EACH node) (Created by Optimizer in ASL tree only.)
- P1 Index of BINDTABL, or zero.
- P4 Zero if join method 1 is used. If join method 2 is used, index of JOIN DESCRIPTOR.
- P5 Index of BQURYNOD that will produce rows to be joined to the next higher level on the query chain.

GRANTNOD (GRANT node)
- P1 Zero after parsing. After optimization, index of USER DESCRIPTOR containing userid of grantor.
- P2 Index to ENTITY DESCRIPTOR containing table name. Zero if a special privilege is being granted.
- P3 Index of first USERNODE in list of grantees.
- P4 GRANTAUT or index of first USERNODE in list of passwords, or zero.
- P5 Index of AUTHNODE.

GROUPNOD (GROUP-BY node)
- P1 Zero after parsing. After optimization, may contain index of BINDTABLE or zero. The entries in this BINDTABL have a special meaning: left side is a COLMNNOD for one of the GROUP BY columns, and right side points to a LITNODE for the value of the current group. The binding is conditional. If value is unchanged, continue to accumulate set-fns for the parent BQURYNOD. If value is changed, compute BQURYNOD output for one group and start a new group. If P1=0, entire table is treated as a group.
- P3 Index of first COLMNNOD of GROUP BY list. Zero if P1=0 (denotes HAVING clause with no GROUP BY).
- P4 Index of root of HAVING tree, else zero.
- P5 After parsing, index of WHERE tree. Ignored after optimization (residual predicates moved to TABLENOD).

IMGOPNOD (INDEX OP node)
- P2 Percentage free space, a halfword integer, filled in by Parser in case of default.
- P3 Unique image. UNIQCODE or DUPLCODE.

INSRTNOD (INSERT node)
- P1 Index of BINDTABL, or zero.
- P3 Index of next node on query chain, if any.
- P4 Index of TABLENOD describing target of insertion.
- P5 Index of LITUPNOD or QUERYNOD or SQUERNOD (SQUERY node) describing data to be inserted.

ISNULNOD (IS-NULL node)
- P4 YESCODE if predicate tests for null; NOCODE if predicate tests for non-null.
- P5 Index of COLMNNOD whose value is to be tested.

LITNODE (Note: In the case of a LITNODE representing a program variable, all P's except P3 are zero after parsing.)
- P1 Data type code: After parsing, always CHARTYPE, NULLTYPE, or USERTYPE. CHARTYPE and NULLTYPE later changed by Optimizer to one of the 26 possible data type codes (see page 151)
- P2 After parsing, if P1=CHARTYPE, P2 is index of character string value, else P2=0. After optimization, P2 is index of LOCATION DESCRIPTOR which, in turn, points to value. The value is an appropriate type and is found in Block 1 or an overflow block. If this LITNODE represents an input variable, LOCATION DESCRIPTOR (P2) points to an empty space of the correct size.
- P3 In a LIT ROW, index of next LITNODE in list, else zero. On the select list, index of next node on select list, else zero.
- P4 After parsing, a type hint (ALLDIGTS (ALL DIGITS), DGTANDPT (DIGITS and DECIMAL POINT), ALPHANUM (for CHAR and hex literals), HEXBINRY, GRAPH (DBCS), FLOATYPE). After optimization, P4 is zero except in the case of an input variable of (N)RSVARCHAR type. In this case, P4 serves as an index to a LIT DESCRIPTOR in Block 0.
- P5 Length specification. Max length for VARCHAR types, (M,N, for DGTANDPT (decimal), else actual length. Does not include null byte or length bytes which may be part of data format.

LITUPNOD
- P5 Index of first LITNODE in literal ROW.

LOCKNODE (LOCK node)
- P1 RELNCODE (RELATION code) or SEGTCODE (DBSPACE code)
- P2 Index of ENTITY DESCRIPTOR.
- P4 Mode: SHARMODE or EXCLMODE.

MINUSNOD (Unary arithmetic MINUS node)
- P1 Zero after parsing. Set by Optimizer to data type decimal if result of expression is decimal.
- P2 Zero after parsing. Set by Optimizer to spec-length (precision and length) if result of expression is decimal.
- P3 Index of next element on select list, else zero.
- P5 Index of any expression-type node.

NOTNODE (NOT node)
- P5 Index of BOOLNOD (BOOLEAN node) or any predicate.

ORDBYNOD (ORDER BY node)
- P1 Order-determining column index value.
- P2 ASCCODE or DESCCODE.
- P3 Index to next ORDBYNOD, else zero.
- P4 Index to the COLMNNOD of the ORDER BY specification.

QUERYNOD (Appears before optimization, but not used in ASL.)
- P1 Index of QUERY DESCRIPTOR, allocated by Parser.
- P2 Zero after parsing. If the QUERYNOD occurs inside a view definition, P2 is set by the Optimizer to RNTMCODE (RUN TIME code) or PRTMCODE (PREP TIME code) to denote whether locks on catalog rows must be released. Otherwise, P2=0.
- P3 Index of first ORDBYNOD of ORDER BY list, else zero.
- P4 Index of first COLUMNOD of UPDATE list, else zero.
- P5 Index of SETOPNOD or BQUERYNOD.

RESTONOD (ROLLBACK WORK node) REVOKNOD (REVOKE node)
- P1 Zero after parsing. After optimization, index of USER DESCRIPTOR containing ID of revoker.
- P2 Index of ENTITY DESCRIPTOR for table being revoked. Zero if a special privilege is being revoked.
- P3 Index of first USERNODE in list of revokees.
- P5 Index of AUTHNODE.

SEGSPNOD (DBSPACE SPECIFICATION node)
- P1 NHEADER (filled in by Parser in case of default).
- P2 LOCKMODE: SEGTCODE (DBSPACE code) or PAGECODE or TUPLCODE (ROW code), or zero if unspecified.
- P3 PCTINDEX (filled in by Parser in case of default).
- P4 PCTFREE (filled in by Parser in case of default).

SETFNNOD (SET FUNCTION node)
- P1 UNIQCODE (DISTINCT code) or DUPLCODE (ALL code).
- P2 Index of SET FUNCTION DESCRIPTOR.
- P3 Index of next node in SELECT list, else zero.
- P4 Set fn code: AVGFN, SUMFN, COUNTFN, MAXFN, MINFN, USERFN.
- P5 Index of expression-type node or zero in case of COUNT(*).

SETNODE (SET node)
- P3 Index of next SETNODE, else zero.
- P4 Index of COLMNNOD that is target of update.
- P5 Index of QUERYNOD or any expression-type node except SETFNNOD.

SETOPNOD (SET OP node)
- P3 Index of BQURYNOD (BQUERY node).
- P4 UNONCODE (UNION code), INSCCODE (INTERSECT code), or SMINCODE (SET MINUS code).
- P5 Index of BQURYNOD or SETOPNOD.

SORTNODE (SORT node) (Created by Optimizer in ASL tree only)
- P1 Index of BINDTABL or zero.
- P3 Index of next node on query chain.

P5 Index of LOCATION DESCRIPTOR for SORTBLK.

SQRY1NOD (SQUERY1 node) (Created by Optimizer in ASL tree only)
Identical to SQUERNOD. Used when a subquery occurs inside a predicate; i.e., under a TBLCMNOD (TABLE COMPARE node).)

SQUERNOD (Created by Optimizer in ASL tree only)
P1 Index of BINDTABLE, or zero.
P2 Index of LITUPNOD for output of superquery.
P3 Next node in unconditional query chain, if any (may be zero if only a P4-chain exists). Structure of P3-chain is:
1. An optional collection of ACQUIRE TEMPORARY nodes (AQRTMNOD), CREATE TEMPORARY nodes (CRTMPNOD), BUILD INDEX nodes (BLDIXNOD), BUILD VALUE nodes (BLDVLNOD), INSERT nodes (INSRTNOD), and SORT nodes (SORTNODE).
2. One DQUERNOD, optionally followed by one FORCHNOD (FOR-EACH node).

P4 Next node in conditional query chain, if any (may be zero if only a P3-chain exists). P4-chain is executed before P3-chain, but only in the case that some conditional entry in the bind-table changed. Structure of P4-chain is same as structure of P3-chain except that DQUERNOD is optional.
P5 Index of SQUERY DESCRIPTOR. SQUERY DESCRIPTOR points to OCL (OPEN CHECK LIST), which may cause this SQUERY to be aborted (return EOF) if a null value is returned by a subquery.

STARNODE (ASTERISK node) (Used for X.*, table.*)
P3 Index of next node on SELECT-list, or zero.
P4 Index of TABLENOD for which this node represents "all columns".

TABLENOD (TABLE node)
P1 Type code -- TABLCODE or VIEWCODE. Set to zero by Parser.
P2 Index of RELATION DESCRIPTOR (TABLE DESCRIPTOR).
P3 If under a DEFVUNOD, the index of the COLUMN node in the view; if under an insertion, index of first COLUMN node of COLUMN-list; else zero.
P4 Index of next TABLE node in FROM list, else zero.
P5 Zero after parsing. The Optimizer uses this place to attach the residual WHERE-tree for this table.

TBLCMNOD (TABLE COMPARE node)
P1 Zero after parsing. If this predicate is non-sargable, the Optimizer sets P1 to index of LITNODE, which serves as a slot to hold the left-hand side (LHS) value for comparison to subquery on right-hand side (RHS). Otherwise, zero.
P2 Zero after parsing. Optimizer sets to 0 or 1 to indicate how LHS value is to be compared to value returned by RHS subquery. 0 means direct value-to-value comparison. 1 means RHS is null-permitting but LHS is not; hence, RHS must be examined for null ('FFFFFFFF' in 4 bytes preceding value-slot) before value-comparison is done.
P3 Index of expression.
P4 Comparison code: one of codes listed in ARIBNUM.
P5 Index of BQURYNOD, SETOPNOD, or SQRY1NOD.

UPDTENOD (UPDATE node)
P3 Index of SETNODE.
P4 Index of TABLE node.
P5 Index of root of WHERE tree (see above for types), or zero.

UPSTANOD (UPDATE STATISTICS node)
P1 Index of ENTITY DESCRIPTOR for table-name.
P2 Index of ENTITY DESCRIPTOR for DBSPACE or zero.
P3 ALLCODE or zero.

USERNODE
P2 Index of a STRING DESCRIPTOR containing a userid, password, or PUBLIC.
P5 Index of next USERNODE in the list, else zero.

VALCMNOD (VALUE COMPARE node)
P1 Reserved for internal use by Optimizer.
P2 Reserved for internal use by Optimizer.
P3 Index of left-hand expression.
P4 Code for comparison operator: one of six simple comparison codes listed in ARIBNUM, or LIKECODE.
P5 Index of right-hand expression. If P4=LIKECODE, P5 is index of pattern LITNODE.

All codes referred to here are declared in ARIBNUM (see page 150). Based masks for data value representation and for DESCRIP structures are declared in ARIBRDS.

DESCRIPTOR RECORDS

Descriptor records are placed in the Op Tree and are pointed to by Nodes in the Op Tree.

BQUERY DESCRIPTOR

BQRYDESC

Hex			
0	BQDBLKPT - ↑ RSI structures for duplicate elimination or null (set to null by Parser).	BQDCOND*	////// ////// ////// //////	BQDJNMET - See Note 1.

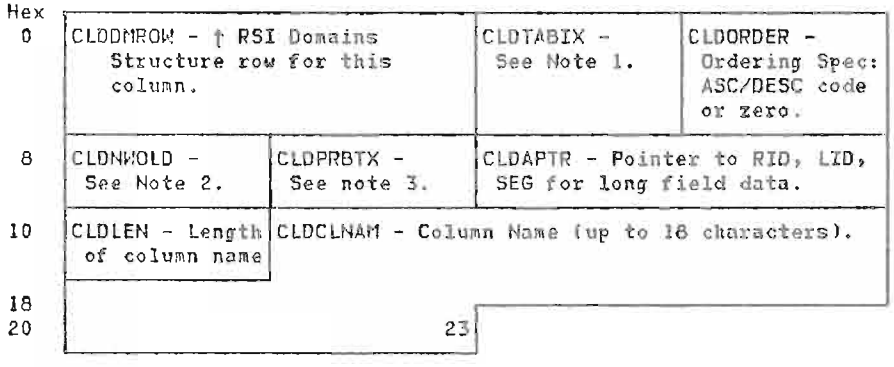
* Flags and masks

Flag Field	Hex Value	Means
BQDCOND	X'80'	Set by Parser or later by Optimizer if SETOPS required.
	X'40'	Query involves ORDER BY.
	X'20'	Query involves GROUP BY.
	X'10'	Query involves JOIN.
	X'08'	Output involves SET FNs.
	X'04'	Output involves SETFN(UNIQUE...).

Note 1: Copied from Miniplan method. Equals: 4 * (access method for this join table, or zero if this plan is first table) + (access method for next table, or zero if this plan is last join table).

COLUMN DESCRIPTOR

COLDESCR

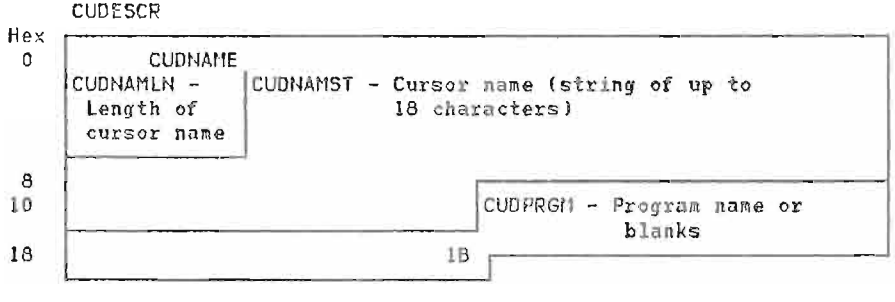


Notes:

1. After parsing, this contains an index to the pertinent TABLENODE or zero. If zero, then Optimizer puts in index to TABLENODE after XCOLLK. This becomes the pertinent TABARRAY.
2. OLDCODE, NEWCODE, or zero. Used with assertions and triggers.
3. Index in PREP array. Used by Optimizer.

CURSOR DESCRIPTOR

CUDESCR



ENTITY DESCRIPTOR

ENTDESCR

Hex		
0	ENTCR8TR - Creator's Userid or blanks	
8	ENTLEN - Length of Entity Name	ENTNAME - Entity Name (up to 18 characters). Size of ENTNAME must be determined by programmer.
10		
18	1B	

FIELD NAME DESCRIPTOR

FNDESCRP

Hex		
0	FNDBUFER - Pointer to SFNDATA allocated by Optimizer.	
8	FNDUSRIX - Index of ENTDESCR for USERFN name	FNDTYPE - Data type of current value (PL/I types only).
10	FNDLENTH - Specified length of current value.	FNDPRVAL - Index of LITNODE for holding previous value for doing UNIQUE, else 0.

JOIN (Method 2 only) DESCRIPTOR

JOINDESC

Hex			
0	JDJOINVL - Index of LITNODE for current join value. Data type matches JDOUTCOL.	JDOUTCOL - Index of COLUMNNODE for join-column in outer (DQ) scan.	
8	JDINCOL - Index of COLUMNNODE for join-column in inner (FE) scan.	JDDIRECT* (See Note 1.)	////////////////////// ////////////////////// //////////////////////
10	FNDLENTH - Specified length of current value.	FNDPRVAL - Index of LITNODE for holding previous value for doing UNIQUE, else 0.	

Note 1. This one-character field inidcates the direction of sort-merge on join-column: 'A' for ascending, or 'D' for descending.

LITNODE DESCRIPTOR

LITDESCR: Indexed by P4 of a LITNODE which represents an RSVARCHAR or NRSVARCHAR corresponding to an input variable. LITDESCR is generated by th Optimizer only if needed to provide extra places to put the actual length (including null byte) of these variables.

Hex		
0	LIT#LNTH - Number of places into which to put the actual length.	////////////////////////////////// ////////////////////////////////// //////////////////////////////////
8	LITLNPTR - Pointers to length spec location of FIXED BIN(15) type. The number of these pointers is in LIT#LNTH.	

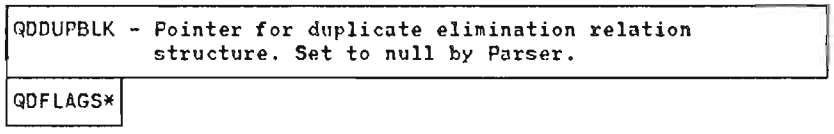
LOCATION DESCRIPTOR

LOCDESC

Hex		
0	4-byte pointer to other blocks	(See AQRTNNOD (P5), BLOIXNOD (P5), CRTMPNOD (P5), LITNODE (P2), and SORTNODE (P5) for block types and further description.)

QUERY DESCRIPTOR

QRYDESCR



* Flags and masks

Flag Field	Hex Value	Means
QDSARGCM	80	If "ON" (1), the statement involves a projection from a single relation (only COLUMNNODEs in select-list) and SARGS generated by the Optimizer contain ORs. Set by the Optimizer.
QDCURQRY	40	Set by PREP Executive before calling Optimizer. If ON, then statement was 'LET CURSOR BE SELECT'; else it was 'SELECT'.

RELATION (TABLE) DESCRIPTOR

RELDESCR

Hex				
0	RDACCTYP See Note 1.	RDSTATUS *	////////////// ////////////// //////////////	RDPLAHLD - Pointer to control block for RSI scan.
8	RDLOCBLK - † lock structure (LBASE) or null. Share lock or relation for SETFN or relation scan.			RDCNTRLP - † RSI control block.
10	RDPROGNM - Program name if specified with CURSORNAME, else blanks.			
18	RDCURSNM - Cursor name for link so TID can be moved into SCANCONBLK.			
20	RDCREATR - Creator's userid if specified, else blanks.			
28	RDLEN - Length of relation (table) name	RDRELNAME - Relation (table) name (up to 18 characters).		
30				
38			3B	

* Flags and masks

Flag Field	Hex Value	Means
RDCORELT	80	Indicates column information required for correlation from lower level.

Note 1: The character in this position indicates:

V = View R = Relation scan
L = Link F = Fetch
I = Image scan C = Cursor

TRACE OUTPUT (Dec)	TRACE DUMP (Hex)	MEANING
0160	00A0	All digits
0161	00A1	Digits and point
0162	00A2	Alphanumeric
0170	00AA	Shared mode
0171	00AB	Exclusive mode
0180	00B4	<
0181	00B5	≤
0182	00B6	>
0183	00B7	≥
0184	00B8	=
0185	00B9	≠
0186	00BA	Like
0190	00BE	< any
0191	00BF	≤ any
0192	00C0	> any
0193	00C1	≥ any
0194	00C2	= any
0195	00C3	≠ any
0200	00C8	< all
0201	00C9	≤ all
0202	00CA	> all
0203	00CB	≥ all
0204	00CC	= all
0205	00CD	≠ all
0220	00DC	Contains
0221	00DD	Does not contain
0222	00DE	Is in
0223	00DF	Is not in

Codes for Op Tree Nodes

0230	00E6	ASSIGN node
0231	00E7	QUERY node
0232	00E8	INSERT node
0233	00E9	DELETE node
0234	00EA	UPDATE node
0235	00EB	SET node
0236	00EC	SET OP node
0237	00ED	COLUMN node
0238	00EE	TABLE node
0239	00EF	BQUERY node
0240	00F0	GROUP BY node
0241	00F1	BOOLEAN node
0242	00F2	NOT node
0243	00F3	MINUS node
0244	00F4	VALUE COMPARE node
0245	00F5	BETWEEN node
0246	00F6	TABLE COMPARE node
0247	00F7	TPLMHNOD ┐ Used for
0248	00F8	IFNODE > View
0249	00F9	SETGNNOD │ Composition
0250	00FA	QCOMPNOD ┘
0251	00FB	ALISTNOD (Arithmetic List Node)
0252	00FC	SETFN node
0253	00FD	ARITHMETIC node
0254	00FE	LITNODE
0255	00FF	LITUPNOD
0256	0100	CURSOR node
0257	0101	CREATE TABLE node
0260	0104	EXPAND node
0262	0106	CREATE VIEW node
0263	0107	INDEX OP node
0264	0108	DROP node
0265	0109	LNKOPNOD \ Used by
0266	010A	CRLNKNOD / Trace Formatter
0268	010C	GRANT node
0269	010D	USER node
0270	010E	Authorization node
0271	010F	REVOKE node
0272	0110	COMMENT node
0273	0111	JOIN node
0274	0112	BEGIN node
0275	0113	COMMIT WORK node
0276	0114	SAVENODE (Used by Trace Formatter)
0277	0115	ROLLBACK WORK node
0278	0116	LOCK node
0280	0118	CURRENT node
0281	0119	ASTERISK (STAR) node
0282	011A	ACQUIRE node
0283	011B	DBSPACE node
0285	011D	CHANGE DBSPACE node
0286	011E	CREATE SYNONYM node
0287	011F	TEMPORARY node
0288	0120	IS NULL node
0289	0121	CHECKPOINT node
0295	0127	ORDER BY node
0296	0128	UPDATE STATISTICS node
0299	012B	STFNINOD (set by Code Generator)
0300	012C	SQUERY node
0301	012D	DQUERY node
0302	012E	FOR EACH node
0303	012F	ACQUIRE TEMPORARY DBSPACE node
0304	0130	CREATE TEMPORARY node
0305	0131	SORT node
0306	0132	BUILD VALUE node
0307	0133	BUILD INDEX node

TRACE OUTPUT (Dec)	TRACE DUMP (Hex)	MEANING
0308	0134	CREATE VIEW node
0309	0135	SQUERY node
0310	0136	COMMIT WORK node
0311	0137	SCHEDULE node
0312	0138	EXIST node
0702	02BE	Run time
0703	02BF	Print time
0704	02C0	Program
0705	02C1	Row
0706	02C2	Page
0708	02C4	Release
0709	02C5	Retain
1186	04A2	Line
1222	04C6	Is in

CODES FOR DATA TYPES

Variable Character Type

0448	01C0	No null
0449	01C1	Null permitted
0450	01C2	DBSS format no null
0451	01C3	DBSS format null permitted

Fixed Character Type

0452	01C4	No null
0453	01C5	Null permitted
0454	01C6	DBSS format no null
0455	01C7	DBSS format null permitted

Long Varying Character Type

0456	01C8	No null
0457	01C9	Null permitted

Floating Point Type

0480	01E0	No null
0481	01E1	Null permitted
0482	01E2	DBSS format no null
0483	01E3	DBSS format null permitted

Decimal Type

0484	01E4	No null
0485	01E5	Null permitted
0486	01E6	DBSS format no null
0487	01E7	DBSS format null permitted

Integer Type

0496	01F0	No null
0497	01F1	Null permitted
0498	01F2	DBSS format no null
0499	01F3	DBSS format null permitted

Halfword Type

0500	01F4	No null
0501	01F5	Null permitted
0502	01F6	DBSS format no null
0503	01F7	DBSS format null permitted

Graphic Type

0468	01D4	No null
0469	01D5	Null permitted
0470	01D6	DBSS format no null
0471	01D7	DBSS format null permitted

TRACE OUTPUT (Dec)	TRACE DUMP (Hex)	MEANING
Varying Graphic Type		
0464	01D0	No null
0465	01D1	Null permitted
0466	01D2	DBSS format no null
0467	01D3	DBSS format null permitted
Long Varying Graphic Type		
0472	01D8	No null
0473	01D9	Null permitted

PGCTRS

PGCTRS contains the number of "cushion" pages, the number of new pages that have been allocated since the last checkpoint, the number of free pages for each storage pool, and a warning flag to keep track of "SOSWARNINGS" for each storage pool. Addressability to PGCTRS is set during DBSS initialization. The anchor pointer, PGCTRSP, is in YRSSCVT. The structure is initialized in module ARISIIO, and MAXPOOLS is the dimension.

Dec(Hex) PGCTRS

0 (0)	CUSHION - Cushion for storage pool	NEWPAGES - New pages since checkpoint	
8 (8)	FREEPGS - Number of free pages in storage pool	WARNED [1]	Reserved

[1] WARNED:
'N' = An SOS warning message has not been sent.
'Y' = An SOS warning message has been sent, but a checkpoint has not been taken yet.
'C' = An SOS warning message was sent, and a checkpoint was taken but did not help reclaim enough space.

PPOPGNST

The PPOPGNST is pointed to by field PPPOPGNP of the PREP Control Block (PREPDSCB). The PPOPGNST structure is passed on a call to ARIXEPP. From PPOPGNST, or from other structures it points to, ARIXEPP can find all the information it needs about the current SQL statement to pass on to the Parser, Optimizer, Code Generator, and Store Access Module Routine.

Dec(Hex) PPOPGNST

0 (0)	PPODCLPT - ↑ DCLLIST	PPOSLTEN - ↑ SLT entry	
8 (8)	PPOOPPTR - ↑ Op Tree	PPONEWOP - ↑ Code Block	
16(10)	PPOSTLDP - ↑ STOLDSTR	[1]	Reserved
24(18)	PPOSPACP - ↑ 1st Space Block		

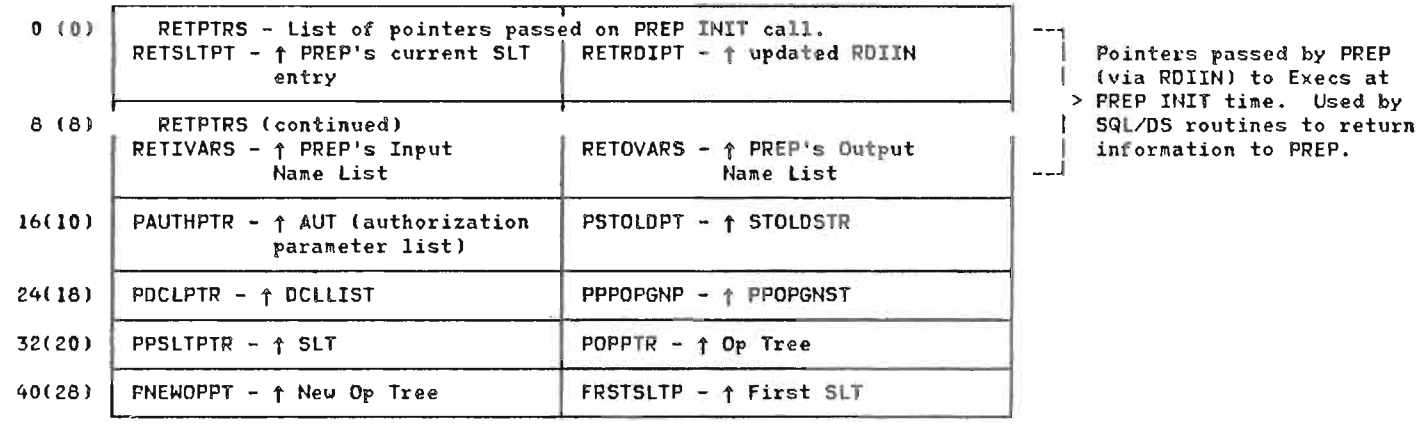
27(1B)

[1] PPOCHKMO 1... .... In Check mode.
PPOISCUR .1.. .... Processing CURSOR statement.
..xx xxxx Reserved.

PREP CONTROL BLOCK (PREPDSCB)

PREPDSCB is pointed to by field RDAPRPDS of the RDAREA. PREPDSCB acts as a save area for various pointers that are used over the different calls to ARIXEDS for the duration of processing a particular program. The pointers are set up on a PREP INIT call and not freed until PREP FINISH call.

Also see SQL/DS Logic, Volume 1, "RDS Executives and Module Flow" in Section 2 for an overview of the relationship among the data areas associated with PREPDSCB.



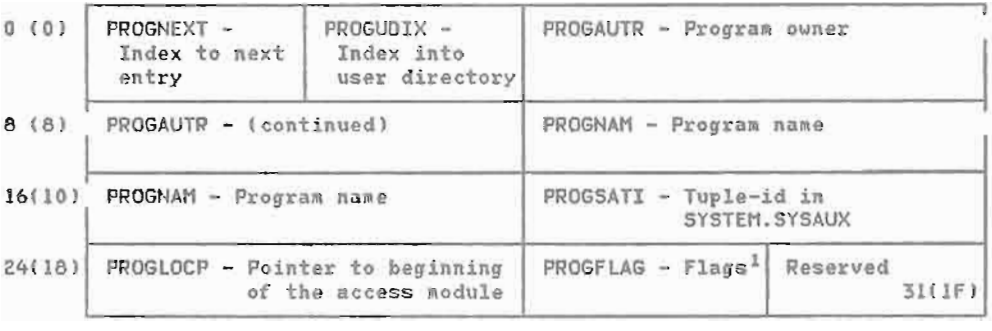
PROGS (PROGRAMS LOADED INTO STORAGE)

PROGS, a list of access modules that have been loaded into storage, is pointed to by RDAPROGS in the RDAREA. All users share PROGS but only one user at a time can change it. RDCLATCH controls use of PROGS and can be set only by the COMPARE AND SWAP instruction. CAUTION: Any change to the size of the PROGS structure will affect module ARIXCR4.

PROGS usage is managed by a series of index chains, which link a given PROGS entry. The blank list, anchored at RDCPBLNK, contains entries not having loaded access modules. After initialization the first 3*NCUSERS' PROGS entries are chained to the blank list. The remaining 7*NCUSERS are put on the extension list, which is anchored to RDCPEXT. The extension list is used only when no entries are available from the blank or free lists. RDS attempts to manage to the 3*NCUSERS level to avoid excessive use of storage.

After the system has been running, access modules may appear on two other chains - the RDAREA (RDAPROGS) chain and the free list (RDCPFREE). The RDAREA chain holds those access modules currently in a unit of work for the agent. The free list contains those loaded access modules not in use by an agent. These access modules are available to all agents.

Dec(Hex) PROGS



[1] PROGFLAG

Byte 1		
PROGLOCK	1...	Entry in SYSACCESS is locked
PROGINVD	.1..	Access module invalid
	..xx xxxx	Reserved
Byte 2	xxxx xxxx	Reserved

RDAREA (RDS CONTROL AREA)

Contains information relating to the RDS environment. Can be used to locate all SQL/DS and DBSS control areas. There is one RDAREA per agent (except for the Operator Agent and Checkpoint Agent) within SQL/DS. (Note: Whenever an RDS module is entered, register 10 always points to an RDAREA. Using this fact, we can find anything else.)

The RDAREA is pointed to by:

```
   DSCAREA (DSCRDAP,  offset X'1C') ┐
   YTABLE1 (YT1RDAP,  offset X'1C') ┘ > 1
   DCE     (DCERDAP,  offset X'1C') ┘
   RDCVT   (RDCRDAP,  offset X'18') 2
   RDAREA  (RDANEXT,  offset X'20') 3
```

For the following notes, also refer to the diagrams on pages 2 and 3.

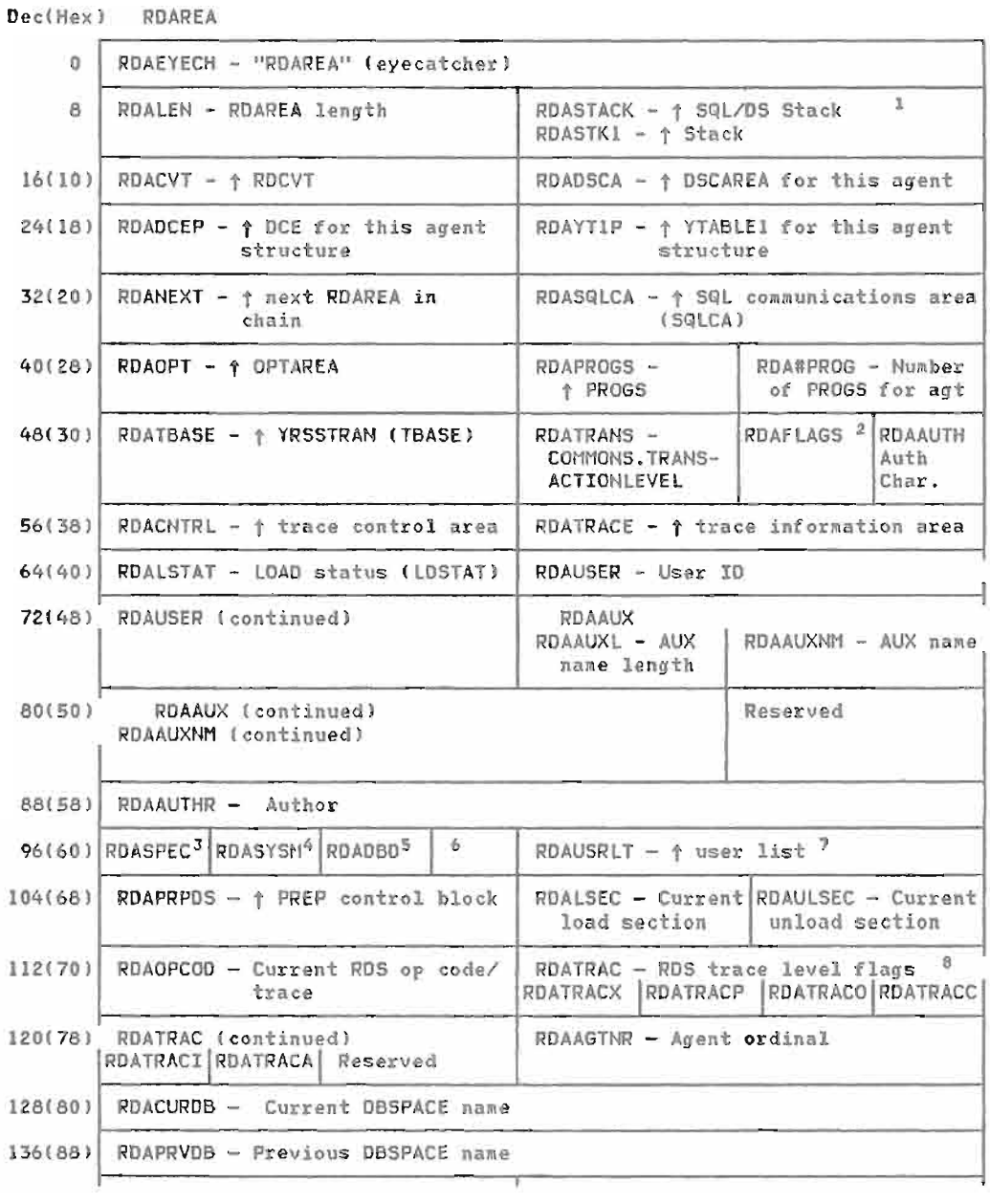
1 These pointers are only within an agent structure. That is, each of these control blocks is directly associated only with other control blocks within the same agent structure.

2 This points only to the first in the chain of RDAREAs (that is, to the RDAREA in the Ready/Recovery Agent (if it exists) or to the first General/Indoubt Agent structure)..

3 This points to the next RDAREA in the chain (for the next agent structure).

Alphabetic List of Field Names

RDAAGTNR 124(7C)	RDAINLUW 99(63) (flag bit)	RDATBASE 48(30)
RDAAUTH 55(37)	RDALEN 8(8)	RDATRAC 116(74)
RDAAUTHR 88(58)	RDALSEC 108(6C)	RDATRAC 116(74) (flag bit)
RDAAUX 76(4C)	RDALSTAT 64(40)	RDATRACA 121(79)
RDAAUXL 76(4C)	RDANEXT 32(20)	RDATRACC 119(77)
RDAAUXNM 78(4E)	RDAOPCOD 112(70)	RDATRACE 60(3C)
RDACNTRL 56(38)	RDAOPT 40(28)	RDATRACI 120(78)
RDACPLUW 54(36) (flag bit)	RDAPATCH 128(80)	RDATRACO 118(76)
RDACTABP 148(94)	RDAPREPM 54(36) (flag bit)	RDATRACP 117(75)
RDACVT 16(10)	RDAPROGS 44(2C)	RDATRACX 116(74)
RDADBD 98(62)	RDAPRPDS 104(68)	RDATRANS 52(34)
RDADCEP 28(1C)	RDAQRYNO 152(98)	RDAULSEC 110(6E)
RDADOREP 54(36) (flag bit)	RDARECUR 54(36) (flag bit)	RDAUPDGR 54(36) (flag bit)
RDADROP 54(36) (flag bit)	RDASPEC 96(60)	RDAUPLUW 54(36) (flag bit)
RDADSCA 20(14)	RDASQLCA 36(24)	RDAUSER 68(44)
RDAEYECH 0(0)	RDASTACK 12(C)	RDAUSRLT 100(64)
RDAFLAGS 54(36)	RDASTK1 12(C)	RDAYT1P 24(18)
RDAFLGS1 99(63)	RDASYSM 97(61)	RDA#PROG 46(2E)
RDAFRIND 86(56)		



[1] This field must stay at offset 12(C) to conform to the equivalent field in YTABLE1.

[2] See "RDA Flags" on page 158.

[3] Used by SCHEDULE

[4] SQL/DS Mode:

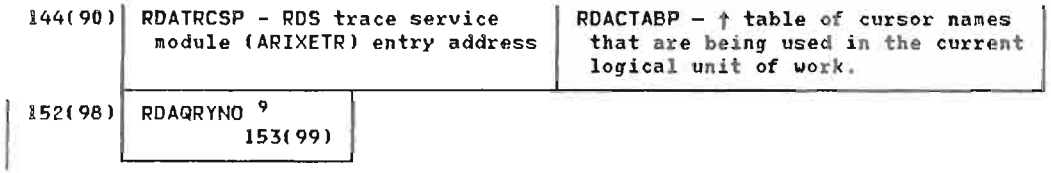
[5] RDADBD

```
RDADBDR 1... ....  Explain "reference"
RDADBDS .1.. ....  Explain "structure"
RDADBDC ..1. ....  Explain "cost"
RDADBDP ...1 ....  Explain "plan"
        .... xxxx  Reserved
```

[6] RDAFLGS1 - See "RDA Flags" on page 158.

[7] User list contains a list of users that can run the identified access module.

[8] 0 if no trace, else 1 or 2. See "Trace Flags" on page 158.



[9] Query number for "explain"

RDA Flags

Offset	Field Name	Bits	Meaning
54(36)	RDAFLAGS		
	RDADROP	0...	SYSDROP is empty.
		1...	SYSDROP is not empty.
	RDADOREP	.0..	Don't do Re-PREP
		.1..	Do Re-PREP
	RDASCHED	..0.	SCHEDULE CONNECT has not been issued.
		..1.	SCHEDULE CONNECT has been issued.
	RDAPREPM	...0	Run-time mode.
		...1	PREP mode.
	RDACPLUW	.... 1...	A CREATE PROGRAM has been issued in the current logical unit of work.
	RDARECUR	.... .1..	ARIXERD1 has been entered during termination. (Prevents recursive entry by termination.)
	RDAUPLUW	.... ..1.	A DROP or PREPARE has been issued against a modifiable access module in the current logical unit of work.
	RDAUPDGR	1	Run authority with GRANT existed at the start of the UPDATE logical unit of work.
99(63)	RDAFLGS1		
	RDAINLUW	1...	A logical unit of work is currently in progress.

Trace Flags

Offset	Field Name	Meaning
116(74)	RDATRAC	RDS trace-level flags
116(74)	RDATRACX	Executives component trace level
117(75)	RDATRACP	Parser component trace level
118(76)	RDATRACO	Optimizer component trace level
119(77)	RDATRACC	Code Generator component trace level
120(78)	RDATRACI	Interpreter and Authorization components trace level
121(79)	RDATRACA	Security/Audit trace level
122(7A)	*	Two extra trace-level flags

RDCVT (RDS COMMUNICATION VECTOR TABLE)

The RDCVT is the global RDS control block. The RDCVT contains information and pointers to other control blocks that are used by every RDS module. The RDCVT will be pointed to from any RDAREA. There is only one RDCVT in the system.

The RDCVT is pointed to by:
- DS2CVT (DS2RDCVP, offset X'14')
- YRSSCVT (YRSRDCVP, offset X'14')
- RDAREA (RDACVT, offset X'10')

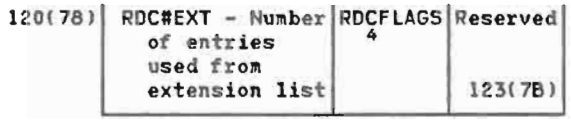
Dec(Hex) RDCVT

0(0)	RDCEYECH - "RDCVT" (eyecatcher)			
8(8)	RDCVTLEN - RDCVT length		RDCPATCH - ↑ RDS patch area.	
16(10)	RDCDS2CV - ↑ DS2CVT		RDCYRSCV - ↑ YRSSCVT	
24(18)	RDCRDAP - ↑ RDAREA chain		RDCPLIRS - Address of DBSS DBSI	
32(20)	RDCCATS - ↑ Catalog information		RDCOSTYP - Type of Op System	RDCFILL - Reserved
40(28)	RDCSTART - Address of ARIXEST		RDCRDI - Address of ARIXERD	
48(30)	RDCSTOP - Address of ARIXESP		RDCRDOPT - Address of ARIXERO	
56(38)	RDCDBUF - Number of page (data) buffers		RDCBBUF - Number of block buffers	
64(40)	RDCBRCH - Address of ARIXEBR		RDCBRCL - Address of ARIXEBRl	
72(48)	RDCDPATH - Float value (1200) minus average number of instructions per DBSS call		RDCRATIO - Float value (1/3600)[1]	
80(50)	RDCTRAC - RDS Trace flags			
88(58)	RDCCWl - Address of entry point ARIXECWl		RDCLATCH - Flag for holding PROGS during use by an agent [2]	
96(60)	RDCUDIRP - Address of start of user director (UDIRECT)		RDCUSERL - Address of start of user list (USRLIST)	
104(68)	RDCPROGS - Address of PROGS structure		RDC#PROG - No. of PROGS slots	RDCPFREE [3]
112(70)	RDC#USER - Number of slots in user list	RDCUFREE - Index into user list of lst free slot or zero	RDCPBLNK - Index into PROGS of lst PROG in blank list	RDCPEXT - Index into PROGS of lst entry in extension list

[1] Used in Optimizer as ratio of importance between CPU cost and I/O cost. 1/RDCRATIO approx. = number of instructions to be weighed as equivalent to one I/O.

[2] RDS Release 2 protocol requires an agent to obtain this latch (set it to the agent ordinal) prior to performing any operations involving the PROGS, USERLIST, or user directory control blocks. No other I/O operations or other DBSS calls should be executed while holding this latch. The exception is the explicit share lock requested on the SYSACCESS TID (SATI) when an access module is reused from the FREE list. In that case, RDCPLOCK bit is set to indicate that we are in lock processing while the latch is being held. Users who would normally hold this lock and request the latch also (currently the invalidate access module function) must not wait for the latch. This user must roll back to avoid a potential deadlock between the SATI lock and the RDS latch.

[3] Index into PROGS of the first slot in the FREE list or zero.



4 RDCFLAGS

Byte 1		
RDCPLOCK	1...	Set only in ARIXERD when locking SYSTEM.SYSACCESS
RDCALUSR	.1..	Indicates implicit CONNECT
	..xx xxxx	Reserved
Byte 2	xxxx xxxx	Reserved

RDIIN

The RDIIN is the control block passed to SQL/DS on all calls to SQL/DS. (RDICTYPE indicates the call types.) Call types are illustrated on the following pages. It is passed to the proper SQL/DS handler for the call type by ARIXERD.

Dec(Hex) RDIIN

0 (0)	RDICTYPE - [1]	RDIAUTHR - Creator of access module [2]					
8 (8)	RDIAUTHR (continued)	RDIPROGL - [3] Length of module name	RDIPROGN - Access module name [3]				
16(10)	RDIPROGN (continued)		RDISECT# - [4]	RDICLSSC - [4]			
24(18)	RDICODEP - ↑ error code structure (SQLCA)		RDIVPARM - [5]				
32(20)	RDIAUXPA - [6]		RDISQTIE - ↑ SQLTIE				
40(28)	RDICNFLG		RDIVIND [7]	RDIAIND [7]	RDIERROR [7]	RDIFILL [7]	
	RDISPEC [7]	RDICALL [7]	RDIWAIT [7]	RDIRELSE [7]			
48(30)	RDIMBLEN - Length of input Mailbox [8]			RDIRELNO [7]	RDIRESVD - 15 bytes reserved		
56(38)							
64(40)	67(43)						

[1] RDICTYPE:
- 30 - AUX CALL
- 35 - SETUP CALL
- 40 - DESCRIBE CALL
- 45 - CLOSE CALL
- 50 - OPEN CALL
- 120 - CONNECT/SCHEDULE CALL
- 125 - RECOVERY LIST CALL
- 130 - PREPINIT CALL
- 131 - CREATE PROGRAM CALL
- 132 - DROP STATEMENT CALL
- 135 - LOOKUP CALL
- 140 - SQL CALL
- 145 - PREPFINISH CALL
- 155 - OPERATOR COMMAND CALL
- 160 - PREPARE-TO-COMMIT CALL
- 165 - SET/RESET EXIT CALL
- 166 - MODIFY CANCEL CALL
- 170 - OPERATOR COMMAND CONTINUE CALL

Examples of each type follow this figure.

[2] For statements with a userid modifier on "IN program" clause or in CREATE PROGRAM, this is set to that value. When userid is defaulted to in these cases, it is set to blanks. In other cases, the value is taken from PREP parameters or defaults.

[3] For statements with "IN program" clauses, these are set to that value. In other cases, the value is taken from PREP parameters or defaults.

[4] - See page 162.

[7], [8] - See page 163.

[5] RDIVPARM points to:
- Input Name List for PREP-INIT.
- Input or output SQLDA on EXECUTE, OPEN, or FETCH.
- SQL statement on SET-UP CALL, SQL CALL, and EXEC IMMEDIATE AUX CALL.
- User SQLDA on DESCRIBE.
- userid on CONNECT.
- ↑ input RMRE on GET RECOVERY LIST or PREPARE-TO-COMMIT; ↑ input OCOMBLK on OPERATOR.
- I/O variable pointers on PREP-INIT call.
- RDIIN pointer on LOOK-UP calls (for feedback).

[6] RDIAUXPA points to:
- Input SQLDA on AUXCALL.
- Password on CONNECT.
- Output Name List on PREP-INIT.
- Name on LOOKUP (search argument for comparison against PLSTCNAME of SLT).
- Token/DCLLIST/caller's RDIIN on SQLCALL.
- User SQLDA on DESCRIBE.
- ↑ output RMRE on GET RECOVERY LIST or PREPARE-TO-COMMIT; ↑ output OCOMBLK on OPERATOR.
- Cursor name for statements with a host variable cursor-name.
- Pointer to a pointer pair on a SETUP CALL (see page 167).

4 RDICLSSC/RDISECT#: For SQLCALLS, RDICLSSC = 0 and RDISECT# = 0 on input; SQL/DS returns RDISECT#. Look up calls are explained below.

LOOK-UP CALL FOR:	RDICLSSC Input Value	RDICLSSC Output Value:	Meaning
PREPARE or DECLARE CURSOR for SELECT (non-extended form of PREPARE/DECLARE)	-1	-1: 0:	Establish a new name. Section number assigned. Name already used. No section assigned.
EXECUTE IMMEDIATE, ROLLBACK WORK, COMMIT WORK	-1	-1: 0:	Establish a new name ('EXECUTE', 'ROLLBACK', or 'COMMIT'). Section number assigned. Name already used. No section assigned. Allows sharing of a single section by AUX calls of the same type.
DECLARE CURSOR FOR stmt-name, OPEN, FETCH, CLOSE, EXECUTE, DESCRIBE, (non-extended form of the statements)	0	0: -1:	Old name found in PSLTCNAM. Set section number in RDISECT#. Old name not found in PSLTCNAM. Error. No section returned.
DECLARE CURSOR (variable statement-id) (extended form of DECLARE CURSOR)	0		Use section number to locate section. Verified that cursor name not previously used. If cursor name used previously, return negative SQLCODE in SQLCA.

[7] RDIIN Flags

Offset	Field Name	Bits	Meanings
40(28)	RDISPEC		
	(PREP Time only)		
	RDIDCLST	1...	DCLLIST must be sent to SQL/DS on SQLCALL.
	RDIISCUR	.1..	Causes SQL/DS to turn ISCURSOR bit on in PPOPGNSTR. Set by caller on SQL call if prompted by "DECLARE_CURSOR FOR SELECT ..." (In Release 1, preprocessors initialized RDISPEC to X'40' which turned this bit on. Since this is used only at PREP time, there is no conflict.)
	RDICHECK	...1	Tell SQL/DS we are in check mode (PREP INIT).
	RDIKEEP	.... 1...	Tell SQL/DS to keep current run authorization if PREP'ed again (PREP INIT and CREATE PROGRAM).
	(Run Time - CREATE PROGRAM Assembler PREP only)		
	RDIDESCR	.... .1..	Allow DESCRIBEs. Specified for CREATE PROGRAM.
	RDINEW	.... ..1.	New access module on CREATE PROGRAM.
	RDIMODFY	1	Establish the access module as modifiable.
41(29)	RDICALL	character:	'E' (COMMIT), 'R' (ROLLBACK), 'C' (CONNECT), 'S' (SET) or 'R' (RESET) on SET/RESET exit call, 'I' for implicit CONNECT on PREP INIT call, 'B' for Extended EXECUTE or Extended DESCRIBE, 'D' for Extended Declare LOOKUP call, or ' '. These are unique only within a particular call type.
42(2A)	RDIWAIT	character:	Always ' '. (Unused)
43(2B)	RDIRELSE	character:	'R' or ' '.
44(2C)	RDIVIND	character:	'S', 'I', 'O' or ' ' for AUXCALL.
45(2D)	RDIAIND	character:	'I' or ' ' for AUXCALL. 'I' on OPEN with input parameters.
46(2E)	RDIERROR	character:	'E', 'W', 'B', or ' '. Tells Communication Manager to check for errors, warnings, or both.
47(2F)	RDIFILL - Reserved		
52(34)	RDIRELNO	character:	SQL/DS release number in which the RDIIN was used/generated (by a Preprocessor). ' ' or X'00' means Release 1, X'02' means Release 2, and so forth.

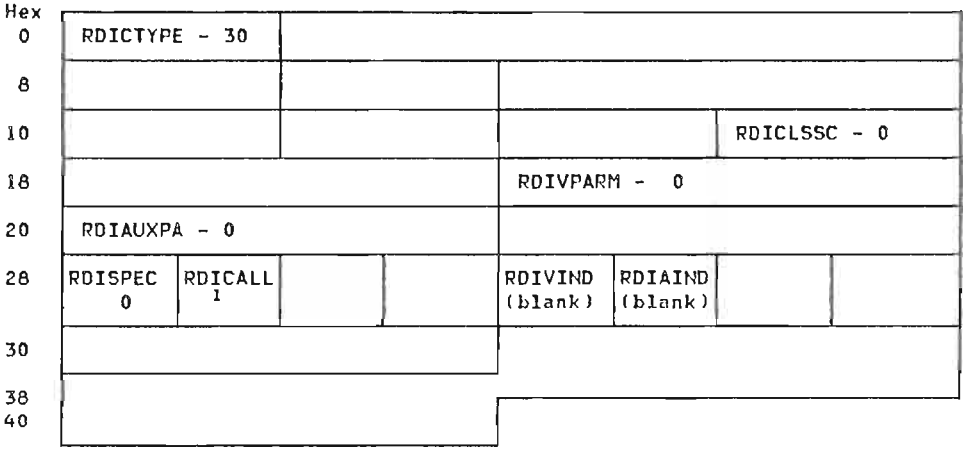
[8] Mailbox contents and lengths:

- Mailbox header — 16 bytes
- RDIIN — 68 bytes
- SQLDA — 16 bytes + 44 bytes per variable
- Input Variables — Number of bytes required to store these variables.
- Other values passed to SQL/DS (SQL statements, user ID, etc. — Number of bytes required to store the values.

RDIIN EXAMPLES

The following are examples of the RDIIN for the various call types. Omitted fields are set as described earlier.

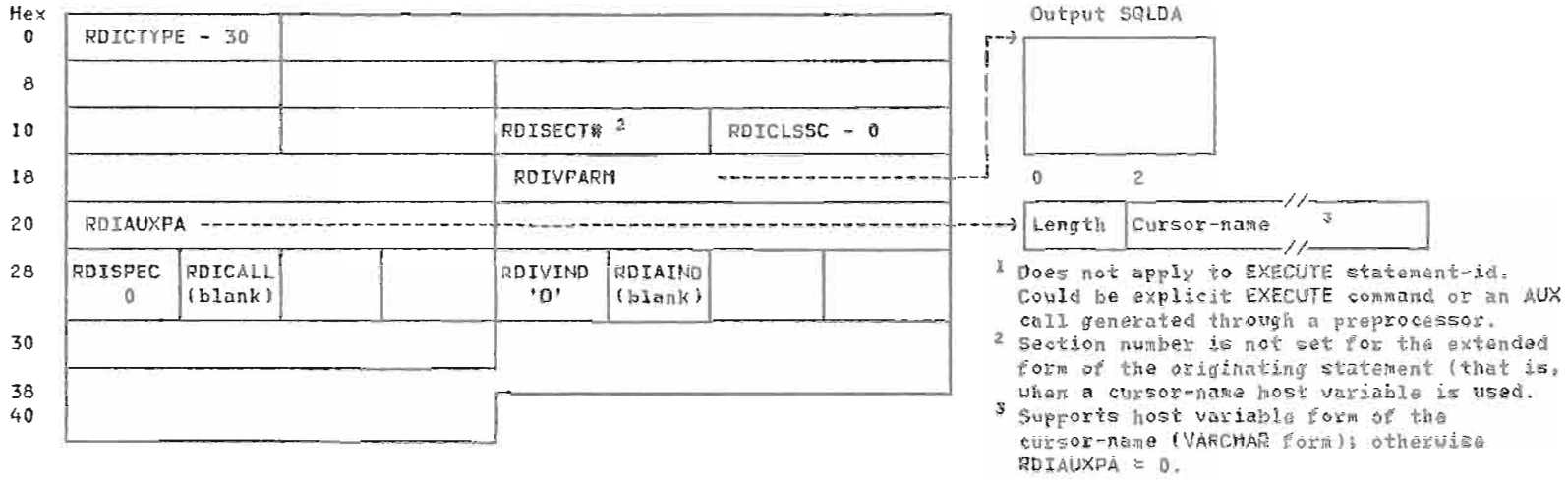
AUXCALL (No Host Variables) (EXECUTE with no host variables; also, COMMIT WORK and ROLLBACK WORK, FETCH with no 'INTO')



[1] RDICALL = 'E' if COMMIT WORK
'R' if ROLLBACK WORK
'B' for Extended EXECUTE
' ' anything else

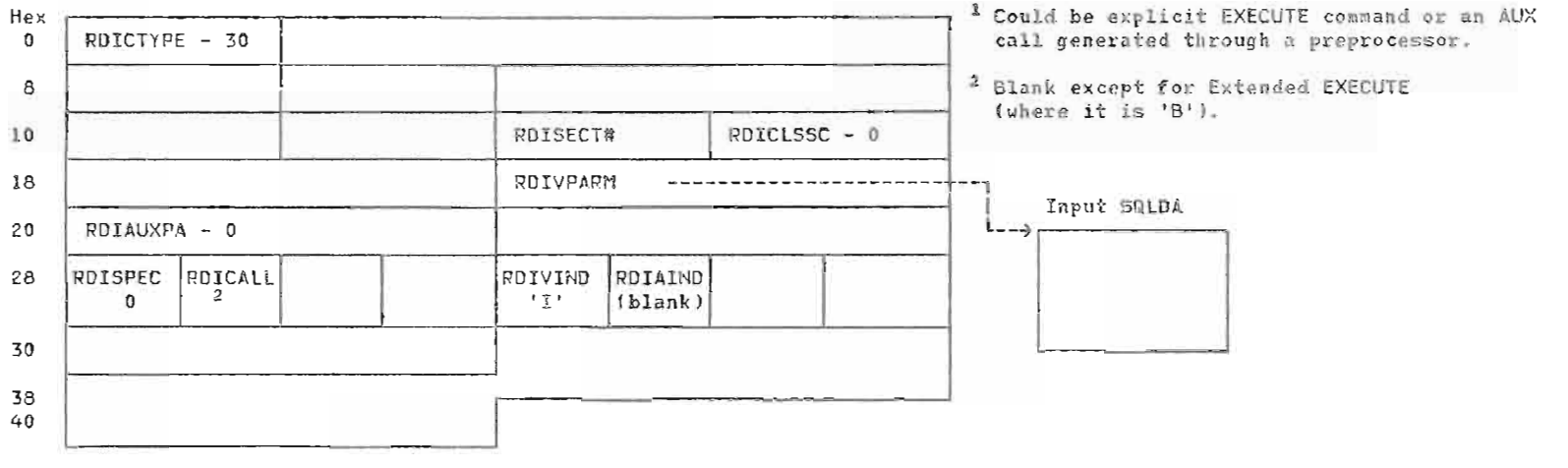
RDIIN Examples (continued)

AUXCALL (Output Host Variables)
(A. EXECUTE of Single Row SELECT with No Input Variables [1]
(B. FETCH



[1] Does not apply to EXECUTE statement-id. Could be explicit EXECUTE command or an AUX call generated through a preprocessor.
[2] Section number is not set for the extended form of the originating statement (that is, when a cursor-name host variable is used.
[3] Supports host variable form of the cursor-name (VARCHAR form); otherwise RDIAUXPA = 0.

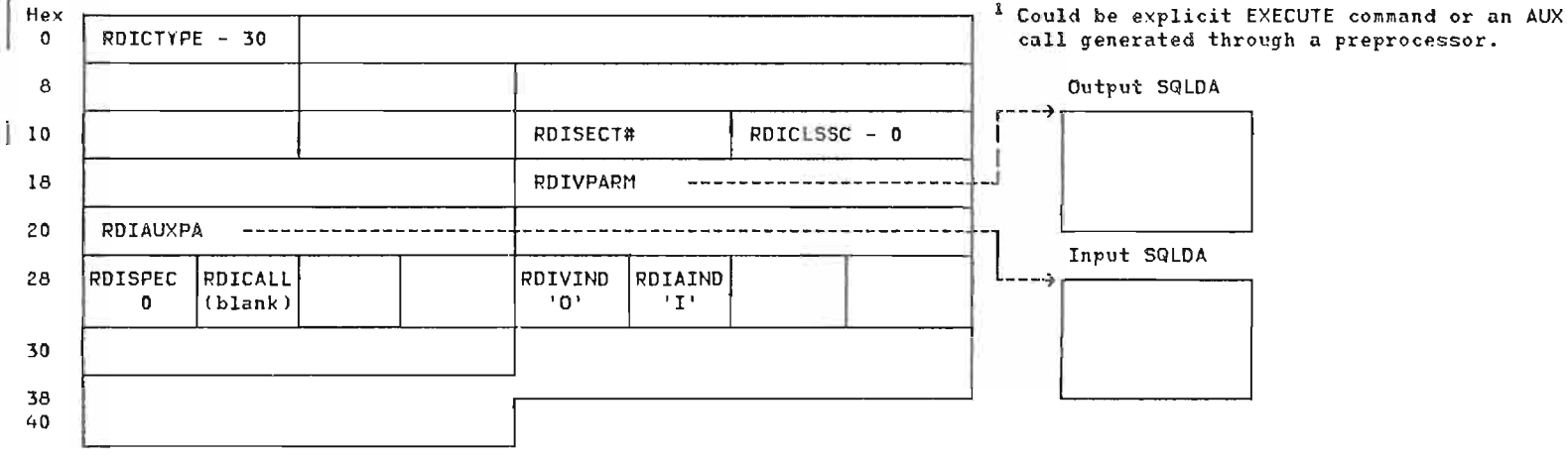
AUXCALL (Input Host Variables) (EXECUTE of INSERT/UPDATE) [1]



[1] Could be explicit EXECUTE command or an AUX call generated through a preprocessor.

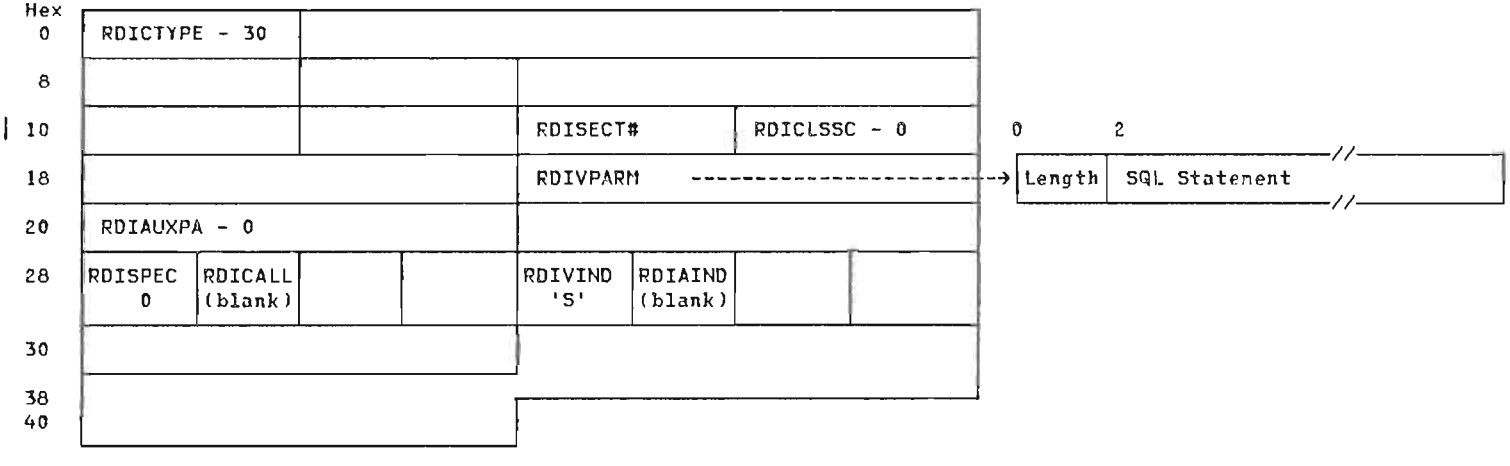
[2] Blank except for Extended EXECUTE (where it is 'B').

AUXCALL (Both Input and Output Host Variables) (EXECUTE of Single Row SELECT with Input Variables) [1]

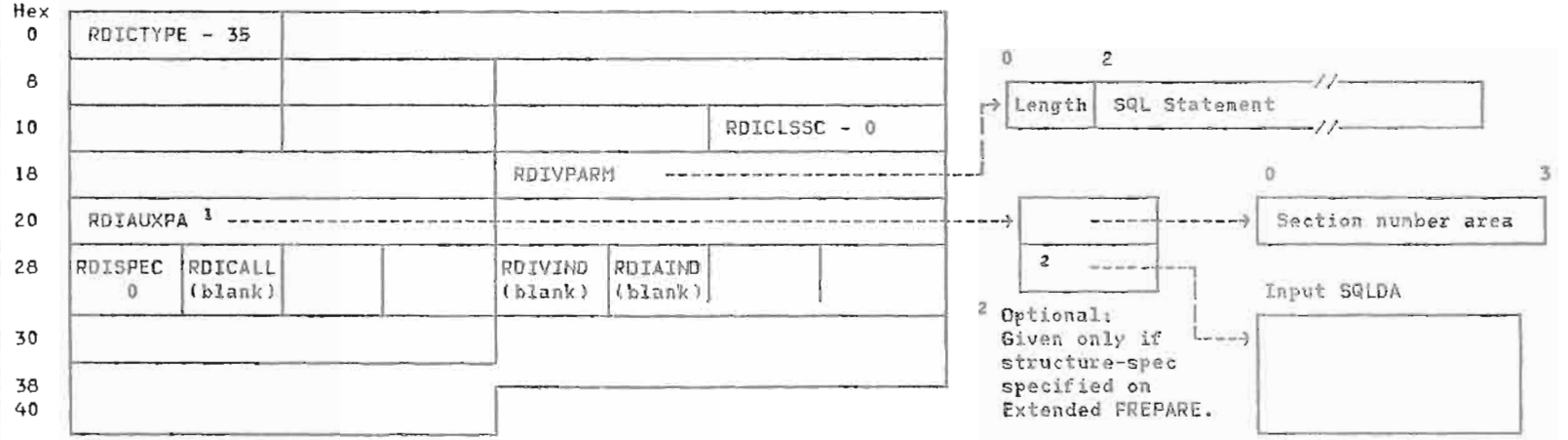


[1] Could be explicit EXECUTE command or an AUX call generated through a preprocessor.

AUX CALL (Execute Immediate)

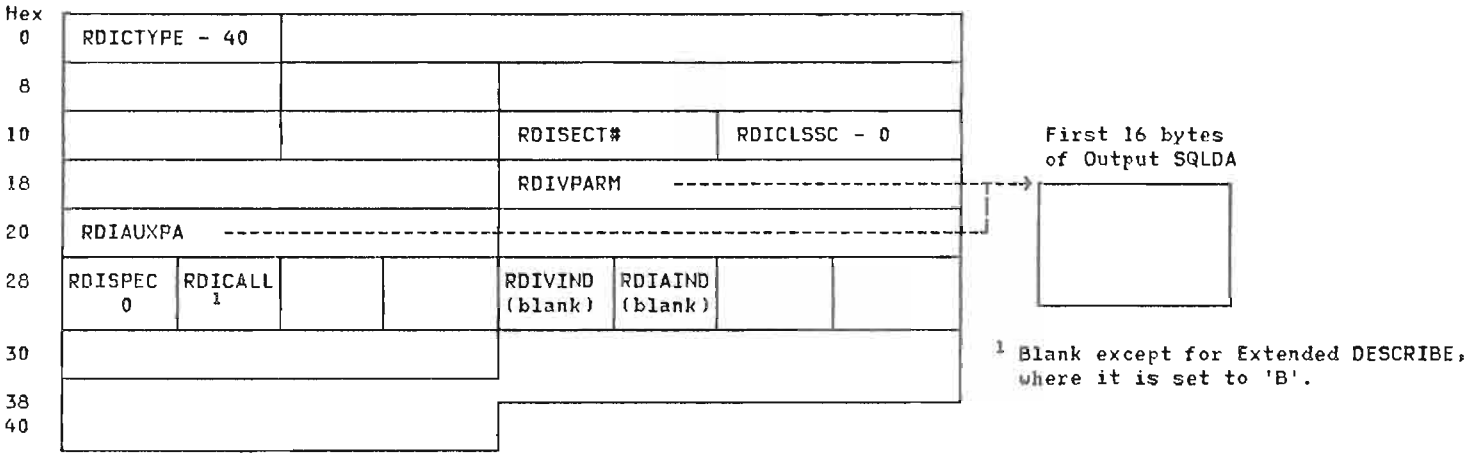


SETUP CALL (Prepare)



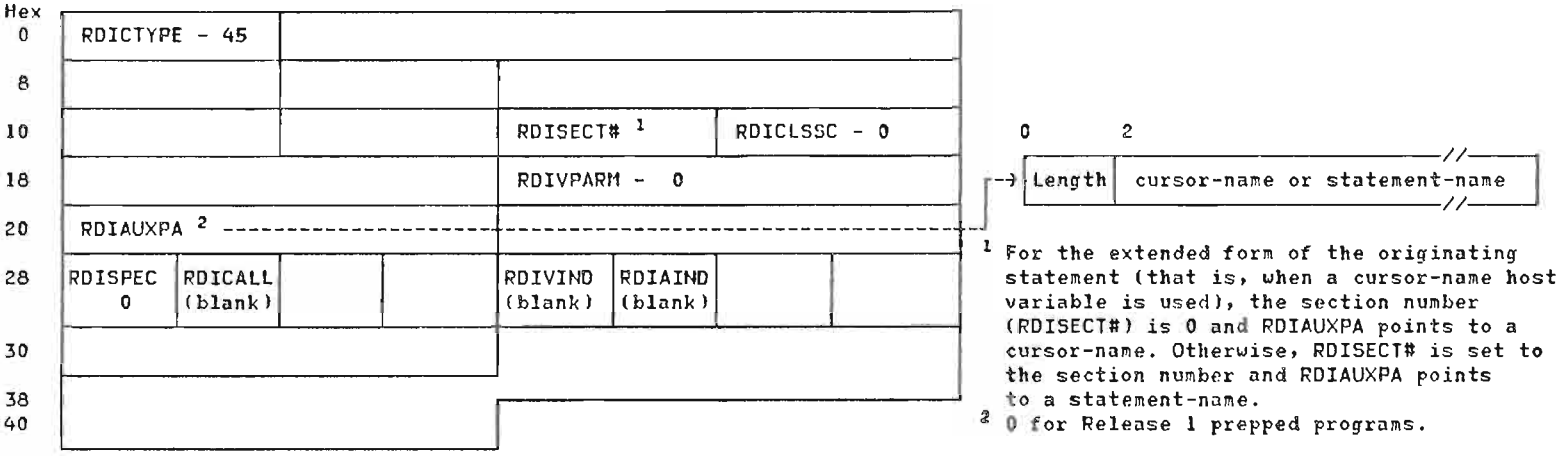
1 This applies to PREPARE with a statement-id only (RDISECT# is 0). If a statement-name is given, RDIAUXPA is 0. The section number area is in the caller's storage an is an integer host variable, which SQL/DS sets to the statement-id as a result of this call.

DESCRIBE CALL



[1] Blank except for Extended DESCRIBE, where it is set to 'B'.

CLOSE CALL

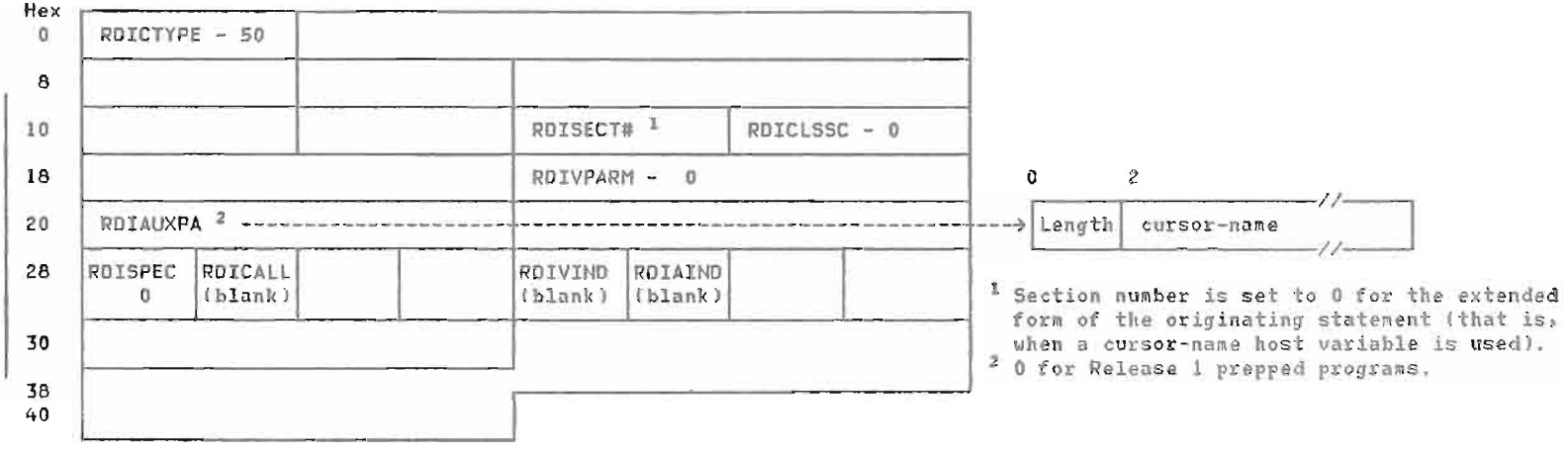


[1] For the extended form of the originating statement (that is, when a cursor-name host variable is used), the section number (RDISECT#) is 0 and RDIAUXPA points to a cursor-name. Otherwise, RDISECT# is set to the section number and RDIAUXPA points to a statement-name.

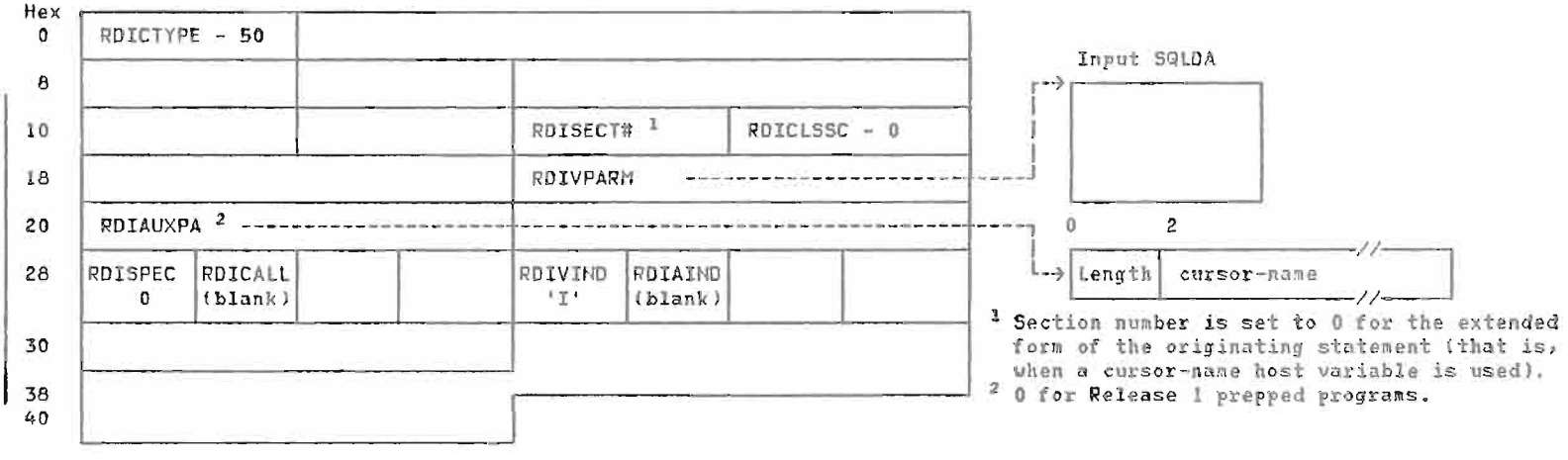
[2] 0 for Release 1 prepped programs.

RDIIN Examples (continued)

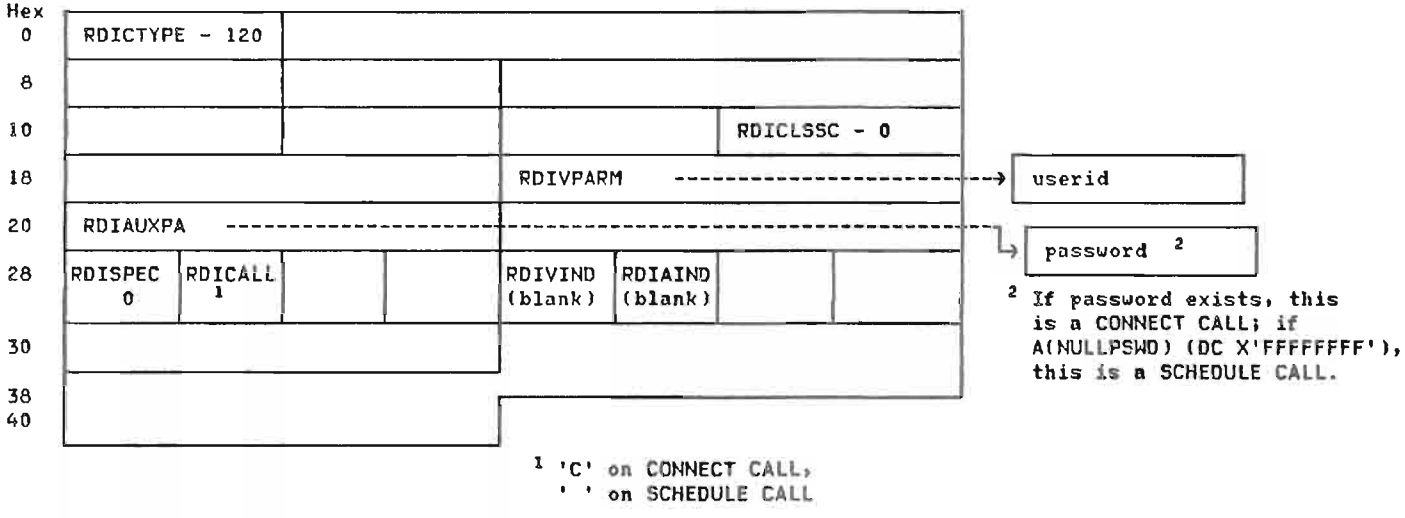
OPEN CALL (No Input Host Variables)



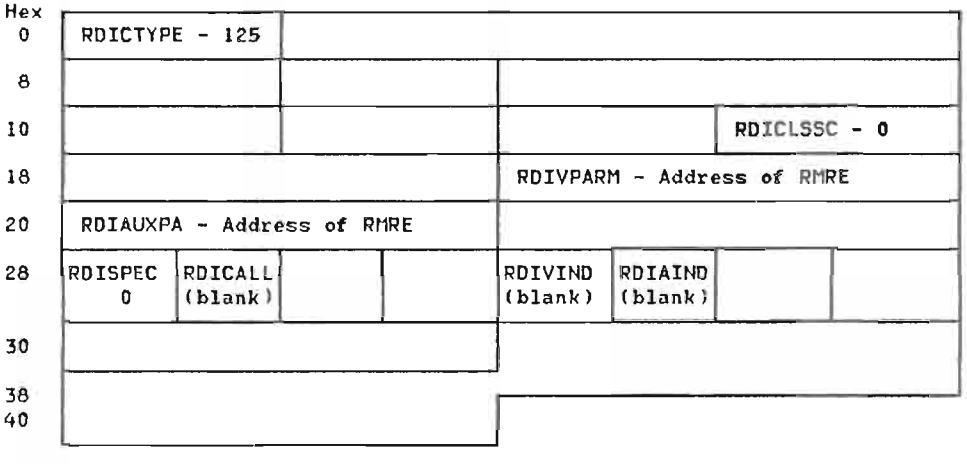
OPEN CALL (Input Host Variables)



CONNECT/SCHEDULE CALL



RECOVERY LIST CALL



RDIIN Examples (continued)

PREPINIT CALL

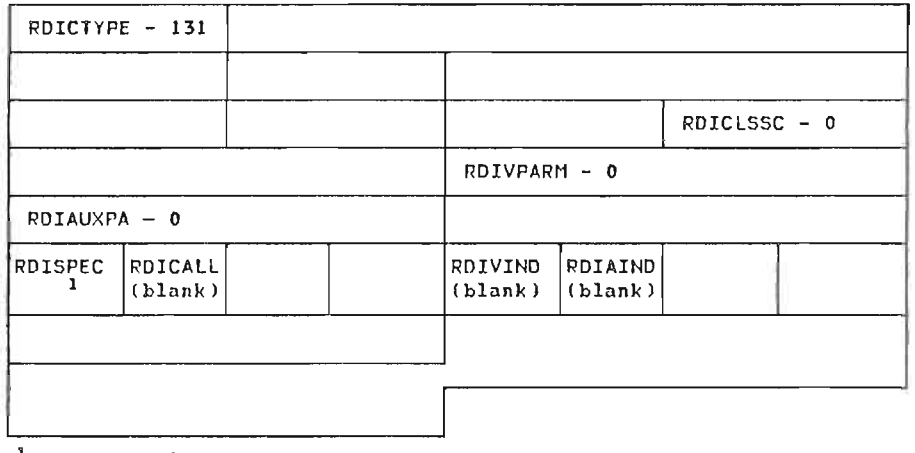
Hex								
0	RDICTYPE - 130							
8								
10							RDICLSSC - 0	
18					RDIVPARM - ↑ Input Name List[1]			
20	RDIAUXPA - ↑ Output Name List[1]							
28	RDISPEC [2]	RDICALL [3]			RDIVIND (blank)	RDIAIND (blank)		
30								
38								
40								

[1] Input Name List and Output Name List are Preprocessor areas used to receive SQL/DS feedback from subsequent calls to SQL/DS. The pointers to these two areas are saved in PREPDSCB by the RDS Executives.

[2] RDISPEC Bits:
3 (RDICHECK) = 0 - Not Check mode
1 - Check mode
4 (RDIKEEP) = 0 - Revoke authorized users
1 - Keep authorized users

[3] blank or 'I'. Upon return from the PREP INIT call (for VM implicit CONNECT), the calling preprocessor sets RDICALL = 'I' and loads the first 4 bytes of RDAUTHR with a pointer to an empty 8-byte area to receive the userid from RDS.

CREATE PROGRAM CALL

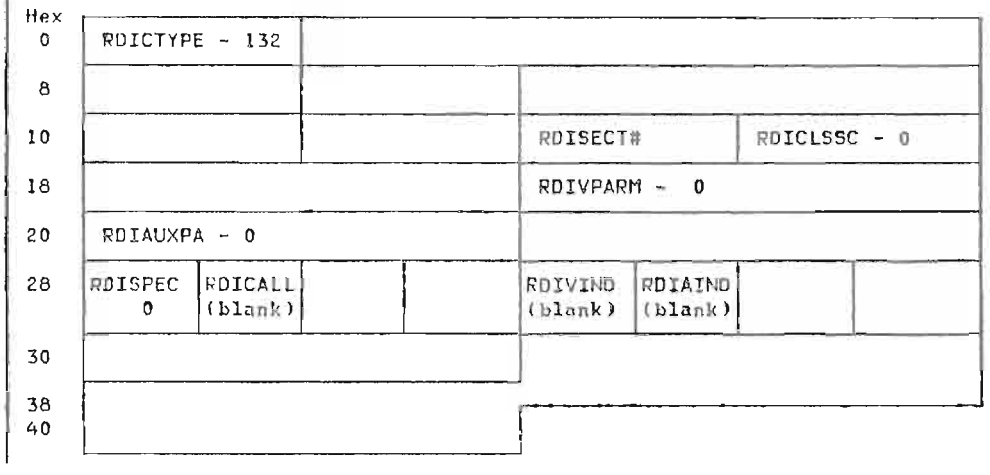


[1] RDISPEC Bits:

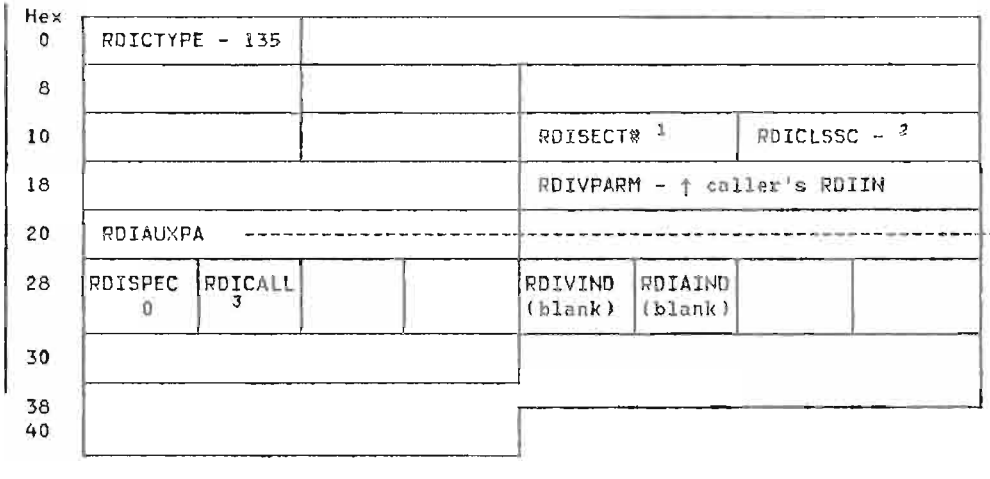
- 3 (RDICHECK) = 0 - Not Check mode
- 4 (RDIKEEP) = 0 - Revoke authorized users
  1 - Keep authorized users
- 5 (RDIDESCR) = 0 - Do not allow DESCRIBEs
  1 - Allow DESCRIBEs
- 6 (RDINEW) = 0 - Replace an old access module
  1 - This is a new access module
- 7 (RDIMODFY) = 0 - No modify
  1 - Modify

RDIIN Examples (continued)

DROP STATEMENT CALL



LOOKUP CALL

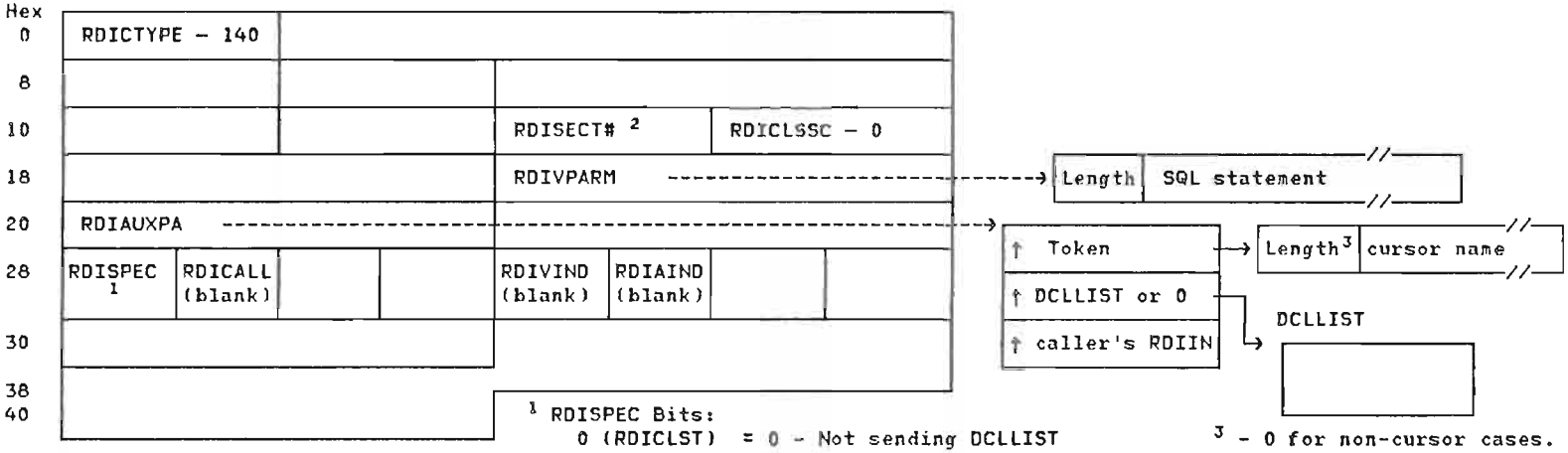


[1] If LOOK-UP call is for DECLARE CURSOR with a statement-id (run-time), the caller sets the section number; otherwise the call otherwise it is set to 0.

[2] RDICLSSC (set by caller)
= 0 if name expected to be found (existing name)
-1 if name not expected to be found (new name)

[3] blank, or 'D' for LOOKUP CALL for Extended DECLARE CURSOR.

SQLCALL

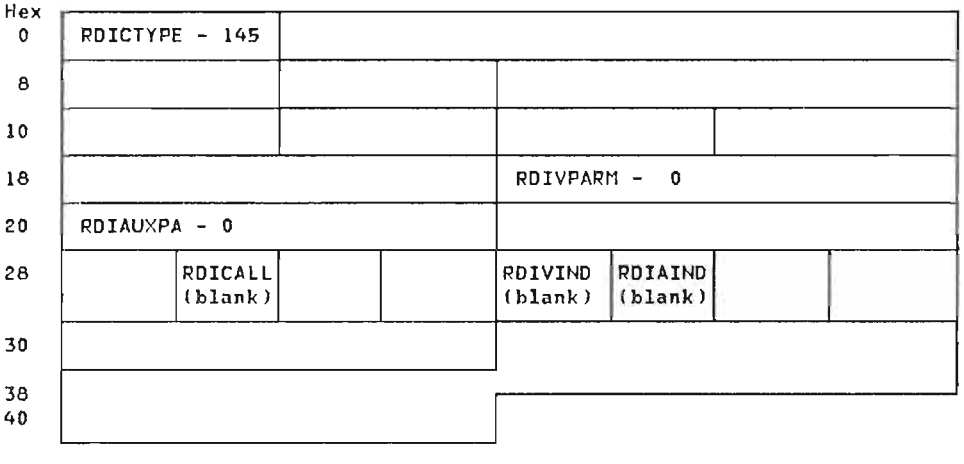


1 RDISPEC Bits:
0 (RDICLST) = 0 - Not sending DCLLIST
- = 1 - Sending DCLLIST
1 (RDIISCUR) = 0 - Not CURSOR statement
1 - CURSOR statement

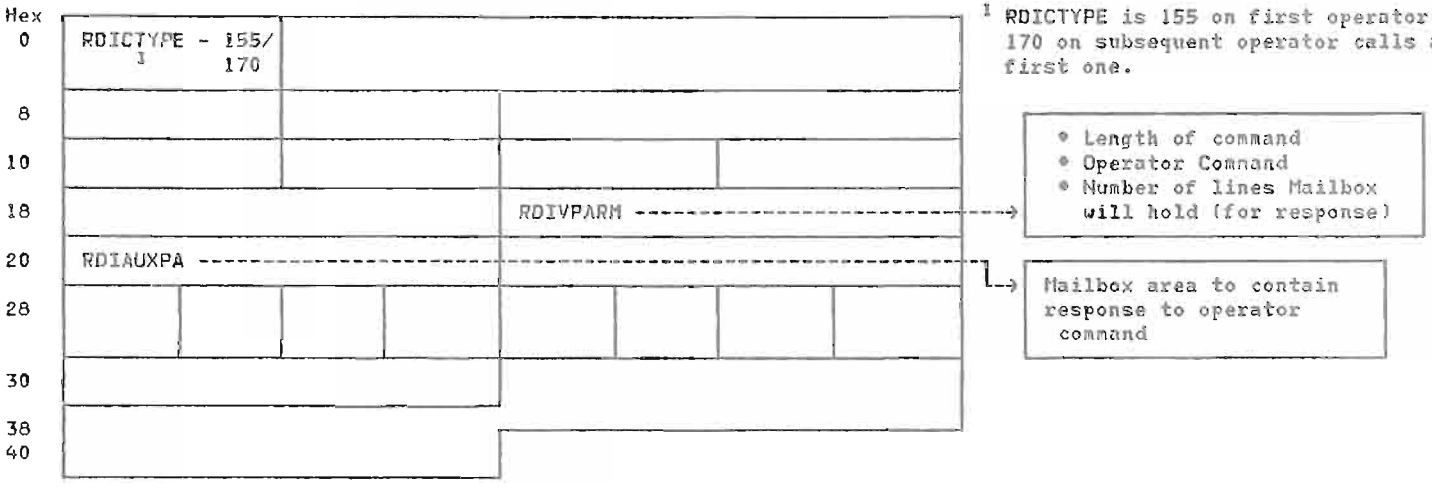
2 Value is returned by SQL/DS.

3 - 0 for non-cursor cases.

PREPFINISH CALL

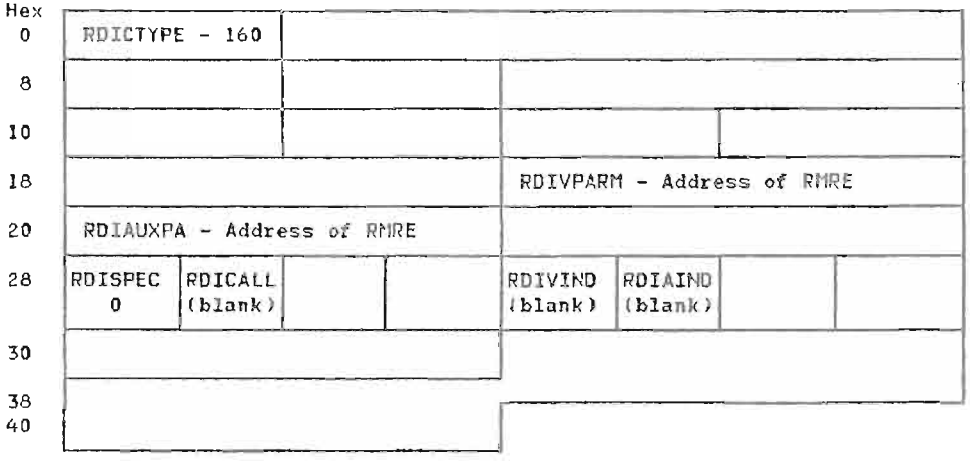


OPERATOR COMMAND CALL (Commands Issued from ISQL)



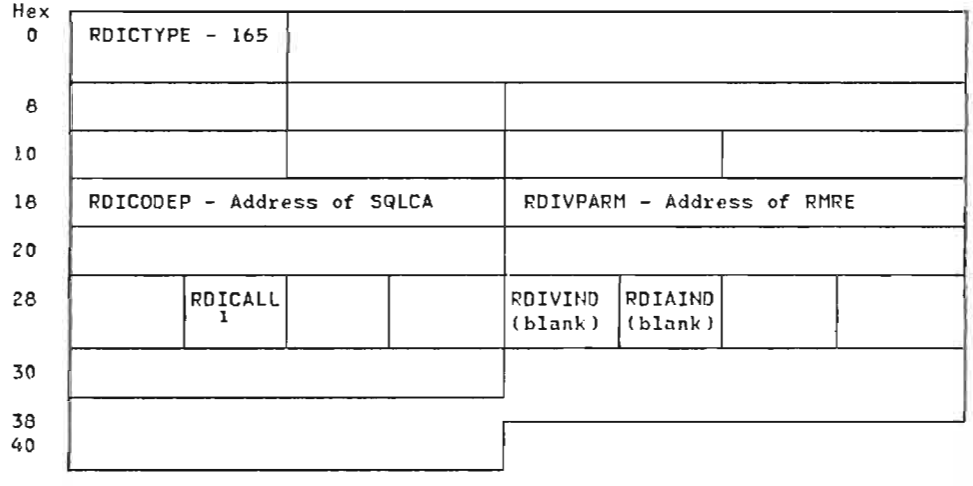
[1] RDICTYPE is 155 on first operator call and 170 on subsequent operator calls after the first one.

PREPARE-TO-COMMIT CALL



SET/RESET EXIT CALL

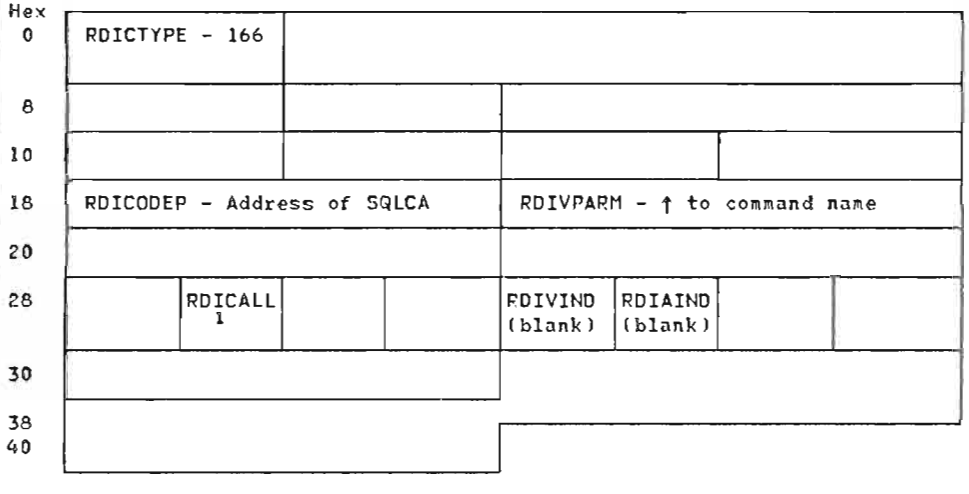
Normal SQLERROR, SQLWARNING settings are allowed.
SET/RESET EXIT is not sent to SQL/DS



1 'S' for SET EXIT
'R' for RESET EXIT

MODIFY CANCEL CALL

Modify Basic VM Cancel Support.
Modify Basic is not sent to SQL/DS



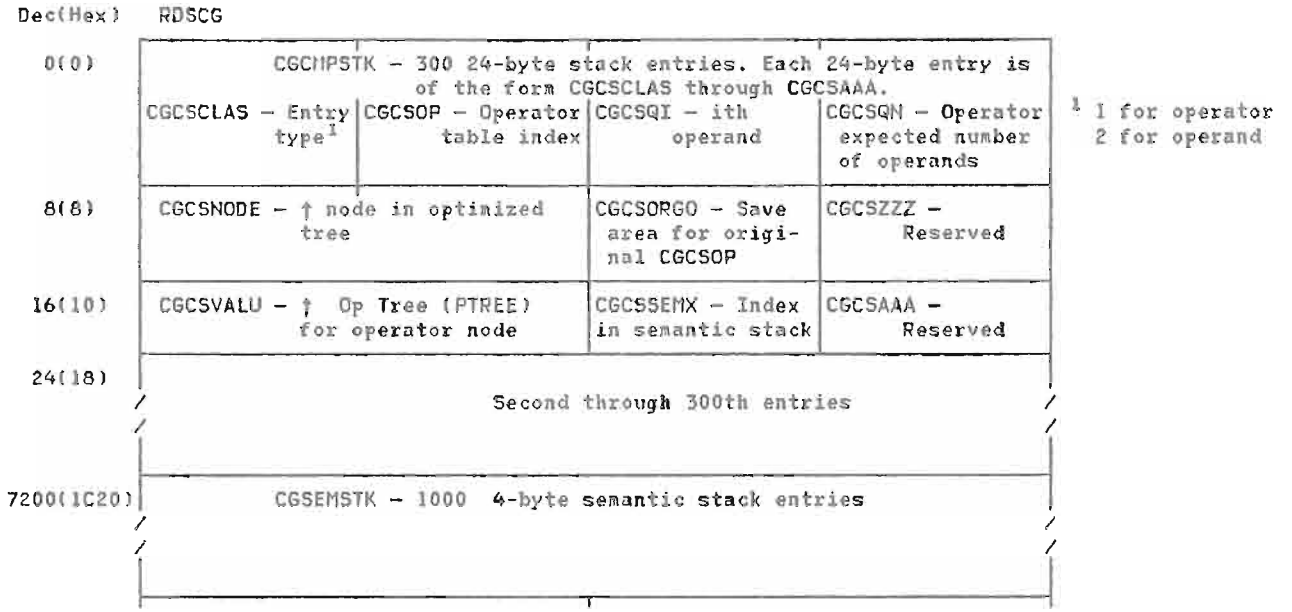
1 'S' for Specify additional immediate command
'R' for Remove the 'SQLHX' immediate command
The address of the RMXC is returned in Reg 15

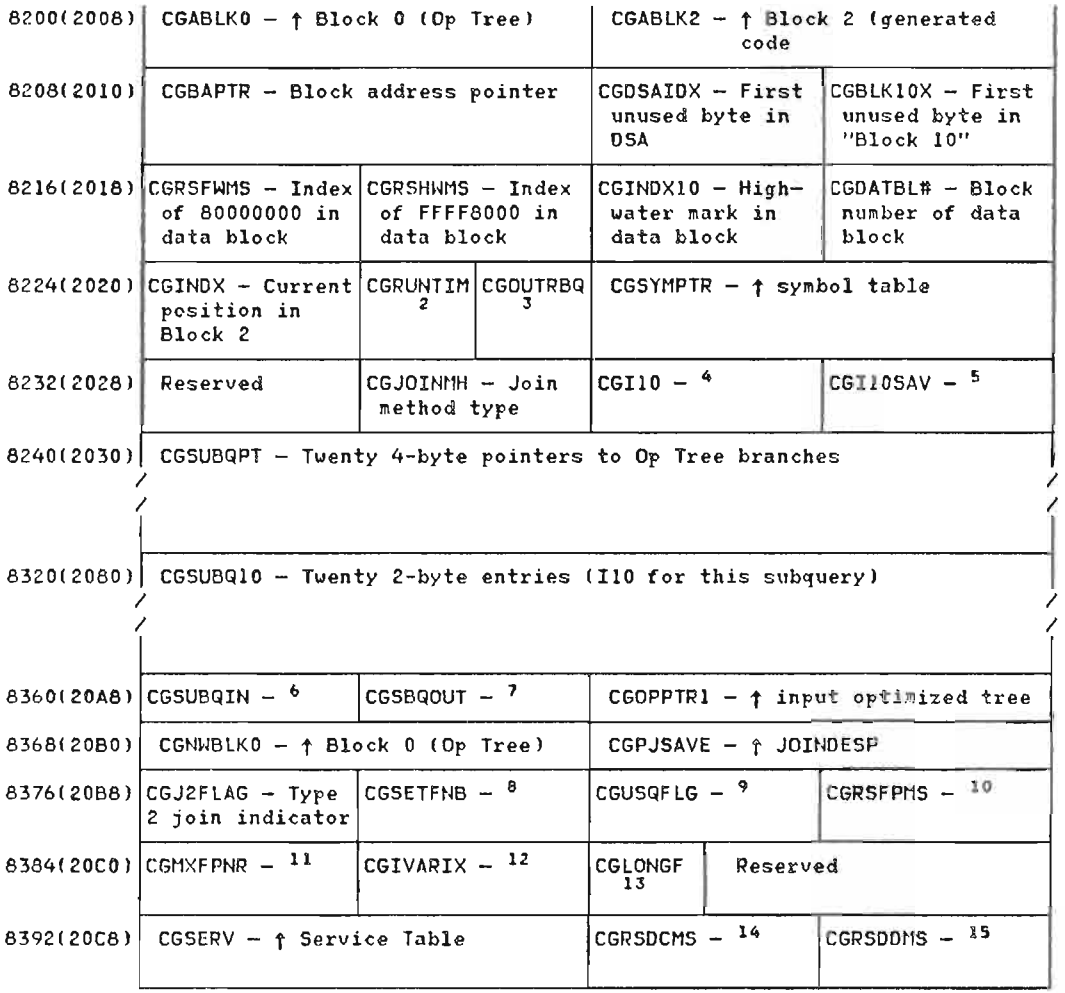
RDSCG (RDS CODE GENERATOR PROCESSING CONTROL BLOCK STRUCTURE)

The RDSCG is the control block used by the Code Generator to communicate between modules. The first parameter in the call list to other Code Generator modules contains the pointer to this communication area.

Alphabetic List of Field Names

CGABLK0	8200(2008)	CGDATBL%	8222(201E)	CGRSDCMS	8396(20CC)
CGABLK2	8204(200C)	CGDSAIDX	8212(2014)	CGRSDDMS	8398(20CE)
CGBAPTR	8208(2010)	CGINDX	8224(2020)	CGRSFPMS	8382(20BE)
CGBLK10X	8214(2016)	CGINDX10	8220(201C)	CGRSFWMS	8216(2018)
CGCMPSTK	0(0)	CGIVARIX	8386(20C2)	CGRSHWMS	8218(201A)
CGCSAAA	22(16)	CGI10	8236(202C)	CGRUNTIM	8226(2022)
CGCSCLAS	0(0)	CGI10SAV	8238(202E)	CGSBQOUT	8362(20AA)
CGCSNODE	8(8)	CGJOINMH	8234(202A)	CGSEMSTK	7200(1C20)
CGCSOP	2(2)	CGJ2FLAG	8376(20B8)	CGSERV	8392(20C8)
CGCSORGO	12(0C)	CGLONGF	8388(20C4)	CGSETFNB	8378(20BA)
CGCSQI	4(4)	CGMXFPNR	8384(20C0)	CGSUBQIN	8360(20A8)
CGCSQN	6(6)	CGNWBLK0	8368(20B0)	CGSUBQPT	8240(2030)
CGCSSEMX	20(14)	CGOPPTR1	8364(20AE)	CGSUBQ10	8320(2080)
CGCSVALU	16(10)	CGOUTRBQ	8227(2023)	CGSYMPTR	8228(2024)
CGCSZZZ	14(0D)	CGPJSAVE	8372(20B4)	CGUSQFLG	8380(20BC)





2 Y = Runtime; N = otherwise
3 Y = Outer query in process

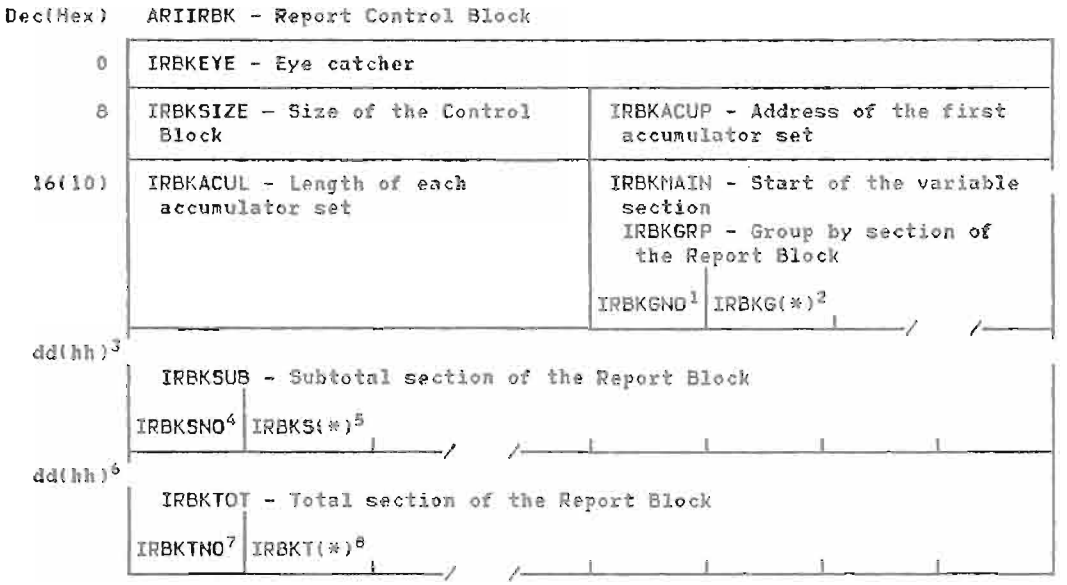
4 Entry point (displacement) of generated routine
5 Entry point of generated routine for join

6 Index to add entry to subquery table.
7 Index to process a subquery table entry.
8 Bit 0: Set function flag.
Bits 1–15: Reserved.
9 Disposition of check byte for testing whether $SELECT is or is not unique.
10 Index of 7FFFFFFF FFFFFFFF.
11 Index of 4E000000 7FFFFFFF.
12 Index of bind table in Block 10.
13 Records whether or not long fields are involved.
14 Index of 0FFFFFFF FFFFFFFF.
15 Index of FFFFFFFF FFFFFFFF.

REPORT CONTROL BLOCK

ARIIRBK is the report control block. It contains subtotal and total accumulators. ARIIRBK stores information from the FORMAT command.

The address of ARIIRBK is contained in the Global Control Block Extension (IGCEXTN), field IGCRPTAP. The Report Control Block exists only when totaling or subtotaling is to be performed for a query.



Notes:

1 IRBKGNO is the number of elements (columns) listed in the FORMAT GROUP statement.

2 IRBKG(*) is the list of the elements (columns) in the FORMAT GROUP statement. The length of IRBKG is (IRBKGNO * 1) bytes. Each byte contains a column number listed in IRBKG.

3 To determine the offset from IRBKMAIN to IRBKSUB, add 1 to the value in IRBKGNO.

4 IRBKSNO is the number of elements (columns) in the FORMAT SUBTOTAL statement.

5 IRBKS(*) is a list of the columns to subtotal. The length of IRBKS is (IRBKSNO * 1) bytes. Each byte contains a column number to subtotal.

6 To determine the offset from IRBKMAIN to IRBKTOT, add 2 to the values in IRBKGNO and IRBKSNO.

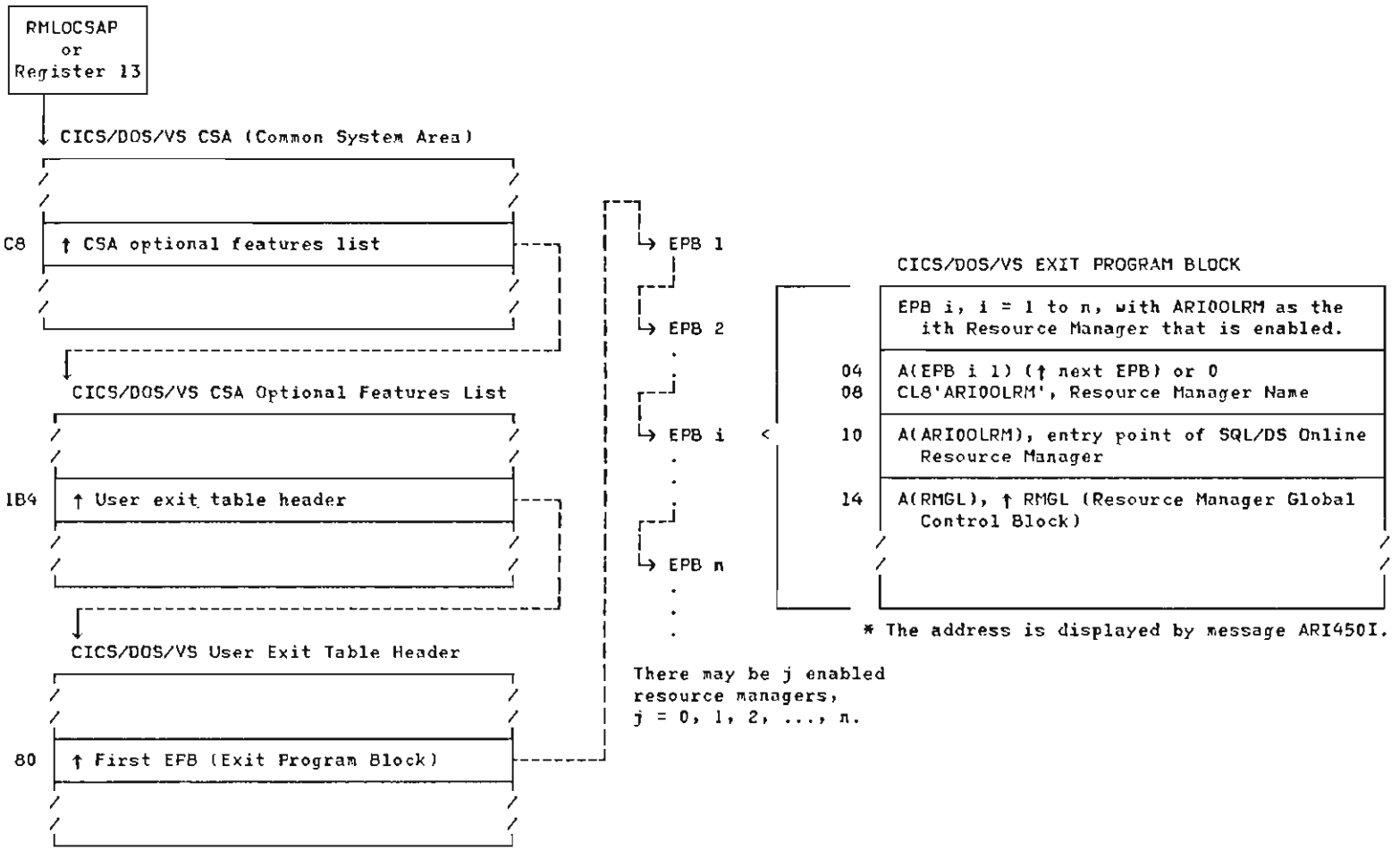
7 IRBKTNO is the number of elements (columns) in the FORMAT TOTAL statement.

8 IRBKT(*) is a list of the columns to total. The length of IRBKT is (IRBKTNO * 1) bytes. Each byte contains a column number to total.

CICS/DOS/VS RELATED CONTROL BLOCKS -- GLOBAL

The following structure shows how to get from the global CICS/VS control block to the RMGL (Online Resource Manager Global Control Block) and resource manager entry point.

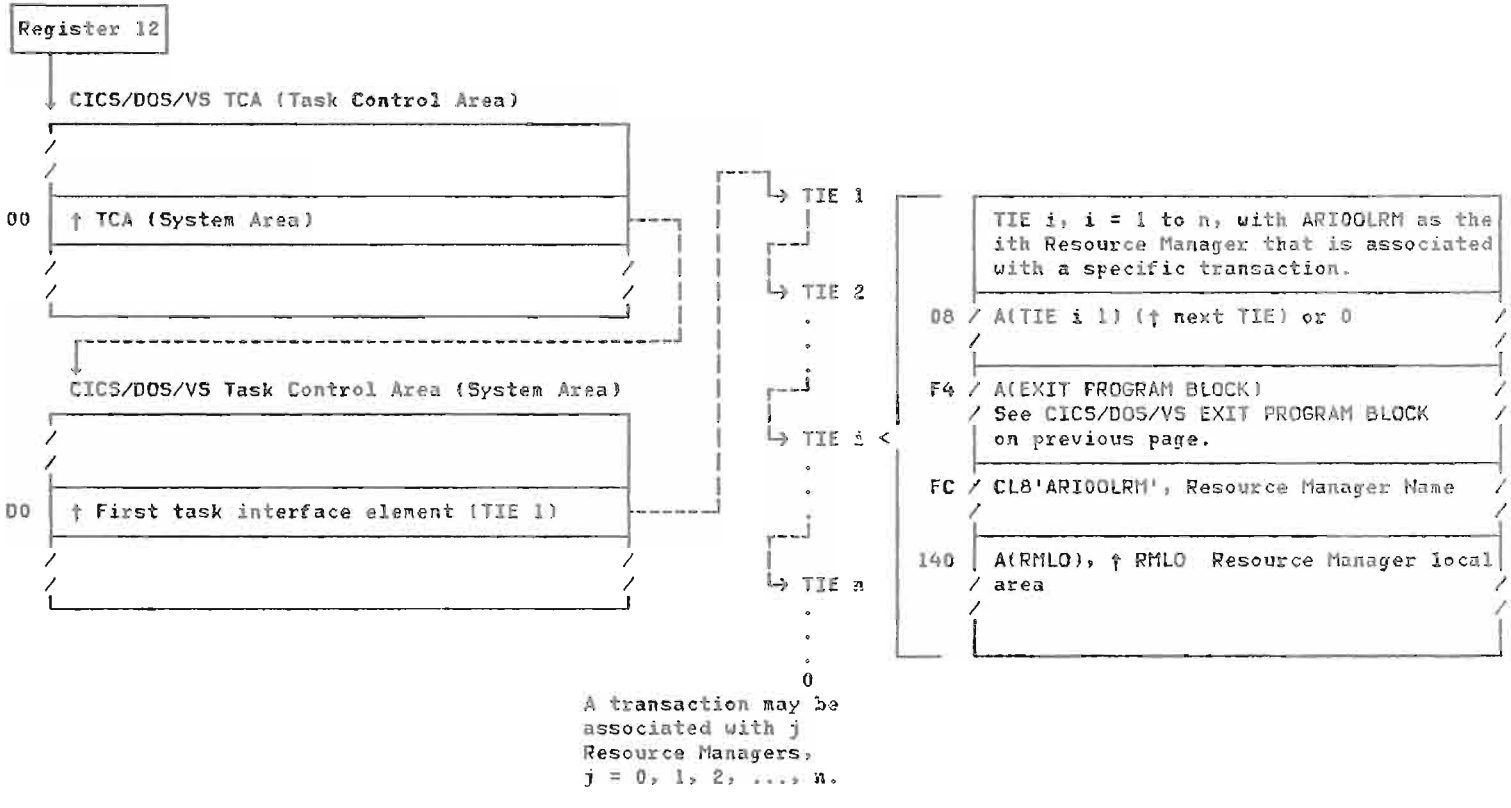
(All offsets are in hexedecimal)



CICS/DOS/VS RELATED CONTROL BLOCKS -- LOCAL

The following structure shows how to get from the CICS/VS transaction local control block to the RMLO (Resource Manager Local Control Block).

(All offsets are in hexedecimal)

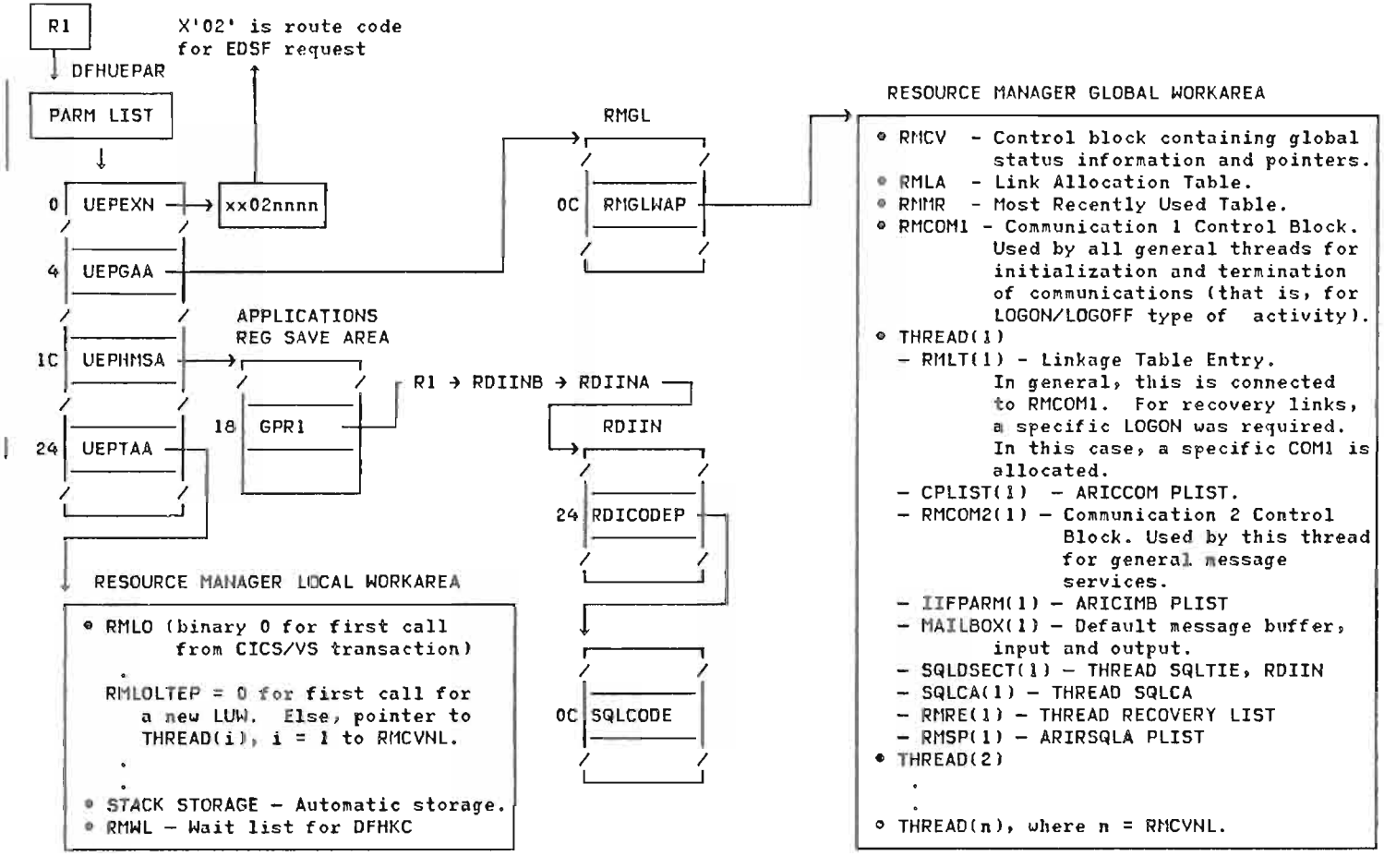


ONLINE RESOURCE MANAGER DATA AREAS

The Online Resource Manager data areas are the RECP, REIB, RMAR, RMCV, RMGL, RMLA, RMLO, RMLT, RMMR, RMRE, RMSP, and RMWL. The next two topics show overviews of the Resource Manager data areas for EDSF (Extended Dynamic Statements Facility) execution and synchronous execution.

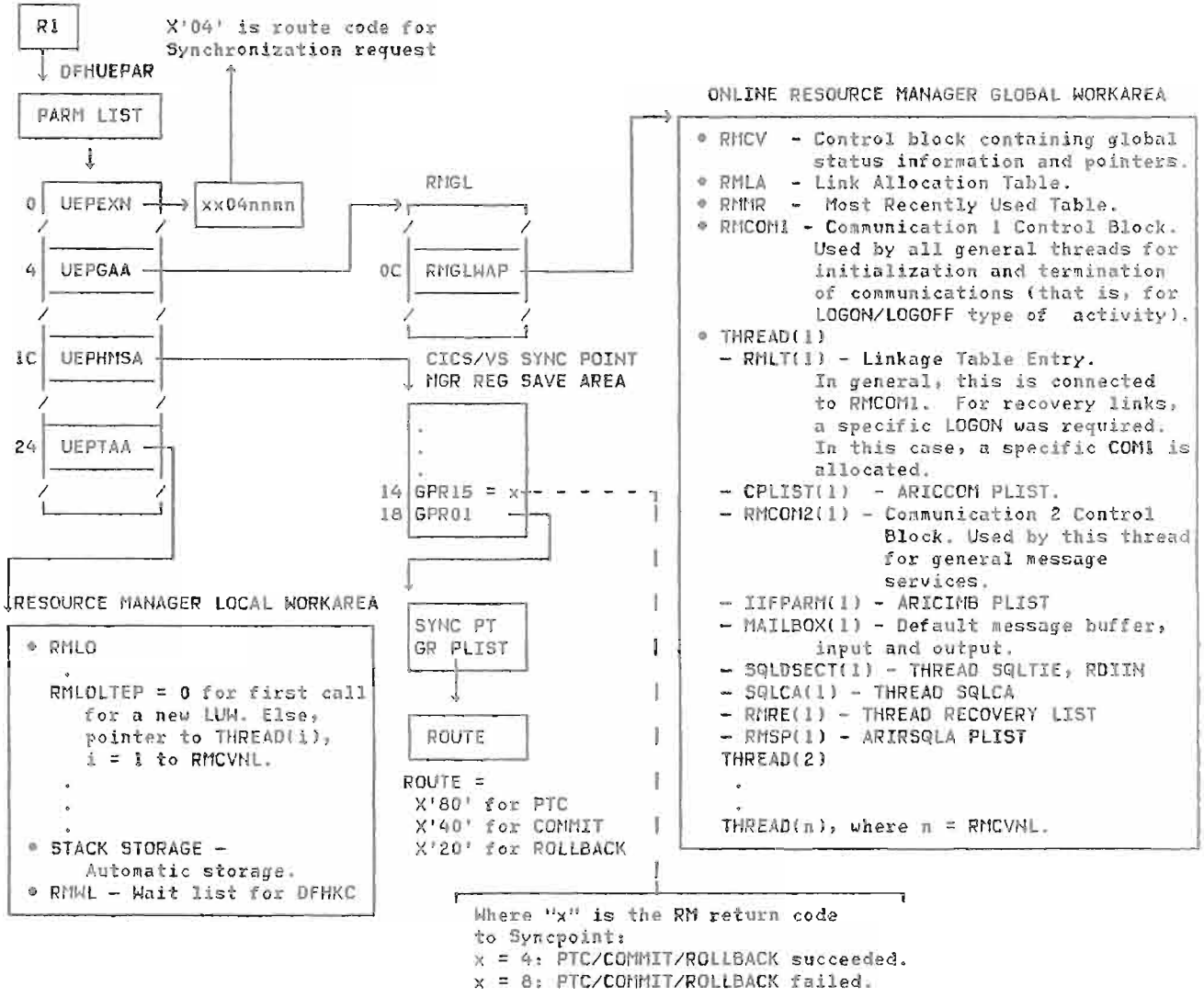
Online Resource Manager - Overview of Data Areas for EDSF Execution

(All offsets are in hexadecimal)



Online Resource Manager — Overview of Data Areas for Synchronization Execution

(Offsets are in hexadecimal)



RECP (ONLINE RESOURCE MANAGER EXEC CICS PLIST)

The RECP is the linkage to ARIRECI, which uses EXEC CICS. For any call to ARICECI, register 1 points to a PLIST, the first word of which points to RECP.

Dec(Hex) RECP

0(0)	RECPFUNC - CICS function being requested [1]		RECPNAME - Name of module	
8(8)	RECPNAME (continued)		RECPSETP - Variable pointer depending on the function	
16(10)	RECPSLEN - Storage length		RECPLENT [2]	RECPHOLD [3] / Reserved 0
24(18)	RECPAPID - CICS application ID			
32(20)	RECPGLEN - Length of global area	RECPTLEN - Length of local area	RECPCSAP - Pointer to CSA	
40(28)	RECPRETC - Return code		RECPSTOR - Return code for operating system storage manager error	
48(30)	RECPTIME - Delay interval '0hhmmss+'X		RECPROW [4]	RECPDEST [5]
56(38)	RECPTXTP - Pointer to message to be sent		RECPQNAM - Name of transient data queue	
64(40)	RECPQNAM (continued) 67(43)			

[1] 'LOAD' = LOAD
'ASGN' = ASSIGN
'RECV' = RECEIVE
'GET' = GETMAIN/GETVIS
'FREE' = FREEMAIN/FREEVIS
'ENAB' = ENABLE (NOSTART)
'DISA' = DISABLE
'EXTR' = EXTRACT
'RELE' = RELEASE
'ADDR' = ADDRESS
'STAR' = ENABLE (START)
'DELY' = DELAY
'SEND' = SEND/WRITEQ TD

[2] Length 5; 2-character length field as requested by CICS/VS.

[3] 'Y' = Invoke HOLD option
'N' = Do not invoke it

[4] Row number of next message to be sent to 3270 display

[5] Additional return code after SEND function:
4 = Message sent to transient queue

RMAR (ONLINE RESOURCE MANAGER ASYNCHRONOUS REQUEST)

This block is used for EXEC RDIIN set or reset user wait exit requests. An "(A)" within the field description indicates that the field is set by the application program. An "(R)" indicates that the Resource Manager sets the field.

For this type of request, the following is applicable in the RDIIN structure:

- RDICTYPE = 165: This is the value reserved for exit calls to the resource manager.
- RDICALL = 'S' for 'SET EXIT' and 'R' for 'RESET EXIT'.
- RDIVPARM = ADDR(RMAR); The RMAR is provided by the application program.
- RDICODEP = ADDR(SQLCA). The following SQLCA values are applicable:
  - SQLCODE=-914 SQLERRD1=0 For valid user requested cancel.
  - SQLCODE=-914 SQLERRD1=4 For user cancel requested with invalid post code.
  - SQLCODE=-824 For invalid set or reset wait exit requests:
    - SQLERRD1=4 When exit already exists.
    - SQLERRD1=8 When RDIVPARM=0.
    - SQLERRD1=12 When exit pointer is 0.
    - SQLERRD1=16 When reset finds no exit to reset.
- RDIERROR should be set to 'B', 'E', or 'W' if the application requires the WHENEVER SQL ERROR OR WARNING process to be active for the specific RDIIN exit request. Otherwise, RDIERROR should be set to a blank ('40'X).
- The remainder of the RDIIN structure is not referenced by the Resource Manager and should be binary zeros.

RESET has two functions. It allows an application to specify a new exit pointer, and it allows for turning off the exit linkage. For each of these functions, RDICALL = 'R'. If RMARXP is not 0, then RMARXP is taken as a new exit pointer. If RMARXP is 0, the Resource Manager pointer to the RMAR is nullified.

When an exit has been established, RMLORMAR contains the pointer to RMAR. When the online support invokes an exit, register 1 points to a PLIST, the first word of which points to RMAR.

Dec(Hex) RMAR

0(0)	RMAREYEC - 'RMAR' eyecatcher (A)	
8(8)	RMARLEN - RMAR length (A)	Reserved (binary zeros) (A)
16(10)	RMARAPPL - Non-architected field reserved for the application (A)	RMARXP - ↑ application exit (A)
24(18)	RMARPC - Post code (A) [1]	RMARXC - Exit code (R) [2]
32(20)	RMARWLP - ↑ RMWL (R)	RMARCNTL - Resource Manager control flags [3]
40(28)	RMARUDAT - User data (terminal id)	
48(30)	Reserved (16 bytes)	

[1] 0: Resource Manager should continue
4: Reserved
8: Resource Manager should cancel
Others: Reserved

[2] 0: WAIT is for a link
a SQL call
Others: Reserved

[3] See RMAR flags (below)

RMAR Flags

Offset	Field Name	Bits	Meanings
36(24)	RMARCNTL		
	RMARWAIT	1...	Tells ARIRSEND to use the wait exit.
		0...	Tells ARISEND to not exit (because call is an EXEC RDIIN PTC or EXEC SQL COMMIT/ROLLBACK WORK (R)
	RMARDSPR	.1..	Tells RMAPI to do a ROLLBACK via DFHSP. Set by ARIRSEN when the attempt to clear did not work because SQL/DS already finished its process.
		..xx xxxx	Reserved. Binary zeros (R)

CANCEL EXIT (VSE ONLY)

The Online Resource manager provides the Set User Wait Exit primarily to allow online applications to perform a cancel function. When it would be appropriate to cancel SQL requests (for example, while waiting for an XPCC link or a SQL request), the resource manager will give control to the user wait exit. This wait exit can then do a CICS wait on ECBs pointed to by the RMWL for normal processing to complete or for the terminal operator to request a cancel. Once posted, the exit can then set the appropriate post-code in the RMARPC field and return to the online resource manager. The resource manager will interrogate the post-code and take the corresponding action.

ARIRCAN Macro (VSE environment)

For the convenience of CICS/VS user transactions to provide a wait exit, a BAL macro is provided to generate the RDIIN set exit call. The format of the macro is:

```
ARIRCAN RMARPTR=addr
```

where:

RMARPTR=addr gives the address of a pointer to the RMAR. This parameter is valid ony in the VSE environment.

Output from this macro with the RMARPTR operand is an RDIIN, type 165, containing the RMARPTR address in field RDIVPARM. See "SET/RESET EXIT CALL" in the RDIIN examples. Also generated is a call to ARIPRDI, which when executed will invoke the online resource manager to perform the Set Exit function.

For the ARIRCAN macro above, the following is required:

1. Invocation must be from a module that has completed assembler preprocessing. This provides for SQLDSECT and SQLCA addressability.
2. As for EXEC SQL, register 13 must point to a standard 72-byte save area, and registers 1, 14, and 15 are modified as a result of the call generated by the macro.

RMCV (ONLINE RESOURCE MANAGER COMMUNICATIONS VECTOR TABLE)

This is a structure containing global information on Online Resource Manager status and anchors to other Resource Manager control blocks.

RMGLWAP in RMGL points to the RMCV.

Dec(Hex) RMCV

0(0)	RMCVEYEC - 'RMCV' eyecatcher	
8(8)	RMCVLEN - RMCV length	RMCVSTAT - RM status flags [1]
16(10)	RMCVSIWA - Size of RM work area	RMCVLATP - ↑ Link Allocation Table
24(18)	RMCVMRUP - ↑ most-recently-used table	RMCVCOMI - ↑ RCOMI [2]
32(20)	RMCVTIP - ↑ first thread in the RM work area	RMCVCXID - Next recovery token [3]
40(28)	RMCVRMRE - ↑ RMRE, which is used for the EXEC RDIIN GET-RECOVERY-LIST call	RMCVDECB - ECB used by the ARIS process when it must wait for unit to finish using the links
48(30)	RMCVWECB - 0 or ↑ chain of one or more local wait elements [4]	RMCVINP - Copy of input parameters
56(38)	RMCVINP (continued)	
64(40)	RMCVCID - CICS application ID. Used for the SQL CONNECT(s) issued by the CIRB process for each link.	
72(48)	RMCVNAME - "ARIOOLRM", the name by which CICS knows the Resource Manager	
80(50)	RMCVAPLN - "ARIxOLRM"[5], the general name by which the Resource Manager is known to the communications facility	
88(58)	RMCVEPP RMCVRMEP - Resource Manager entry point	RMCVCOM - Entry point of system-dependent communication module
96(60)	RMCVSIZE RMCVCSIZ - Size of system-dependent communication control block	RMCVMSIZ - Size of default Mailbox

[1] Byte 1:

RMCVINIT	1...	Initialization in process
RMCVIMED	.1..	Shutdown immediate in process
RMCVQUIE	..1.	Shutdown quiesce in process
RMCVINT	...1	Shutdown activated internally by RM
RMCVPD	.... 1...	Post shutdown process at link unallocate time
RMCVRTRY	.... .1..	Retry termination - CICS disable shows some active units
RMCVFWA	.... ..1.	Shutdown will free RM workarea (not contiguous with RMGL)
RMCVNRTR	1	Terminal operator retry termination - time interval delay occurred before all units completed work

Byte 2:

RMCVIRT	1...	CIRT activated by RM
RMCVEOT	.1..	EOT supported
	.0..	EOT not supported
RMCVRSYN	..1.	All links used by start resynch
	..0.	Resynch not processed
	...x xxxx	Reserved, binary 0's

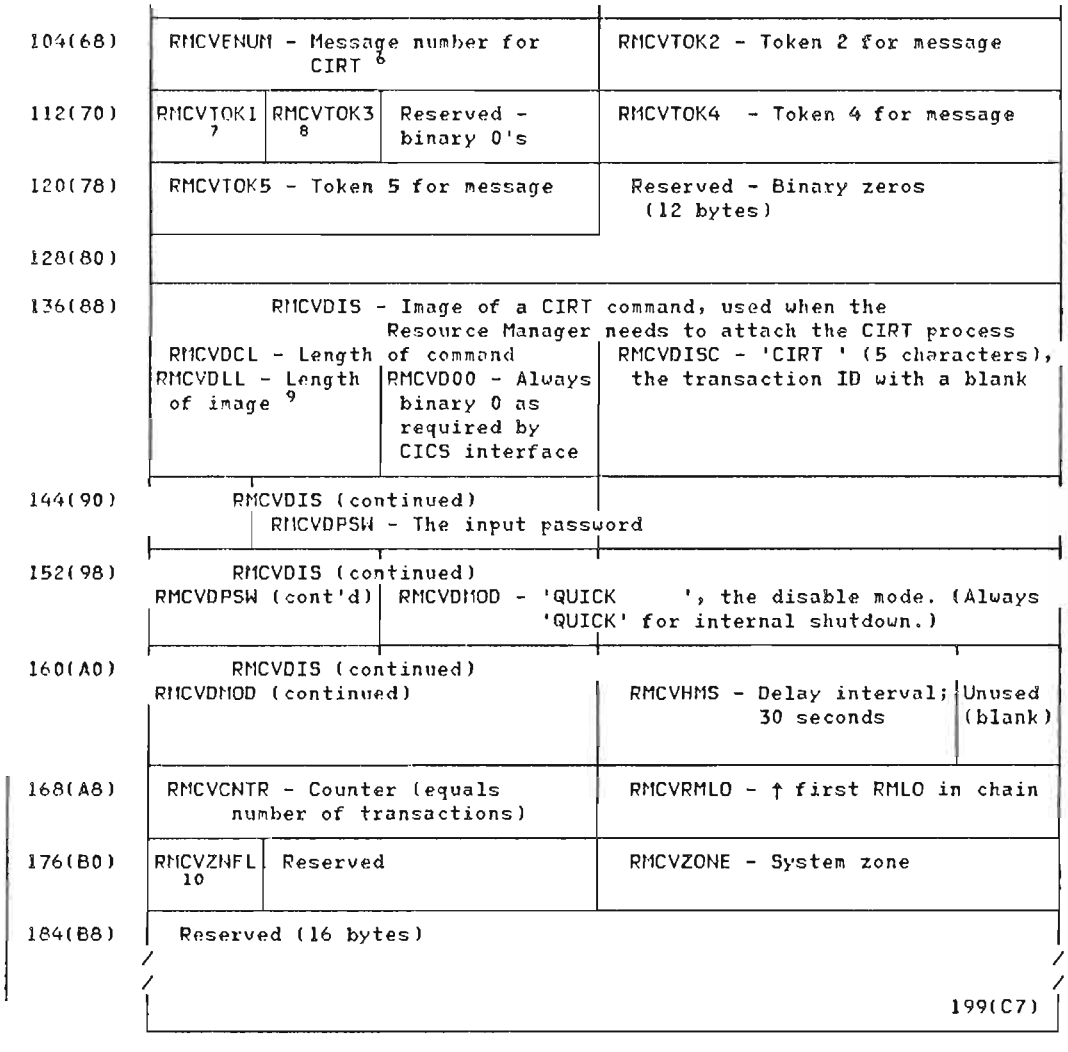
Bytes 3 - 4: Reserved, binary 0's

[2] RCOMI is the system-dependent communication control block used for general logon/logoff of communications

[3] Used only when UEPCXID is binary 0 on initial call from transaction

[4] A local wait element exists for a transaction waiting on a link

[5] x is the input RMID to CIRB



[6] When online support must abnormally terminate, this message and tokens that follow indicate reason for termination.

[7] Token 1 for message

[8] Token 3 for message

[9] Includes image following plus this four-character field.

[10] System zone flag
'Y' = no need to call SYSDE module (get zone)

RMGL (ONLINE RESOURCE MANAGER GLOBAL AREA)

This structure anchors the Online Resource Manager work area. It is allocated by CICS when a CICS ENABLE command is issued. The pointer to RMGL is made available to the Resource Manager on every RMI call.

The field UEPGAL in DFHUEPAR points to RMGL. The pointer to RMGL may also be obtained via EXEC CICS EXTRACT.

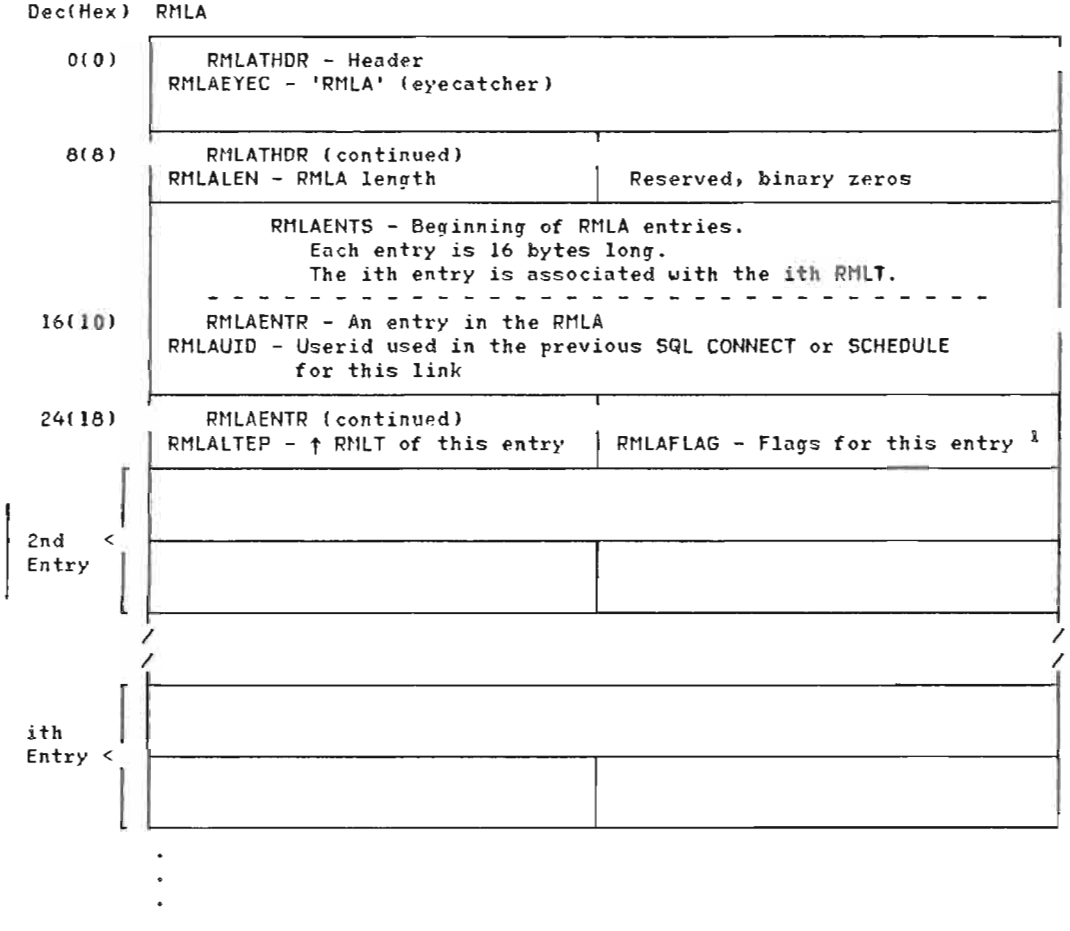
Dec(Hex) RMGL

0(0)	RMGLEYEC - 'RMGLO' (eyecatcher)	
8(8)	RMCVLEN - Length of this area	RMGLWAP - † Resource Manager work area, which contains RMCV, RMLA, etc

RMLA (ONLINE RESOURCE MANAGER LINK ALLOCATION TABLE)

This structure contains allocation information on each link managed by the Online Resource Manager. The RMLA is made up a header and a number of entries. The ith entry is associated with the ith RMLT.

RMCVLATP in RMCV points to the RMLA.



1 In RMLAFLAG:
- Bit 0 (RMLAFREE)
  - 0 = A free link
  - 1 = Link is allocated
- Bit 1 (RMLAWAIT)
  - 0 = A link is ready
  - 1 = Wait for communication to complete before using the link (wait on SENDECB in the CMPLIST)
- Bit 2 (RMLANR)
  - 0 = Restart resynch complete
  - 1 = Restart resynch not complete
- Bits 3 - 31
  - Reserved and binary zeros

RMLO (ONLINE RESOURCE MANAGER LOCAL AREA)

This structure applies to all Resource Managers. It exists for the life of a task. The task could be a CICS transaction or an operating system task. The structure contains control information and stack storage.

In general, during execution register 10 points to the RMLO. The field UEPTAL in DFHUEPAR points to the RMLO for any RMI call.

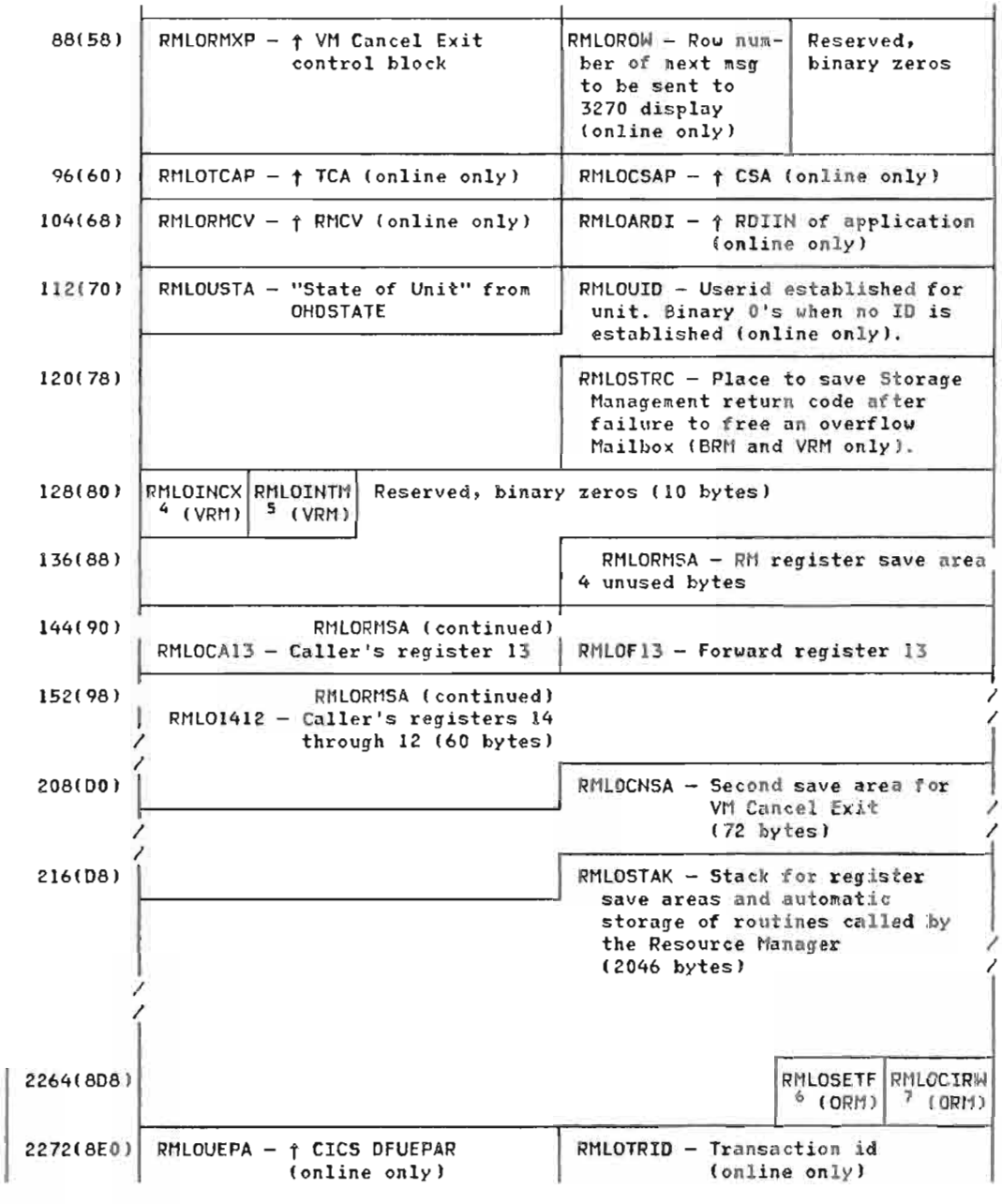
Dec(Hex) RMLO

0(0)	RMLOEYEC – 'RMLO' eyecatcher		
8(8)	RMLOLEN – RMLO length	RMLOSTK – ↑ beginning of stack	
16(10)	RMLODRSP – ↑ dummy YRSSCVT [1]	RMLONXTP – ↑ next RMLO in a partition or virtual machine (BRM and VRM only).	
24(18)	RMLOLTEP – ↑ RMLT. 0 on first call.	RMLOTEI – Index to RMLT. 0 on first call.	RMLOSTAT – Status flags [2]
32(20)	RMLOPATC – ↑ RMLO patch CSECT	RMLOSTRL – Length of storage for batch RMLO, RMLT, ...	
40(28)	RMLOIDS RMLOTID – Task ID (BRM and VRM only)	RMLODID – SQL/DS ID. Must be binary zeros.	
48(30)	RMLOWAIT [3] RMLOWCH – ↑ next local wait element or 0 (ORM only)	RMLOWECB – The wait ECB posted by Link Unallocate when the local wait element is queued off the RMCV (ORM only)	
56(38)	RMLOWAIT (continued) RMLOWLP – ↑ WAITLIST (RMWL) (ORM only)	RMLOCECB – CANCEL ECB posted by RM or user CANCEL EXIT (VRM only)	
64(40)	RMLOCTRS – CMF counters (ORM only) RMLONLA – Number of link allocates	RMLONLW – Number of link waits	
72(48)	RMLOCTRS (continued) RMLONDM – Number of SQL/DS requests	RMLONDE – Number of errors	
80(50)	RMLOEIBP – ↑ EIB (Online RM only)	RMLORMAR – 0 when Set Exit is not active for the asynchronous request linkage. Otherwise, ↑ RMAR established by the application. (ORM only.)	

[1] The dummy YRSSCVT starts at RMLONXTP. The pointer to the RM patch CSECT (ARIRPAT) is located at offset 12(0C) from RMLONXTP.

[2] RMLOSTAT:
- Bit 0 (RMLORMG):
  - 1 = Resource Manager generated external id.
- Bit 1 (RMLOOLRM):
  - 1 = Online RMLO
- Bit 2 (RMLOWAI):
  - 1 = Online link wait
- Bit 3 (RMLOCNSL):
  - 1 = Online terminal is also system console
- Bit 4 (RMLOCNSY):
  - 1 = Commit or rollback is being canceled
- Bit 5 (RMLOPRMX) (VRM only):
  - 1 = RM is to return ↑ to RMX in register 15
- Bit 6 (RMLOFRST) (BRM and VRM only):
  - 1 = This is first pass through RM
- Bits 7 - 15: Reserved and binary zeros

[3] Local wait element used to wait on links. The links may be in use by other units, or they may be busy with a communications post pending after a cancel.

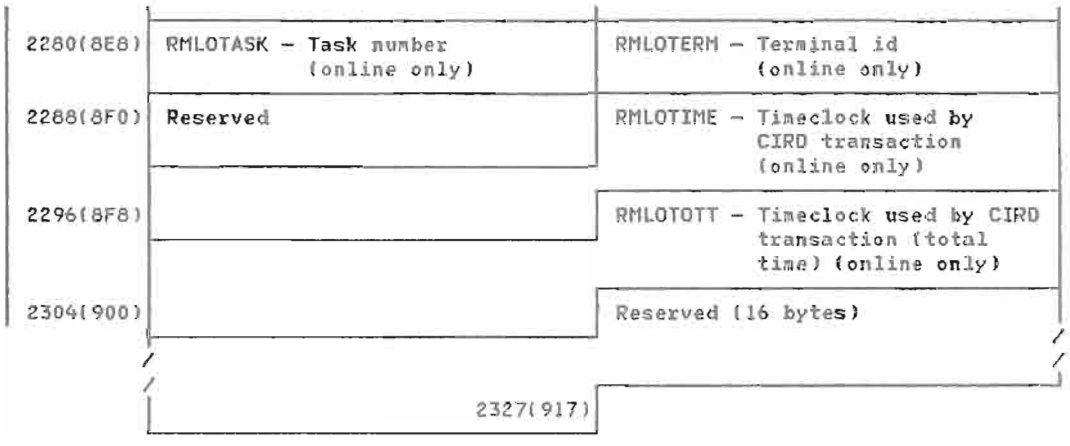


[4] Bit 0: 1 = VRM is in Cancel Exit routine
Bits 1 – 7: Unused (set to 1 by Test and Set)

[5] Bit 0: 1 = VRM is in Termination Exit routine. A NUCXDROP within the exit causes recursion into the exit.
Bits 1 – 7: Unused (Set to 1 by Test and Set)

[6] Bit 0: 1 = Do not return to RMAPI
Bits 1 - 7: Reserved

[7] Flag for RMLO wait chain. Set only by CIRD.


2280(8E8)
RMLOTASK - Task number (online only)
RMLOTERM - Terminal id (online only)
2288(8F0)
Reserved
RMLOTIME - Timeclock used by CIRD transaction (online only)
2296(8F8)
RMLOTOTT - Timeclock used by CIRD transaction (total time) (online only)
2304(900)
Reserved (16 bytes)
2327(917)

RMLT (RESOURCE MANAGER LINK TABLE ENTRY)

This structure applies to all Resource Managers. It contains link control information. It gives the status of links and anchors communication control blocks and buffers.

RMLT pointers are maintained in the RMLAENTS entries of RMLA. Each RMLAENTS contains an RMLALTEP, which points to an RMLT. Once an RMLT is allocated, field RMLOLTEP of the RMLO points to the RMLT.

Dec(Hex) RMLT

0(0)	RMLTEYEC - 'RMLT' eyecatcher	
8(8)	RMLTLEN - RMLT length	RMLTSTAT - Link status bytes [1]
16(10)	RMLTCPLP - ↑ Communications Manager PLIST	RMLTICBP - [2]
24(18)	RMLTCCBP - ↑ CONNECT/SEND system-dependent communications control block	RMLTMLST - ↑ Mailbox PLIST
32(20)	RMLTSQLD - ↑ SQLDESCT used for RM initialized SQL requests (online only)	RMLTSQLC - ↑ SQLCA used for RM initialized SQL requests (online only)
40(28)	RMLTRMRE - ↑ RMRE used to prepare to COMMIT and to get the recovery list (online only)	RMLTSPP - ↑ RMSP for linkage to ARIRSQLA (online only)
48(30)	RMLTRDII - ↑ RDIIN. This points to the application's RDIIN (online only)	RMLTRMI - ↑ RMI (DFHUEXIT pointer) (online only)
56(38)	RMLTTOK - Recovery token (saved for restart resynch) (online only)	
64(40)	Reserved - 8 bytes binary zeros	71(47)

[1] See flags on page 195.

[2] Pointer to INIT/TERM system-dependent communications control blocks. (In general, for Online Resource Manager, this points to the control block in the RM work area and is equal to RMCVCOM1. For links that are used for recovery, this points to a communications control block that is unique for the recovery link.)

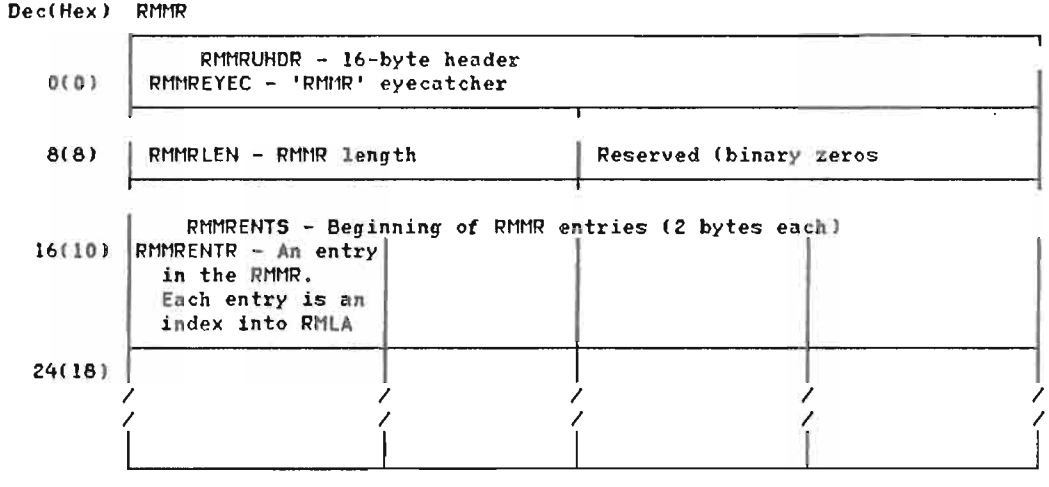
RMLT Flags

Offset	Field Name	Bits	Meanings
12(0C)	RMLTSTAT		
Byte 0	RMLTIDEN	1...	Communication logon done.
		0...	Communication logon required.
	RMLTCON	.1..	Communication connect to SQL/DS done.
		.0..	Communication connect to SQL/DS required.
	RMLTUS	..1.	The RMAPI issued an 'unsolicited' DFHSP TYPE= ROLLBACK because of an unsolicited ROLLBACK.
		..0.	No 'unsolicited' DFHSP (online only).
	RMLTSCH	...1	SQL SCHEDULE done (online only).
		...0	SQL SCHEDULE required (online only).
	RMLTINRM	.... 1...	Unit is in Resource Manager (ORM and VRM).
		.... 0...	Unit not in Resource Manager (ORM and VRM).
	RMLTRMSP	.... ..1.	Resource Manager requested SYNCPOINT because of EDSF request for COMMIT or ROLLBACK work (online only).
		.... ..0.	User/CICS requested SYNCPOINT (online only).
	RMLTSPR	1	TYPE=ROLLBACK was specified by the Resource Manager DFHSP request (online only).
		0	TYPE=USER (COMMIT) was specified by the Resource Manager DFHSP (online only).
		.... .x..	Reserved.
Byte 1	RMLTABEN	1...	Resource Manager issued the DFHPC request to abend the unit (online only)
		0...	No Resource Manager-generated abend for this unit (online only).
		.xxx xxxx	Reserved.
Bytes 2 and 3			Reserved.

RMMR (ONLINE RESOURCE MANAGER MOST-RECENTLY-USED TABLE)

The RMMR contains entries of the most-recently-used links. It is used to select a free link, which will minimize paging. The last entry contains the index into the RMLA of the link that was most-recently freed; the second-from-last entry is the index of the second-most-recently freed; and so on. The first entry is the index of the link that has not been used for the greatest period of time.

The RMMR is pointed to by the RMCVMRUP field of the RMCV.



RMRE (ONLINE RESOURCE MANAGER RECOVERY LIST)

The RMRE structure is used for the Two-Phase Syncpoint linkage to SQL/DS. It is used to prepare a unit for COMMIT WORK or ROLLBACK WORK and to get the list of units that are prepared for a specific Resource Manager. This is the GET INDOUBT LIST request. In this case, the status of the SQL/DS log for the specific Resource Manager is set.

The RMRE has a 4-byte prefix. The prefix is used only by the Input Mailbox Build routine (ARICIMB).

The pointer to an RMRE is maintained in RMLTRMRE of the RMLT. For a GET RECOVERY LIST call and a PREPARE TO COMMIT call, the RDIVPARM field of the RDIIN points to the RMRE.

Dec(Hex) RMRE

0(0)	RMREPREF - 4-byte prefix RMREMBL [1] | Reserved (binary 0's)	RMREEYEC - 'RMRE' eyecatcher
8(8)	RMREEYEC (continued)	RMRELEN - RMRE length
16(10)	RMRECOO - Coordinator Id. The name by which CICS is known to a VTAM net, APPLID.	
24(18)	RMRERMID - ID of the Resource Manager RMREARI - 'ARI' constant | RMRERMID [2]	RMRECOLD - SQL/DS cold log time stamp
32(20)	RMRECOLD (continued)	RMRENTOK - Number of recovery tokens that could fit in this list
40(28)	RMRETOUT - Number of recovery tokens that were found on a request for recovery list	RMREENTS - RMRE entries RMRETOK - The recovery token RMRETOK1 - Word 1
48(30)	REMRETOK (continued) RMRETOK2 - Word 2	RMREUID - SQL userid
56(38)	REMREUID (continued)	RMRETOI - CICS terminal operator ID (if any)
64(40)	RMRETID - CICS terminal ID (if any)	REMRECTI - CICS transaction ID (character format)
72(48)	RMRECTIB - CIS transaction IO - (unique binary number in a domain of CICS transactions)	RMREAGID - Agent identifier in SQL/DS
80(50)	Reserved	

[1] Length for building the RMRE as an element in the Mailbox. Used by ARICIMB. Has value of:
LENGTH(RMRE)
LENGTH(RMREMBL)
The value should be 2 less than the value in RMRELEN.

[2] Binary value of the RM input parameter - RMID

RMSP (LINKAGE TO THE ONLINE RESOURCE MANAGER SQL LINKAGE MODULE)

RMSP is the parameter list for an Online Resource Manager call to ARIRSQLA.

The pointer to an RMSP is maintained in the RMLTSPP field of the RMLT. For a call to ARIRSQLA, register 1 points to a PLIST, the first word of which points to RMSP.

Dec(Hex) RMSP

0(0)	RMSPFUNC - Function code[1]	RMSPUID - Userid
8(8)	RMSPUID (continued)	RMSPPSWD - Password
16(10)	RMSPPSWD (continued)	RMSPRDIL - Length of RDIIN
24(18)	RMSPPRLP - ↑ Recovery List	RMSPRSD - ↑ SQLDSECT
32(20)	RMSPRSC - ↑ SQLCA	

[1] Function codes:
1 - EXEC SQL CONNECT
2 - RDI SCHEDULE
3 - EXEC SQL COMMIT WORK
4 - EXEC SQL ROLLBACK WORK
5 - RDI PREPARE TO COMMIT
6 - RDI GET RECOVERY LIST

RMWL (ONLINE RESOURCE MANAGER WAIT LIST)

RMWL is the Online Resource Manager wait list for the CICS/VS interface DFHKC TYPE=WAIT,DCI=LIST. When the Resource Manager does the wait on this list, it gives ADDR(RMWLENTS) to the host as the pointer to the wait list. When a user exit waits on the list, it gives ADDR(RMWLUEP) to the host.

The pointer to RMWL is maintained in the RMLOWLP field of the RMLO. For a call to a user wait exit, the RMARWLP field of RMAR points to the RMWL.

Dec(Hex) RMWL

<table>
<tr><td>0(0)</td><td colspan="2">RMWLEYEC - "RMWL" eyecatcher</td></tr>
<tr><td>8(8)</td><td>RMWLLEN - RMWL length</td><td>RMWLNRME - Number of Resource Manager elements in this list. The number includes the delimiter element (word) 'FFFFFFFF'X</td></tr>
<tr><td>16(10)</td><td>RMWLUEP - Slot for user exit to place a pointer to its CICS Event Control Area (ECA)</td><td>RMWLENTS - Entries for Resource Manager
RMWLENTR(*) - ↑ ECA or the delimiter word [1]</td></tr>
</table>

[1] The number of these 4-byte entries is given in RMWLNRME.

RMXC (RESOURCE MANAGER CANCEL EXIT CONTROL)

This control block is used for controlling the VM/SP resource manager cancel support. It is used by the immediate command exit supplied with the VM resource manager as well as by the resource manager cancel support. An '(A)' within the field description indicates that the filed is set by the application. An '(R)' indicates that the VM resource manager sets the field. Storage for the RMXC is allocated by the VM resource manager. Application addressability to the RMXC is provided through the ARIRCAN macro.

Dec(Hex) RMXC

0(0)	RMXCEYEC - "RMXC" eyecatcher	
8(8)	RMXCLENF - RMXC length	Reserved
16(10)	RMXCAPPL - For application use	RMXCCXIT - Optional pointer to an application cancel exit routine (binary zero if no exit defined)
24(18)	RMXCCONT [1] | Reserved	
32(20)	Reserved	RMXCXC - Exit code [2]
40(28)	Reserved (binary seros)	
48(30)	RMXCECBP - † Cancel ECB	RMXCPC - Post mask, SQL/DS cancel
56(38)	RMXCPHX - Post mask, halt execution	Reserved (binary zeros)
64(40)	Reserved (binary zeros)	Reserved (binary zeros)
72(48)	Reserved (binary zeros)	

[1] Application cancel exit stop indicator:
'Y' to continue cancel
'N' to stop cancel

[2] Exit codes are:
RMXCWLNK: 0 - RM Link wait
RMXCWSQL: 4 - RM SQL wait
RMXCWSC: 8 - RM wait for SQL COMMIT WORK
RMXCWSRB: 12 - RM wait for SQL ROLLBACK WORK
RMXCWNO: -1 - RM Not waiting
RMXCCAN: -2 - RM not waiting and cancel has been requested

The VM resource manager provides a cancel support for applications by using the VM/SP immediate command extension support. During initialization in th MVMM environment, the resource manager establishes an immediate command of SQLHX. When this command is entered, if the resource manager is waiting for a reply from the SQL/DS virtual machine, the SQL command that is currently being processed will be canceled, and the resource manager will return to the application with a SQLCODE of -914.

If the SQLHX command is entered and the resource manager is not waiting for a reply from SQL/DS, the next SQL command that is issued will be canceled with a return code of -914.

Cancel support is intended for interactive applications such as ISQL, where the terminal operator is obviously aware of what is being canceled. Applications that perform preplanned SQL queries in a batch type environment might be adversely affected by the arbitrary issuing of the SQLHX command. Also, applications may wish to have the cancel support invoked by a command other than 'SQLHX'.

In order to provide flexibility in the VM resource manager cancel support, a BAL macro is provided to allow applications to modify the basic cancel support in the following ways:

1. Create additional immediate command names with which to invoke the cancel support command exit.
2. Remove the SQLHX immediate command exit, which will effectively discontinue VM resource manager cancel support unless the application chooses to provide its own cancel exit.

The format of the ARIRCAN macro is:

```
          { CMDNAME=pointer-address }
          { TYPE=USER               }
```

where

CMDNAME=pointer-address is the address to an 8-byte field containing an alternate command name to be used to invoke the VM resource manager immediate command exit. This parameter is valid only in the VM/SP environment. The command name 'SQLHX' remains in effect. The additional command will function as an alias to invoke the immediate command exit.

TYPE=USER specifies that the user will handle any cancel exit function. It causes the VM resource manager to remove the immediate command exit established at initialization. This parameter is valid only in a VM/SP environment.

Output from this macro with the CMDNAME or TYPE operands is an RDIIN, type 166. For the CMDNAME operand, the command name pointer is placed in the RDIVPARM field. Also generated is a call to ARIPRDI which, when executed, will invoke the VM resource manager to either establish an additional immediate command name or remove the SQLHX immediate exit, depending on the parameters specified in the ARIRCAN macro.

On successful return to the application, register 15 points to the RMXC control block. The following is required for the ARIRCAN macro:

1. Invocation must be from a module that has completed assembler preprocessing. This provides for SQLDSECT and SQLCA addressability.

2. As for EXEC SQL, register 13 must point to a standard 72-byte save area, and registers 1, 14, and 15 are modified as a result of the call generated by the macro.

The cancel function is supported by two parts of the VM resource manager:

1. The immediate command process (modules ARIRCL1C and ARIRCL2C).

2. The mainline VM resource manager process that gets notified (posted) by the immediate command process when a cancel is being requested.

When the resource manager calls ARICCOM to send a message across the path to SQL/DS, a CMS WAITECB macro is issued for two event control blocks (ECBs). One ECB will be posted when the communication is complete. The other will be posted by the immediate command process if it is invoked by the SQLHX command (or any other user-defined command). If the cancel ECB is posted, the resource manager mainline (ARIRVRM) will terminate the SQL command by calling ARICCOM to do IUCV SEVER. This SEVER will cause SQL/DS to rollback the processing that was taking place for the SQL command. The resource manager will then return to the application with a -914 SQLCODE.

The resource manager provides for a user exit to be called from the mainline cancel process. If the RMXCCXIT field contains a non-zero value, it is assumed that a user exit routine exists. The resource manager branches to this exit prior to issuing the IUCV SEVER to cause the cancel. At this point the user exit can discontinue the cancel request by setting the RMXCCONT field to 'N'.

Register contents on entry to the user exit are:

```
Register  1 - pointer to RMXC control block
Register 13 - Pointer to 72-byte save area
Register 14 - Return address
Register 15 - Entry point of user exit.
```

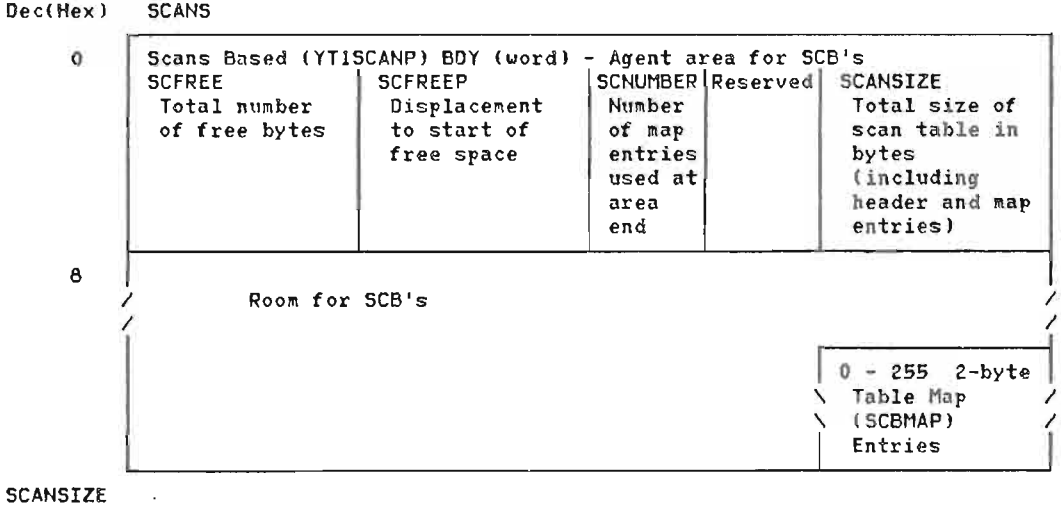
It is assumed that registers will be restored upon return from the exit.

SCAN TABLE

There is one Scan Table for each SQL/DS Agent. The Scan Table is allocated during SQL/DS initialization. The size (stored in SCANSIZE) of a Scan Table is equal to 50 times the NCSCANS SQL/DS initialization parameter plus 8 (for scans header). YTABLE1 field YT1SCANP points to the Agent Scan Table.

The Scan Table is used for allocation and maintenance of Scan Control Blocks (SCBs). When a scan is closed (and deallocated) the storage is available for allocation of new SCB's. The Scan Table is compacted whenever there is not sufficient contiguous free space to allocate a new SCB. The format of the Scan Table consists of a header (SCANS) at the beginning of the area and table map entries (SCBMAP) at the end, with all remaining space available for SCB's.

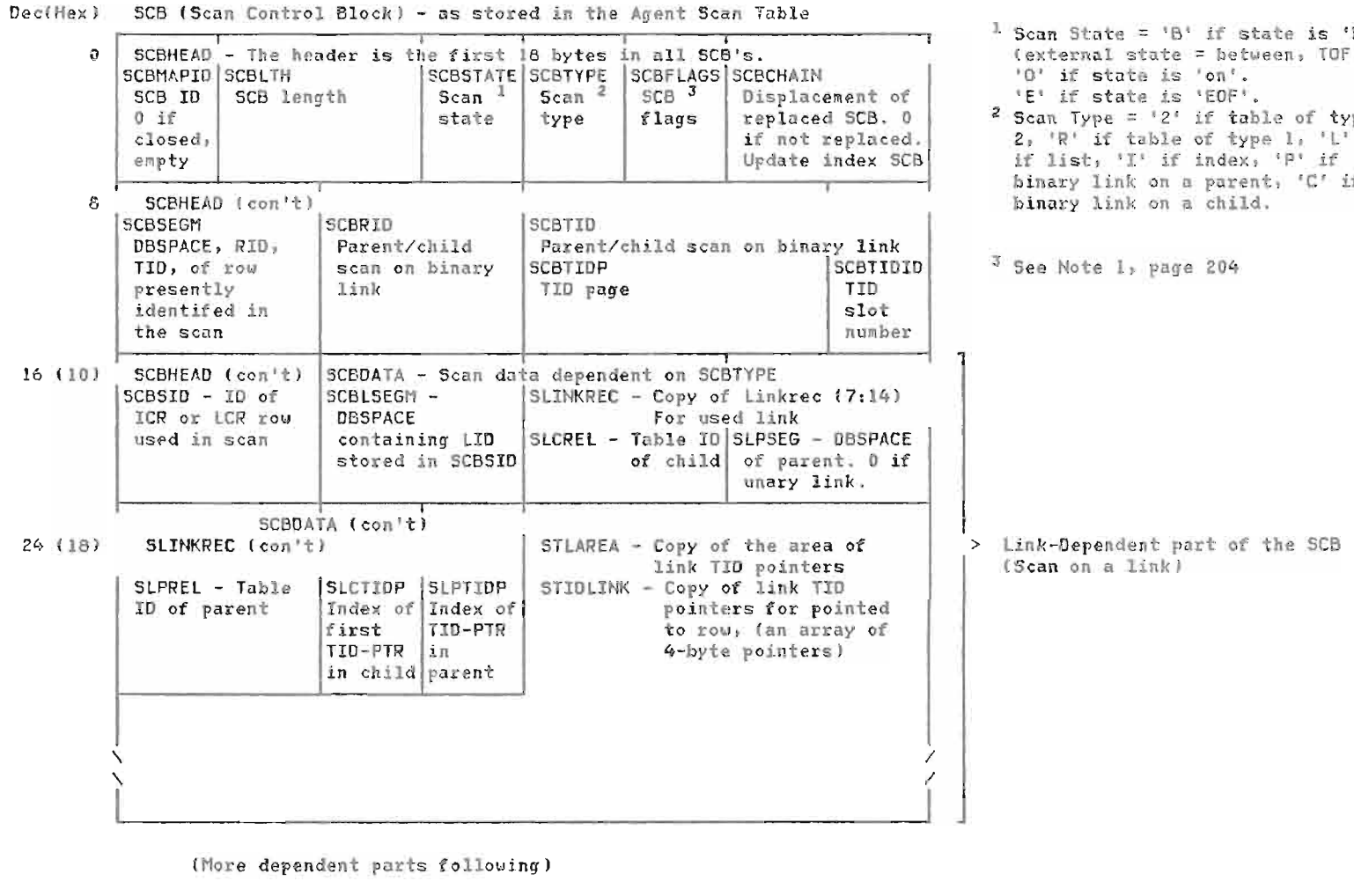
There are SCNUMBER of SCBMAP entries (one per scan) at the end of the of the Scan Table. The SCBMAP entry contains 0 if the corresponding SCB is closed, 1 if the SCB is open for an empty scan-set, otherwise it contains the SCB displacement within the Scan Table. A map entry is allocated each time an SCB is allocated and points to the location of the SCB at SCANS + (SCANSIZE - (2 times I)) within the table. The ordinal of the map is the Scan ID. Where I is the Scan ID of an SCB, the SCBMAP entry is located at SCANS + (SCANSIZE - (2 times I)). Map entries are allocated sequentially and are reuseable.



SCB (SCAN CONTROL BLOCK)

The DSC allocates a Scan Table for each SQL/DS Agent.
Within the Scan Table, Scan Control Blocks are allocated.

Each allocated (and open) SCB has a Scan-ID which is the descending slot number of the SCBMAP entry.



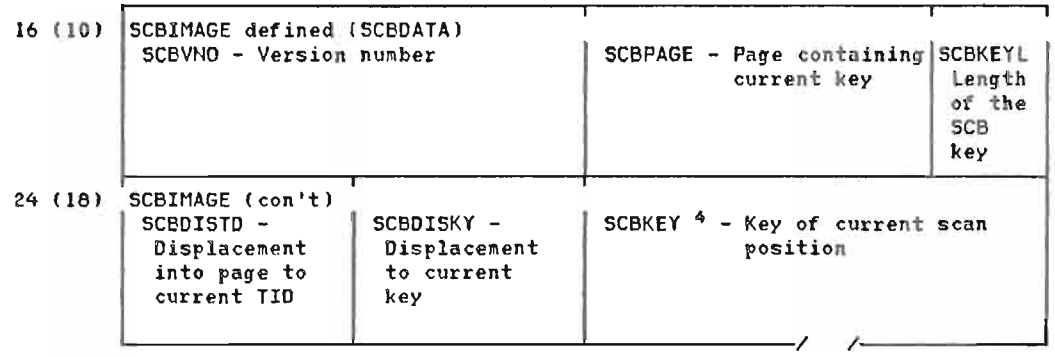
[1] Scan State = 'B' if state is 'before' (external state = between, TOF). 'O' if state is 'on'. 'E' if state is 'EOF'.

[2] Scan Type = '2' if table of type 2, 'R' if table of type 1, 'L' if list, 'I' if index, 'P' if binary link on a parent, 'C' if binary link on a child.

[3] See Note 1, page 204

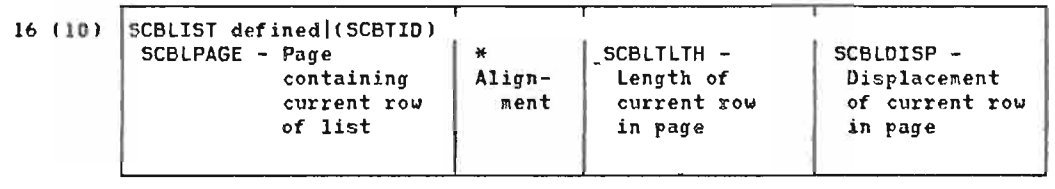
> Link-Dependent part of the SCB (Scan on a link)

Index Dependent Part of the SCB (Scan on an index)



[4] The number of bytes depends on the key length.

List Dependent Part of the SCB (Scan on a list)

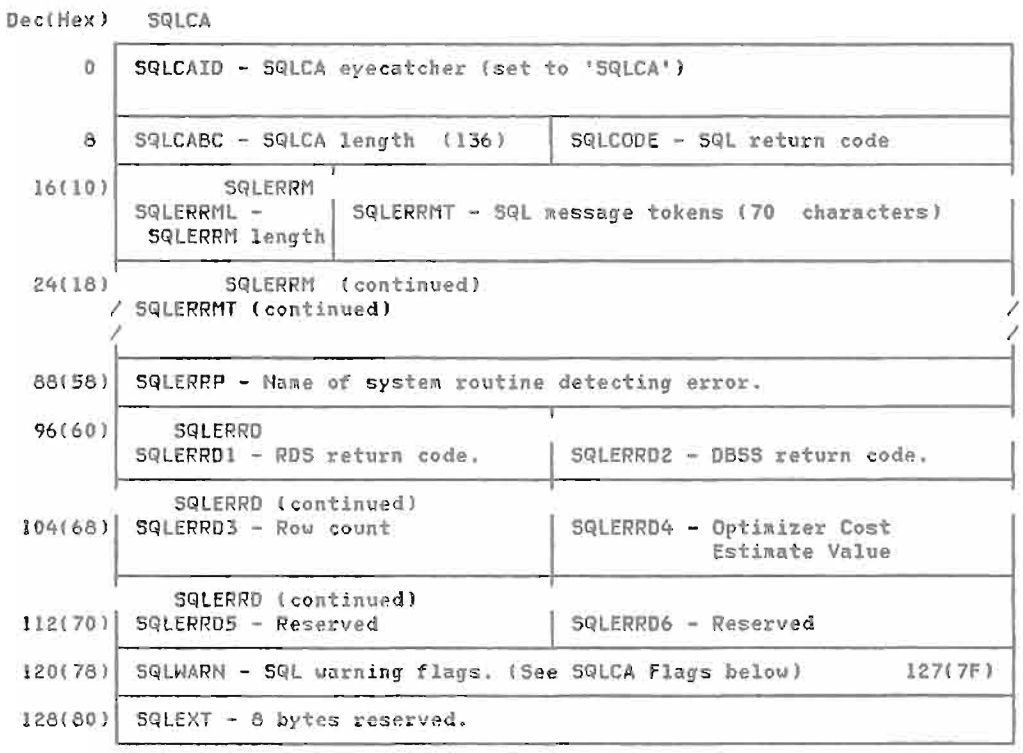


Note 1:

Offset	Field Name	Bits	Meanings
5	SCBFLAGS		
		0...	SCB not dirty
		.0..	Physical information is OK
		..0.	Public or private DBSPACE (SCB to be saved at SAVE)
		...1	DBSPACE requires TID locking (from YTLOCK in YTABLE1)
		.... 1...	No scan lock required
		.... .1..	RELSTATE>=SMODE
		.... ..1.	Index scan with key value locking
		1	Scan is frozen. This is used for COMMIT.

SQLCA

The current SQLCA is used to return error diagnostics to the using program. It is pointed to by field RDSQLCA of the RDAREA.

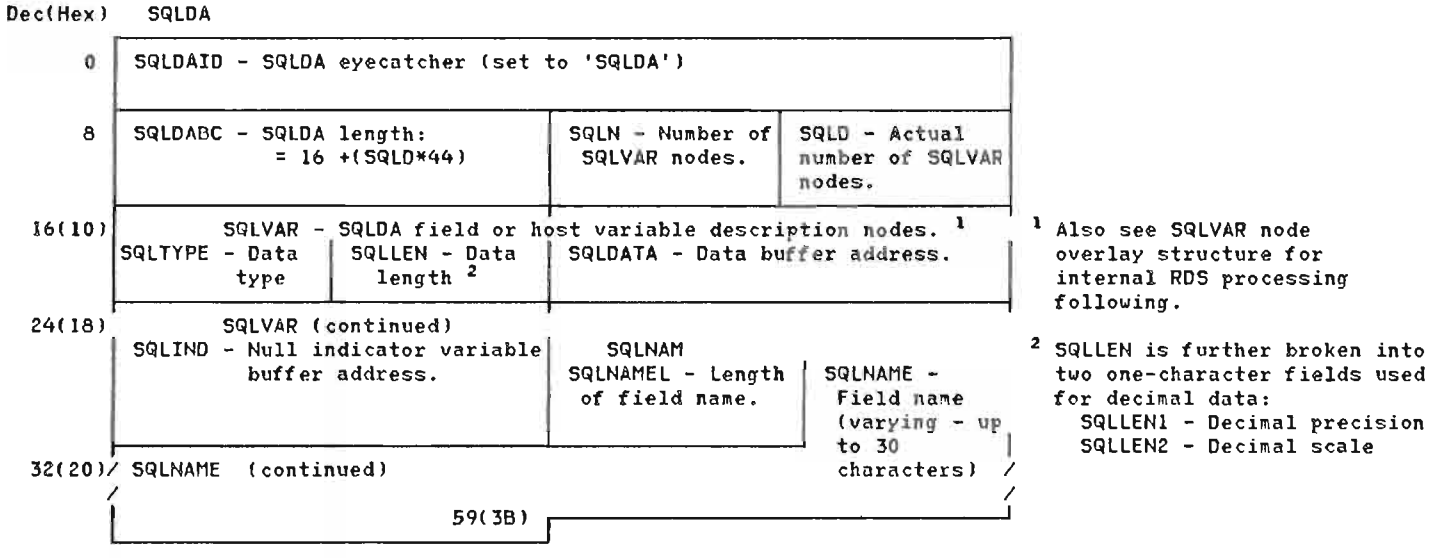


SQLCA Flags

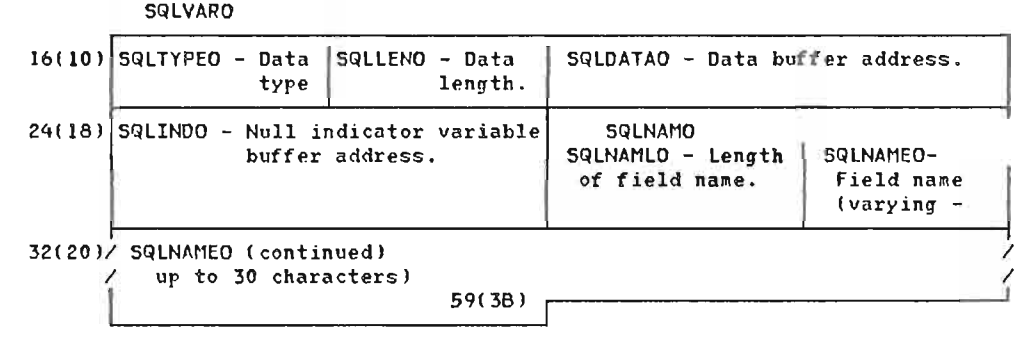
Offset	Field Name	Meanings
120(78)	SQLWARN	
120(78)	SQLWARN0	Warning indicator.
121(79)	SQLWARN1	Data item truncation.
122(7A)	SQLWARN2	Nulls ignored.
123(7B)	SQLWARN3	Items mismatch.
124(7C)	SQLWARN4	Null WHERE clause.
125(7D)	SQLWARN5	Reserved.
126(7E)	SQLWARN6	No logical unit of work in progress.
127(7F)	SQLWARN7	Reserved.

SQLDA

The SQLDA is used to communicate descriptive information about host variables and SQL/DS columns between SQL/DS and using programs. It is pointed to from the RDIVPARM or RDIAUXPA fields of the RDIIN structure (passed on calls to SQL/DS).



SQLVAR node overlay structure for internal RDS processing:



SRTBASE (SORT BASE) AND AUXILIARY STRUCTURES

The SRTBASE and its auxiliary structures SORTLIST (sort list), SORTSPEC (sort specification), SARGS (search arguments), and KDOMAINS are used by the Sort component of DBSS in performing sort operations against SQL/DS objects.

The pointer to SRTBASE, along with the op code and return code parameters, is passed to the DBSI (ARIYM00) from RDS by module ARIXEDB.

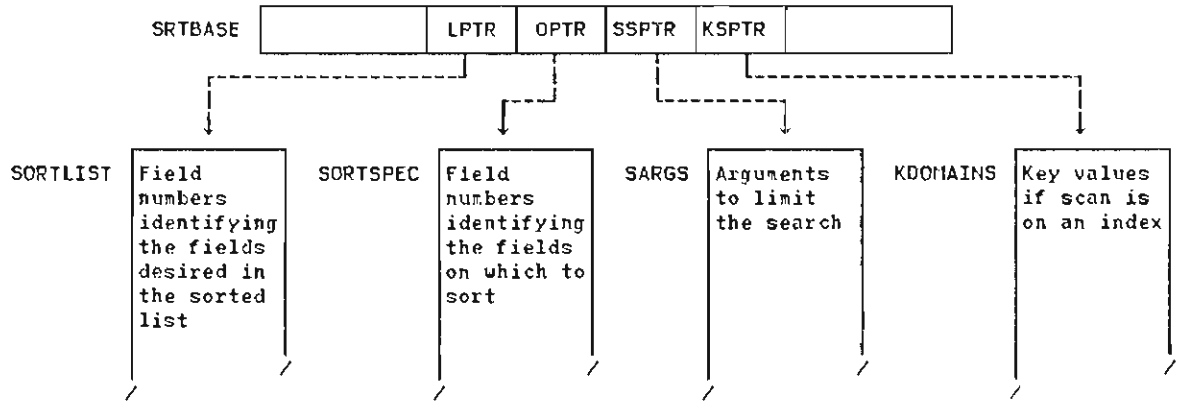
The auxiliary structure SORTLIST is pointed to by SRTBASE field LPTR. Its array dimension is SRTBASE field NSORTLIST.

The auxiliary structure SORTSPEC is pointed to by SRTBASE field OPTR. Its array dimension is SRTBASE field NSORDSPC.

The auxiliary structure SARGS is pointed to by SRTBASE field SSPTR. Its array dimension is SRTBASE field NSSARGS.

The auxiliary structure KDOMAINS is pointed to by SRTBASE field KSPTR. It array dimension is SRTBASE field NSKEYDOM.

During DBSS execution, YTABLE1 field YP points to SRTBASE, and SRTBASE itself is copied into YTABLE1. The following diagram shows the connection of the structures:



SRTBASE

Dec(Hex) SRTBASE

0 (0)	SSEGMENT - DBSPACE number of sorted list	SSCANID - Scan ID of opened scan	NSORTLIST - Number of requested fields for the sorted list	NSORDSPC - Number of submitted order specs for the sort	
8 (8)	NSSARGS - Number of submitted search arguments	NSKEYDOM - Number of submitted key fields	LPTR - Pointer to SORTLIST		
16(10)	OPTR - Pointer to sort order specifications		SSPTR - Pointer to sort search arguments		
24(18)	KSPTR - ↑ key domain fields if sort object is an index		DUPELIM[1]	SECOMP[2]	SHOLDIND[3]
32(20)	SQUALF[4]	Reserved	SRCODE - Register 15 return code		
40(28)	SRTFBACK(1) - Feedback area 1		SRTFBACK(2) - Feedback area 2		
48(30)	SRTFBACK(3) - Feedback area 3		SRTFBACK(4) - Feedback area 4		

Notes

[1] Duplicate elimination indicator:
'K' - Eliminate duplicate with respect to sort key.
'S' - Stop if duplicate sort keys found.
'N' - No duplicate elimination done.

[2] Comparison operator if sort object is an index:
'M' = Match
'B' = Before
'MB' = Match or before

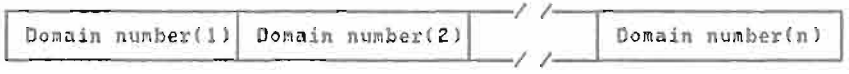
[3] Hold indicator:
'H' if hold is indicated.
Any other character means no hold.

[4] Call qualifier used in sort:
'H' = Here
'A' = After

SRTBASE AUXILIARY STRUCTURES

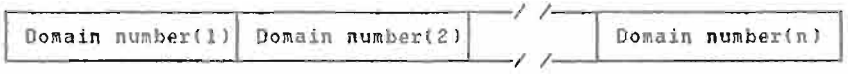
SORTLIST

SORTLIST is an array of rank NSORTLST. Each element consists of a halfword containing the domain number of the requested domain.



SORTSPEC

SORTSPEC is an array of rank NSORDSPC. Each element consists of a halfword containing the domain number(s) of the domain(s) that will be in the sort field(s).



SSARGS

SSARGS is an array of rank NSSARGS. The format is identical to SARGS in the data manipulation BASE auxiliary structure (see page 16).

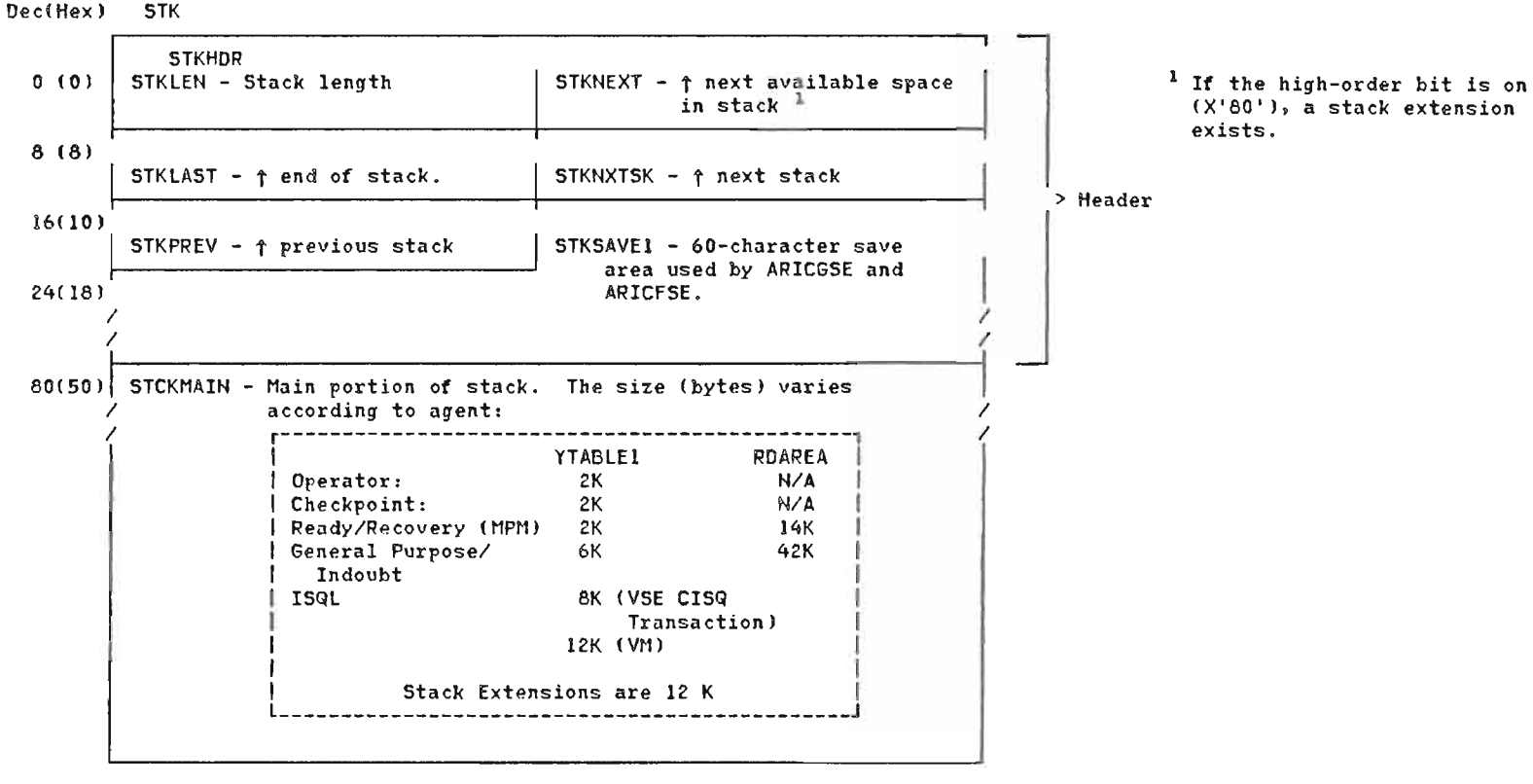
KDOMAINS

The key domain descriptor is used to describe submitted key domains if the sort object is an index. The format is described in KDOMAINS in the BASE auxiliary structures (see page 15).

STK (STACK)

ARIBSTK is the mapping macro for the stack. The stack is where all automatic storage in all SQL/DS modules is allocated from.

The pointer to the stack is in field RDASTK1 (offset X'0C') of the RDAREA and field YT1STK (offset X'0C') of YTABLE1.



STOLDSTR (STORE OR LOAD STRUCTURE)

The STOLDSTR is pointed to by field PSTOLDPT of the PREP Control Block (PREPOSCB). STOLDSTR holds the input passed by ARIXEDS to the access module load routine (ARIXELX), the access module store routine (ARIXESX) and the Re-PREP routine (ARIXERP).

Note: Read "AUX" as "access module".

Dec(Hex)	STOLDSTR				
0 (0)	STLDCTOR - Creator of the access module				
8 (8)	STLDTNML - Length of access module name	STLDTNAM - Access module name. (18 characters)			
16(10)	/				/
24(18)			Reserved	STLDSEG# - [1]	
32(20)	STLDRID - [2]	STLDULID - [3]	STLD#SLT - [4]	STLD#SEC - [5]	
40(28)	STLDSTBA - ↑ RSIBASE for storing		STLDLDBA - ↑ RSIBASE for loading		
48(30)	STLDSLTP - ↑ PSLT		STLDENTP - ↑ current entry in PSLT		
56(38)	STLDAUXP - ↑ AUX		STLDSSPT - ↑ SQL statement		
64(40)	STLD1TID - Primary TID (row ID); the first row in the AUX.		STLDSATI - TID (row ID) of entry in SYSACCESS		
72(48)	STLDCSQI - TID of last inserted SQL statement row in AUX		STLDCSET - TID of last inserted section row in AUX		
80(50) 88(58)	STLDWHEN - Timestamp (17 characters)				
96(60)		STLDFLGS[6]	STLDVALD	STLDFLGI[7]	STLDDBTD - TID of row in SYSDBSPACE.
104(68)	STLD#USD - [8]		Reserved		STLDSTID - TID of last SLT row in AUX. 111(6F)

[8] STLD#USD - Total number of sections used in modifiable AUX (not reusable).

[1] DBSPACE number that AUX resides in.
[2] RID (Table ID) of AUX.
[3] Unary link ID for this AUX relation.
[4] Number of sections in AUX.
[5] Total bytes stored for current section.
[6] STLDFLGS:

STLDSBIT	1...	Store access module.
STLDDBIT	.1..	Existing AUX of same name being replaced.
STLDRPRP	..1.	In Re-PREP mode.
	..0.	Not in Re-PREP mode.
STLDVBIT	...1	Storing a view.
	...0	Not working with a view.
STLDKBIT	.... 1...	Keep existing RUNAUTH for replaced AUX.
STLDDONE	.... .1..	Done with Re-PREP.
STLDDESC	.... ..1.	DESCRIBEs are permitted for queries in this AUX.
STLDNEW	1	This AUX is new; it does not replace an existing AUX of the same name.

[7] STLDFLGI:

STLDQUER	1...	The existing section is a query.
STLDMODF	.1..	AUX may be modified.
STLDLAST	..1.	Set when section being added to modifiable AUX is last section in PSLT.
STLDNCOM	...1	Set by ARIXERD1 on termination to signal to ARIXESX3 to avoid issuing COMMIT.
	 xxxx	Reserved.

TPMAP ENTRY (TPENTRY - ENTRY FOR LOGICAL UNIT OF WORK)

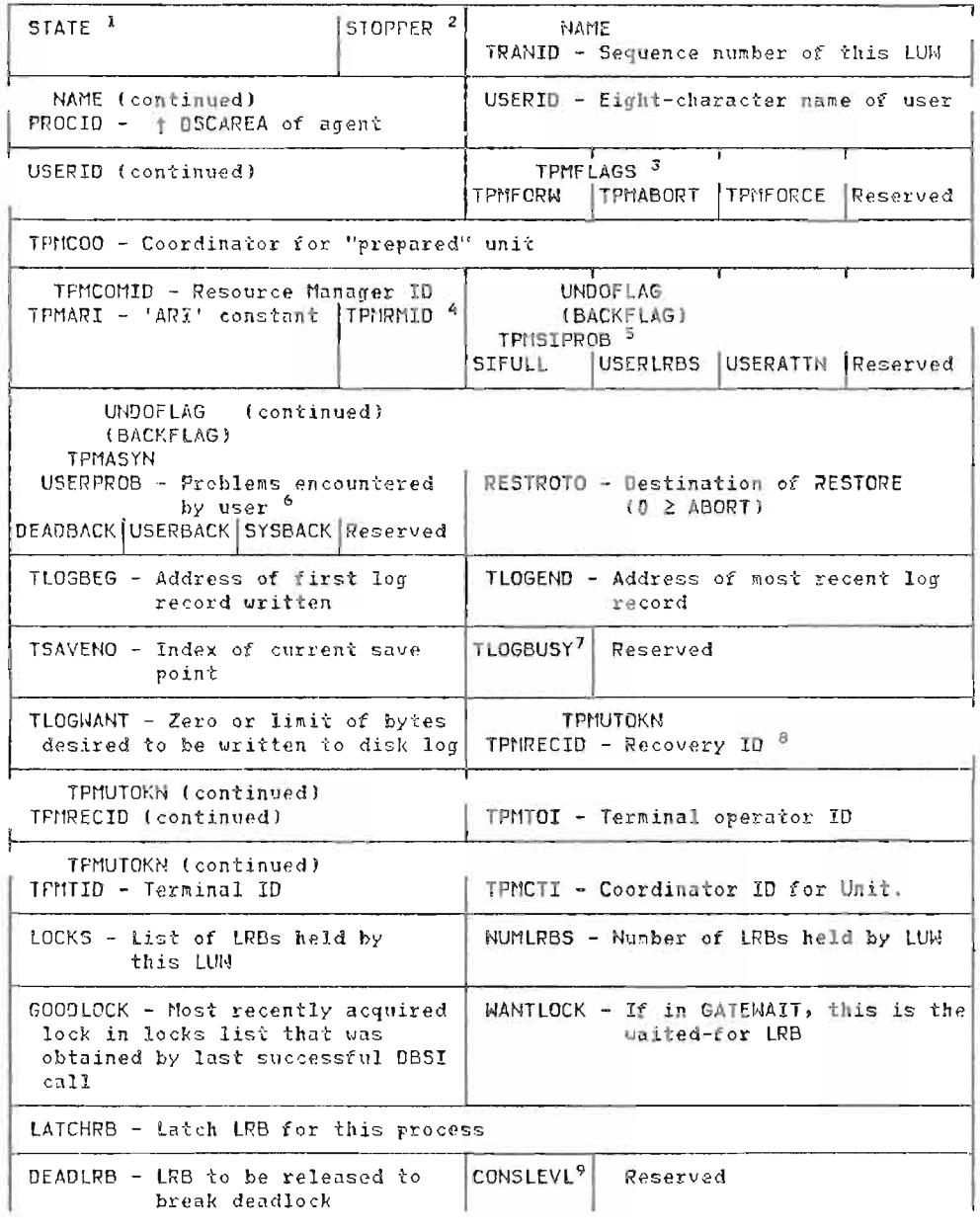
The TPMAP is a set of entries for each system or general agent. TPMAPP in YRSSCVT points to the start of the TPMAP and first entry. Specfific entries are accessed via the pointer WHOAMIP in the YTABLE1 of an agent, or via an index. The field WHOAMI in YTABLE1 is the required index for a given agent.

Each entry contains control information for management of a logical unit of work by the Work, Log and Lock subcomponents of DBSS.

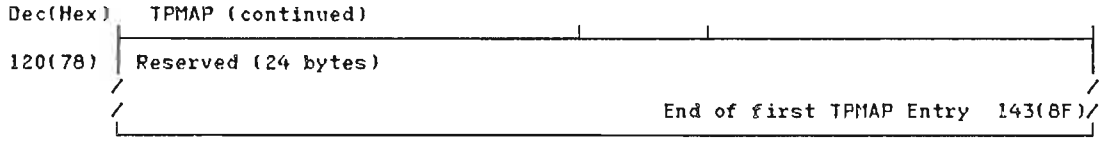
Alphabetic List of Field Names

BACKFLAG	36(24)	TLOGBEG	48(30)	TPMRMID	35(23)
CONSLEVL	116(74)	TLOGBUSY	60(3C)	TPMSIPROB	36(24)
DEADBACK	40(28)	TLOGEND	52(34)	TPMTID	80(50)
DEADLRB	112(70)	TLOGWANT	64(40)	TPMTOI	76(4C)
GOODLOCK	96(60)	TPMABORT	21(15)	TPMUTOKN	68(44)
LATCHRB	104(68)	TPMARI	32(20)	TRANID	4(4)
LOCKS	88(58)	TPMASYN	40(28)	TSAVENO	56(38)
NAME	4(4)	TPMCONID	32(20)	UNDOFLAG	36(24)
NUMLRBS	92(5C)	TPMCOO	24(18)	USERATTN	38(26)
PROCID	8(8)	TPMCTI	84(54)	USERBACK	41(29)
RESTROTO	44(2C)	TPMFLAGS	20(14)	USERID	12(C)
SIFULL	36(24)	TPMFORCE	22(16)	USERLRBS	37(25)
STATE	0(0)	TPMFORW	20(14)	USERPROB	40(28)
STOPPER	3(3)	TPMRECID	68(44)	WANTLOCK	100(64)
SYSBACK	42(2A)				

Dec(Hex) TPMAP
(TPENTRY)



[1] See TPMAP flags, page 214
[2] Zero or index of LUN that raised stop flag. (The stop flag is raised during checkpoint and prevents entry into DBSS.)
[3] See TPMAP flags, page 214
[4] Binary Resource Manager ID that was input to CIRB.
[5] SIFULL - DBSI ran out of resources
USERLRBS - User exceeded his LRB limit
USERATTN - User attention signal
[6] DEADBACK - ROLLBACK due to deadlock
USERBACK - User initiated ROLLBACK
SYSBACK - System initiated ROLLBACK
[7] 'Y' = Log being written
'N' = Log not being written
[8] The token provided by the user (subsystem) at PREPARE-TO-COMMIT time. In case of a system or subsystem crash, the user subsystem may input this token to resolve the doubt.
[9] Consistency level - copy of YTABLE1/CLEVEL



TPMAP Flags

Offset	Field Name	Hex Value	Meanings
0(0)	STATE (3 bytes)		
	STATUS		
	ASSIGNED	800000	TPMAP entry not free
	INTRAN	400000	LUW in progress
	GOLDEN	200000	Backing up; doesn't lose deadlock fights
	INRSS	100000	DBSS call in progress
	RSSSTOP	080000	Stopped by someone or will be at next entry
	RSIWAIT	040000	Waiting to get into DBSS; agent is stopped
	GATEWAIT	020000	Waiting for lock
	LACTWAIT	010000	Waiting for latch
	MONWAIT	008000	Monitor (Operator) event wait
	CHKPT	004000	Checkpointer wait
	IOWAIT	002000	I/O wait
	TPMNDOUT	001000	A unit is prepared for COMMIT or ROLLBACK
		000800	Reserved
	LOGWAIT	000400	Log I/O wait
	SPWAIT	000200	Suspend wait
		000100	Reserved
20(14)	TPMFLAGS (4 bytes)		
20(14)	TPMFORW		
	TPMFORS	1...	Asynchronous FORCE COMMIT has been activated.
	TPMFWID	.1..	Asynchronous FORCE COMMIT started by operator command.
	TPMFWS	..1.	Asynchronous FORCE COMMIT started by system after application caused a normal link drop
		...x xxxx	Reserved
21(15)	TPMABORT		
	TPMREMDC	1...	"Remember DBSSCODE". After an asynchronous abort (ROLLBACK), the agent is cleaned up.
			Large mail boxes are freed, etc. If there was no disable of the link, the application still owns the link and may issue another SQL call. The bit tells the cleanup not to clear the reason for the abort so that the DBSSCODE can be converted to a SQLCODE.
	TPMDOSCH	.1..	Do system checkpoint after ROLLBACK. Bit is turned on by Abort (ARIYT16) when reason is 'LOG FULL'. Turned off by UNDO (ARIYT19) after work is rolled back.
		..xx xxxx	Reserved (and binary zeros)
22(16)	TPMFORCE		
	TPMSHOWA	1...	SHOW ACTIVE command has been executed since the unit of work started. A FORCE command may be entered for this unit.
	TPMSUBSY	.1..	The link associated with this unit belongs to a subsystem (such as the Online Resource Manager). The DISABLE option of the FORCE command may not be used.
	TPMWAPTC	..1.	The unit is indoubt and a BEGIN WORK was completed after PTC.
		...x xxxx	Reserved (and binary zeros).
23(17)		xxxx xxxx	Reserved (and binary zeros).

TRACE POINT DESCRIPTOR MODULE STRUCTURES (TRACMAP)

TRACMAP describes structures/substructures in trace point descriptor modules (generated by various macros). These structures are: Trace Point Descriptor Directory Structure, Trace Point Descriptor Structure (maps for each substructure), Table-Find Directory Structure and Table-Find Structure.

The Trace Point Descriptor Directory Structure is generated by the TPDIR Macro. This directory allows the trace services modules to table-lookup and locate the Trace Point Descriptor associated with a trace CALL. There is one TPDIR macro for each Trace Point Descriptor (plus a TPDIR EOF Macro to define end of structure). The TPDIR macro list must precede the TPOINT Macros.

Trace Point Descriptor Structures are generated by a Trace Point Descriptor Macro Set (TPOINT Macro, followed by 0 or more of the secondary macros TPENT, TPEXIT, TPREP, TPCON, TPRET, TPLVAR, TPGVAR and TPCVAR and always terminated by the TPEND Macro). Trace point macro sets should be in ascending order by trace point number. A trace point descriptor set may contain only one (or none) TPENT, TPEXIT or TPRET macro (that is, they are mutually exclusive and may not be repeated). All other macros (except TPOINT and TPEND) may be repeated. Where the macros require input parameters, they must be in the same order as the input parameters. If no input parameters are required, the macros must be in the same order as you want their output to be displayed. There is one Trace Point Descriptor Structure for each Trace Point defined.

The Macros and their functions are:

- TPDIR - generates Trace Point Descriptor Directory Structure entry for a trace point (or end of directory).
- TPOINT - defines the beginning of a trace point descriptor structure.
- TPENT - defines trace point descriptor structure as a module entry trace point.
- TPEXIT - defines trace point descriptor structure as a module exit trace point.
- TPREP - defines trace point descriptor structure as a module general purpose (neither entry nor exit) trace point.
- TPCON - defines in the trace point descriptor structure a character string to be displayed when the trace point is activated.
- TPRET - defines in the trace point descriptor structure that a caller provided return code value is to be displayed when the trace point is activated.
- TPLVAR - defines in the trace point descriptor that a caller provided variable is to be displayed when the trace point is activated.
- TPGVAR - defines in the trace point descriptor structure that a global variable is to be displayed when the trace point is activated.
- TPCVAR - defines in the trace point descriptor structure that a caller provided variable is to be displayed when the trace point is activated.
- TPEND - defines the end of the trace point descriptor structure.

The Table-Find Directory Structure contains a directory entry for each unique table name specified in the Trace Point Descriptor Structures via the TABLENAME parameter of the TPGVAR Macro (plus an end of table entry). For each table named in a TPGVAR Macro, there must be a corresponding entry in the Table-Find Directory (and a corresponding Table-Find Structure). The Table-Find Directory Structure (structure name TBLDIR) is used to locate the address of the Table-Find structure (which provides trace addressability to globally addressable SQL/DS tables) specified in the TABLENAME parameter of the TPGVAR macro. A TABLENAME of all hex F's signals end of table.

There is one Table-Find Structure for each unique table name specified in the Trace Point Descriptor Structures via the TABLENAME parameter of the TPGVAR Macro. Structure name is TFSx...x where x...x is unique for each instance of the structure.

The various structures are given on the following pages.

Trace Point Descriptor Directory Structure

TPDIRSTR - Trace Point Descriptor Directory Structure (header)

Dec (Hex)		
0	TPDIRLEN - Length of directory entries excluding EOF	TPDTFDRP - ↑ Table-Find Directory Structure

TPDIRENT - Trace Point Descriptor Directory Entry

Dec (Hex)			
0	TPDIRNR - Entry trace pt number TPDEND - Constant X'FFFF' indicates end of the directory	Unused	TPDIRPTR - ↑ trace point descriptor

This is an array entry. There will be one entry in the Trace Point Descriptor Directory Structure for each TPDIR Macro.

Trace Point Descriptor Structure

TPOINT is always the first Substructure. The TPOINT macro defines the beginning of a Trace Point Descriptor. Each Trace Point Descriptor must end with a TPEND Substructure.

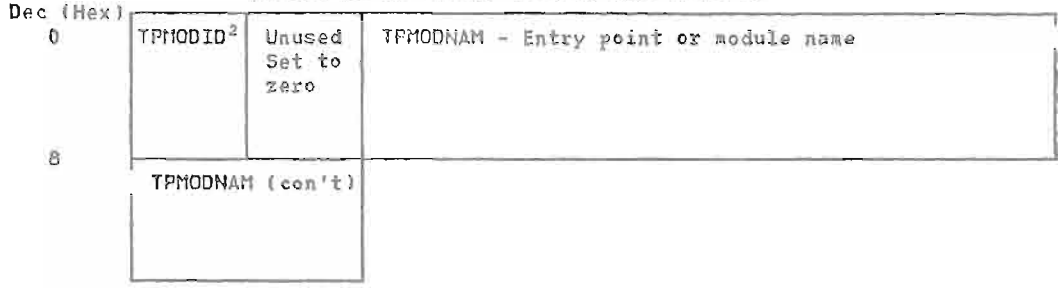
TPOINT - Trace Point Descriptor Substructure generated by TPOINT Macro.

Dec (Hex)			
0	TPOINTLN - Length of entire trace point descriptor structure	TPOINTFN number [1]	Unused Set to zero

[1] This number is the number assigned to subcomponent/function being traced. Values are:

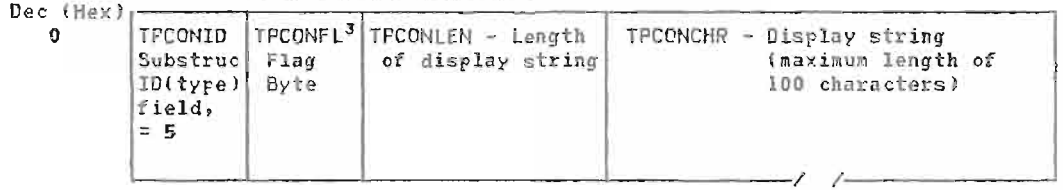
1 RDS Executives	12 DBSS Lock
2 RDS Parser	13 Reserved for DBSS LUW
3 RDS Optimizer	14 DBSS Data Control
4 RDS Code Generator	15 DBSS Data Manipulation
5 RDS Interpreter and Authorization	16 DBSS Storage (I/O)
6 RDS Security/Audit	17 DBSS Sort
9 DBSS Entry	18 DBSS Index
10 DBSS Exit	19 DBSS Update Statistics
11 DBSS Log/Recovery	

TPMOD - Trace Point Descriptor Substructure
generated by TPENT, TPEXIT, and TPREP Macros.



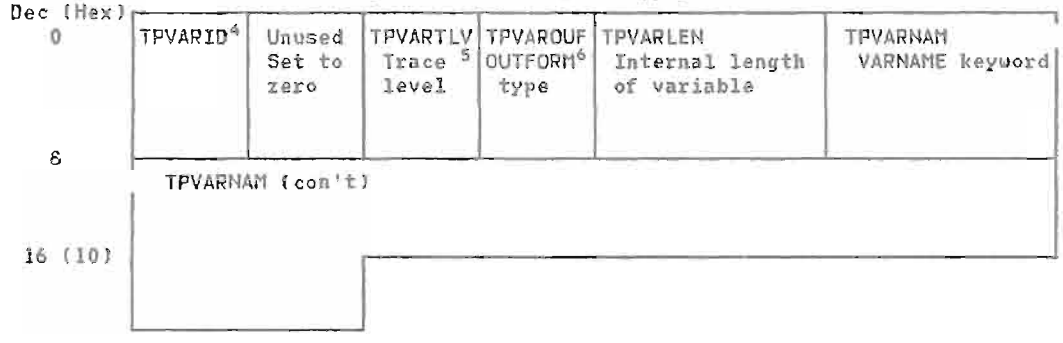
[2] Substructure ID (type) field:
2 is for TPENT
3 is for TPEXIT
4 is for TPREP

TPCON - Trace Point Descriptor
Substructure generated by TPCON Macro.



[3] Bit 0 (TPCONFL1) is a flag to indicate if string length is odd or even (0 if even), used to determine if padding character is at end of substructure for halfword alignment. Last 7 bits are spare flags. Set to 0.

TPVAR - Trace Point Descriptor Substructure
generated by TPRET, TPLVAR and TPCVAR Macros.
Note: Only the first 4 bytes apply to the TPRET substructure.



[4] Substructure ID (type) field:
6 = TPRET; 7 = TPLVAR; 9 = TPCVAR
[5] 1 or 2 (2 is default, except for TPRET where 1 is default)
[6] N = Binary-to-decimal with leading zero suppression (-) if negative
C = No conversion, but suppress low order blanks
X = Convert binary to hex
H = Display in hex-dump format (can be multi-line)
S = Special "private" formatting done in formatter utility (used for Op Tree output in RDS)

TPGVR - Trace Point Descriptor generated by TPGVAR Macro.

Dec (Hex)						
0	TPGVRID Substruc ID(type) field, = 8	Unused Set to zero	TPGVRTLV Trace level (1 or 2, default is 2)	TPGVROUF OUTFORM[6] type	TPGVRLEN - Internal length of variable	TPGVRNAM- VARNAME keyword
8	TPGVRNAM (con't)					TPGVRTBL[7] - System table name
16 (10)	TPGVRTBL (con't)					TPGVROFF - Displacement of VARNAME within system table

[7] Level 1 declared name of the system table containing the variable identified by VARNAME.

The TPEND Macro (last macro in Trace Point Descriptor) generates the last substructure in the Trace Point Descriptor Structure. This last substructure consist of a fixed (8) (one byte field) with a value of 255 ('FF'X).

Table-Find Directory Structure

For each table name specified in a TPGVAR macro, there must be a Table-Find Directory entry (array entry) in the Table-Find Directory Structure.

TBLDRMAP - Table-Find Directory Entry (array entry)

Dec (Hex)		
0	TBLDRNAM Global table name or TBLDREOT Indicate EOT [8]	TBLDRPTR - Address of Table-Find Structure for global table name

[8] EOT - End of table, 'FFFFFFFFFFFFFFFF'X

Table-Find Structure

For each table name specified in a TPGVAR macro, there must be a Table-Find Structure which is an array of flag and displacement entries as shown below.

TBLFIND - Table-Find Structure Array Entry

Dec (Hex)			
0	TBLFNDFL 9	Unused	TBLFNDSP - Displacement within table for the pointer to the next table (or 0 if no next pointer)

9 1... .... Last entry (no more array entries, if 0 then there are more array entries).
.0.. .... Offset, locates table pointer (if 0 TBLFNDSP is the displacement in the current table to the next table in the address chain to the target table. If 1, target table already found.)
..00 0000 Reserved

TRACE POINT OUTPUT OBJECTS (ARICTRO)

These objects are used to describe the contents of the SQL/DS Trace Output file. They are used by DBSS and RDS Trace Services Modules and by the Trace File-to-Print utility program ARIMTRA. They can also be used by DBSS and RDS modules which explicitly create trace point output records. Trace Point Output Objects correspond to Trace Point Descriptor Module Structures (TRACMAP). The header is the first object outputted for each trace point. The header precedes all other trace point information in the printed listing.

TROHDOBJ Trace Point Header Object

Dec (Hex)					
0	TROHDCTL Record/ object control flags [1]	TROHDTYP Output object (type) field, = 1	TROHDTPN - Trace point number (TP#)	TROHDCHK - Contains 'DS2' to allow trace print program to verify first LRECL is SQL/DS trace header	TROHDMCO Identif. as DBSS (D)/RDS (R)
8	TROHDSCO [2] - Number assigned to subcomponent/ function being traced.		TROHDAGT - Traced Agent Structure	TROHDLUW - DBSS agent LUW-ID. Undefined, if RDS header object.	
16 (10)	TROHDUID - Trace agent structure userid				
24 (18)	TROHDCLK - Used to display date and time in printed output Is S/370 Time of Day Clock value.				
32 (20)	TROHDOP - Gives the current DBSS/ RDS opcode being executed	TROHDRC - Return code value from TPRET object (or null - see TROHDFL5)			

[1] If bit 0 (TROHDFL1) equals 1 then there are more objects in LRECL. The flags in bits 1 - 3 do not apply to HDROBJ. If bit 4 (TROHDFL5) equals 1 then the header contains return code (TROHDRC). Bits 5 - 7 equal 0 and are reserved.

[2] Values are:
- 1 RDS Executives
- 2 RDS Parser
- 3 RDS Optimizer
- 4 RDS Code Generator
- 5 RDS Interpreter and Authorization
- 6 RDS Security/Audit
- 9 DBSS Entry
- 10 DBSS Exit
- 11 DBSS Log/Recovery
- 12 DBSS Lock
- 13 Reserved for DBSS LUW
- 14 DBSS Data Control
- 15 DBSS Data Manipulation
- 16 DBSS Storage (I/O)
- 17 DBSS Sort
- 18 DBSS Index
- 19 DBSS Update Statistics

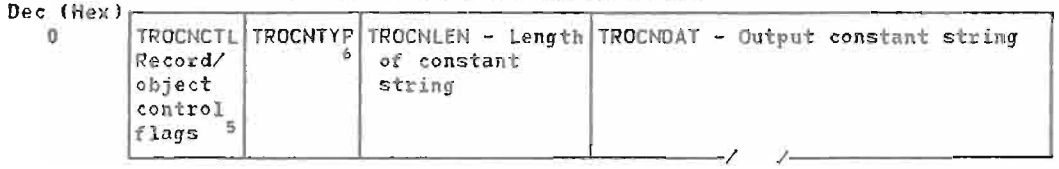
TROMDOBJ - TPENT/TPEXIT/TPREP Trace Output Object (MODOBJ)
TROMDOBJ corresponds to TPMOD - Trace Point Descriptor Substructure in TRACMAP.

Dec (Hex)			
0	TROMDCTL Record/ object control flags [3]	TROMDTYP [4]	TROMDNAM - Entry point/module name
8	TROMDNAM (con't)		

[3] If bit 0 (TROMDFL1) equals 1 then there are more objects in LRECL. The last 7 flag bits do not apply to MODOBJ.

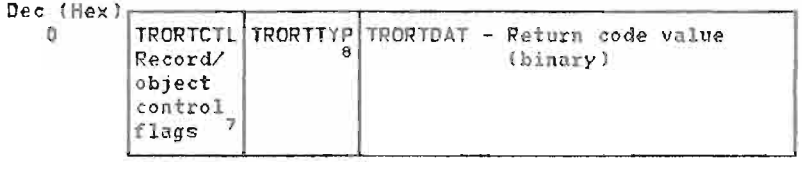
[4] Output object ID (type) field:
- 2 = TPENT Substructure ID
- 3 = TPEXIT Substructure ID
- 4 = TPREP Substructure ID

TROCNOBJ - TPCON Trace Output Object
TROCNOBJ corresponds to TPCON - Trace Point
Descriptor Substructure in TRACMAP.



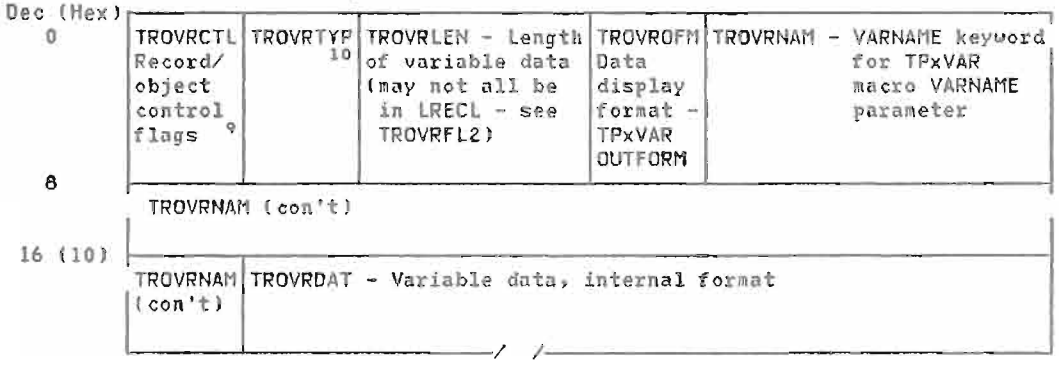
[5] If bit 0 (TROMDFL1) equals 1 then there are more objects in LRECL. The last 7 flag bits do not apply to TPCON.
[6] Output object ID (type) field equals 5.

TRORTOBJ - TPRET Trace Output Object
TRORTOBJ corresponds to TPVAR - Trace Point
Descriptor Substructure in TRACMAP.



[7] If bit 0 (TRORTF1) equals 1 then there are more objects in LRECL. The last 7 flag bits do not apply to TPRET.
[8] Output object ID (type) field equals 6.

TROVROBJ - TPLVAR/TPCVAR Trace Output Object (VAROBJ)
TROVROBJ corresponds to TPVAR - Trace Point
Descriptor Substructure in TRACMAP.



[9] If bit 0 (TROVRFL1) equals 1 then there are more objects in LRECL.
Bit 1 (TROVRFL2) for OUTFORM types H and S only, object data spans into the next LRECL (and begins here).
The last 6 flag bits do not apply to VAROBJ.
[10] Output object ID (type) field:
7 = TPLVAR Substructure ID
9 = TPCVAR Substructure ID

TROGVOBJ - TPGVAR Trace Output Object
TROGVOBJ corresponds to TPGVR - Trace Point Descriptor Substructure in TRACMAP.

Dec (Hex)					
0	TROGVCTL Record/ object control flags [11]	TROGVTYP [12]	TROGVLEN - Length of variable data (may not all be in LRECL - see TROGVFL2)	TROGVOFM Data display format- TPGVAR OUTFORM	TROGVNAM - VARNAME keyword for TPGVAR Macro VARNAME
8	TROGVNAM (con't)				TROGVDAT - Variable data, internal format

[11] If bit 0 (TROGVFL1) equals 1 then there are more objects in LRECL.
Bit 1 (TROGVFL2) for OUTFORM types H and S only, object data spans into next LRECL (and begins here).
The last 7 flag bits do not apply to TPGVAR.

[12] Output object ID (type) field equals 8

One or more TROSPOBJs are created whenever a TPLVAR, TPGVAR, or TPCVAR output object is created with data display format type of H or S and the variable data is too long to fit in one LRECL. Record/object control flags indicate data spanning. When data spanning occurs, the LRECLs contain only one output object. The data length is implied by the LRECL length.

TROSPOBJ - Data Continuation Trace Output Object (SPNOBJ)

Dec (Hex)			
0	TROSPCTL Record/ object control flags [13]	TROSPTYP [14]	TROSPDAT - Variable length data - goes to end of LRECL

[13] Bit 0 (TROSPFL1) is always off (0) for TROSPOBJ.
Bit 1 (TROSPFL2) is always off for TROSPOBJ.
If bit 2 (TROSPFL3) equals 1 then data continuation object but not last control object.
If bit 3 (TROSPFL4) equals 1 then data continuation object and last control object.
Last 4 bits are spare flags, and always equal 0.

[14] Output object ID (type) field ID is of object for which this is a control object.

USER LIST AND USER LIST DIRECTORY DATA AREAS

The ARIBUSR macro maps the User List (UDIULIST) and User List Directory (UDIRECT) data areas. UDIULIST is a list of userids authorized to run a particular access module. In each entry of PROGS (list of loaded access modules), is an index into the user list directory (PROGUDIX). The user list directory shows how many access modules with the same name are currently loaded, and points to the top of a chain of users authorized to run this access module.

The following fields in RDCVT are used with UDIULIST and UDIRECT:

- RDCUSERL - Address of the start of user list
- RDC#USER - Number of slots in the user list
- RDCUFREE - Index into user list of the first free slot or zero
- RDCUDIRP - Address of the start of the user list directory

Dec(Hex) USRLIST - User list

0(0)	USRNXTIX - Index of next list element	USRID - Userid
8(8)	USRID (continued) 11(8)	

Dec(Hex) UDIRECT - User list directory

0(0)	UDICOUNT - Number of PROGS referencing user list	UDIULIST - Index into user list 7(7)

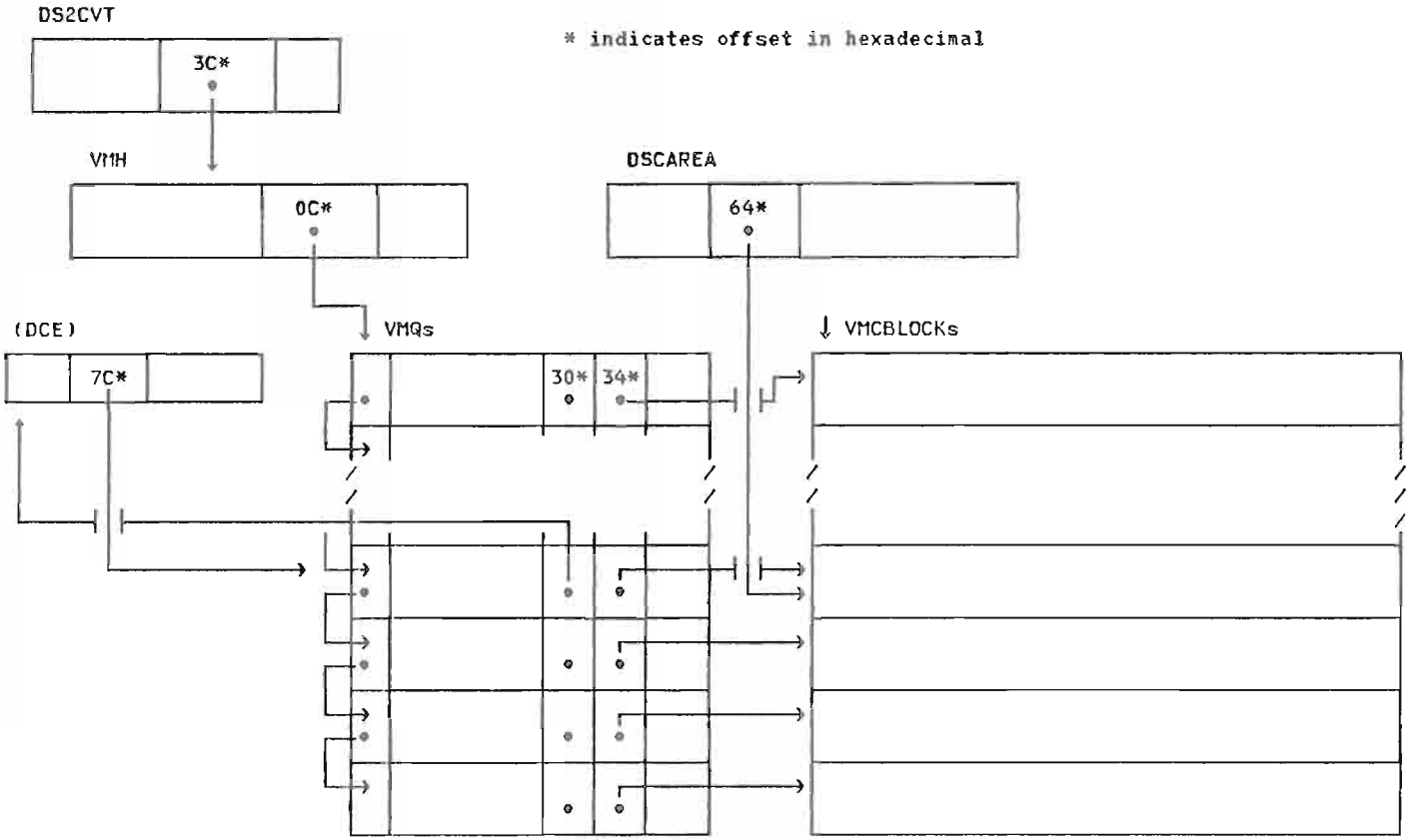
VM CROSS-MACHINE COMMUNICATION DATA AREAS

The following diagram shows the general connections of the SQL/DS VM cross-machine communication control blocks.

The VMH (VM Communication Block Queue Head), VMQs (VM Communication Block Queue Elements), and VMCBLOCK (VM Cross-Machine Control Block) are shown on the following pages:

VMCBLOCK - 225
VMH - 230
VMQ - 231

The VMQ elements and the VMCBLOCKs together make up the "pseudo-agents".



VMCBLOCK (VM CROSS-MACHINE COMMUNICATION CONTROL BLOCK)

The VMCBLOCK is used by the Communication Manager (ARICCOM) and the external interrupt handlers (ARICINT and ARIRINT) to perform the VM IUCV functions. This block is divided into three sections:

- A common section used to keep information about the IUCV requests or interrupts IUCV function requests
- IUCV parameter list used by the Communication Manager to perform the IUCV request
- IUCV buffer save area used by the external interrupt handler to save the contents of the IUCV buffer.

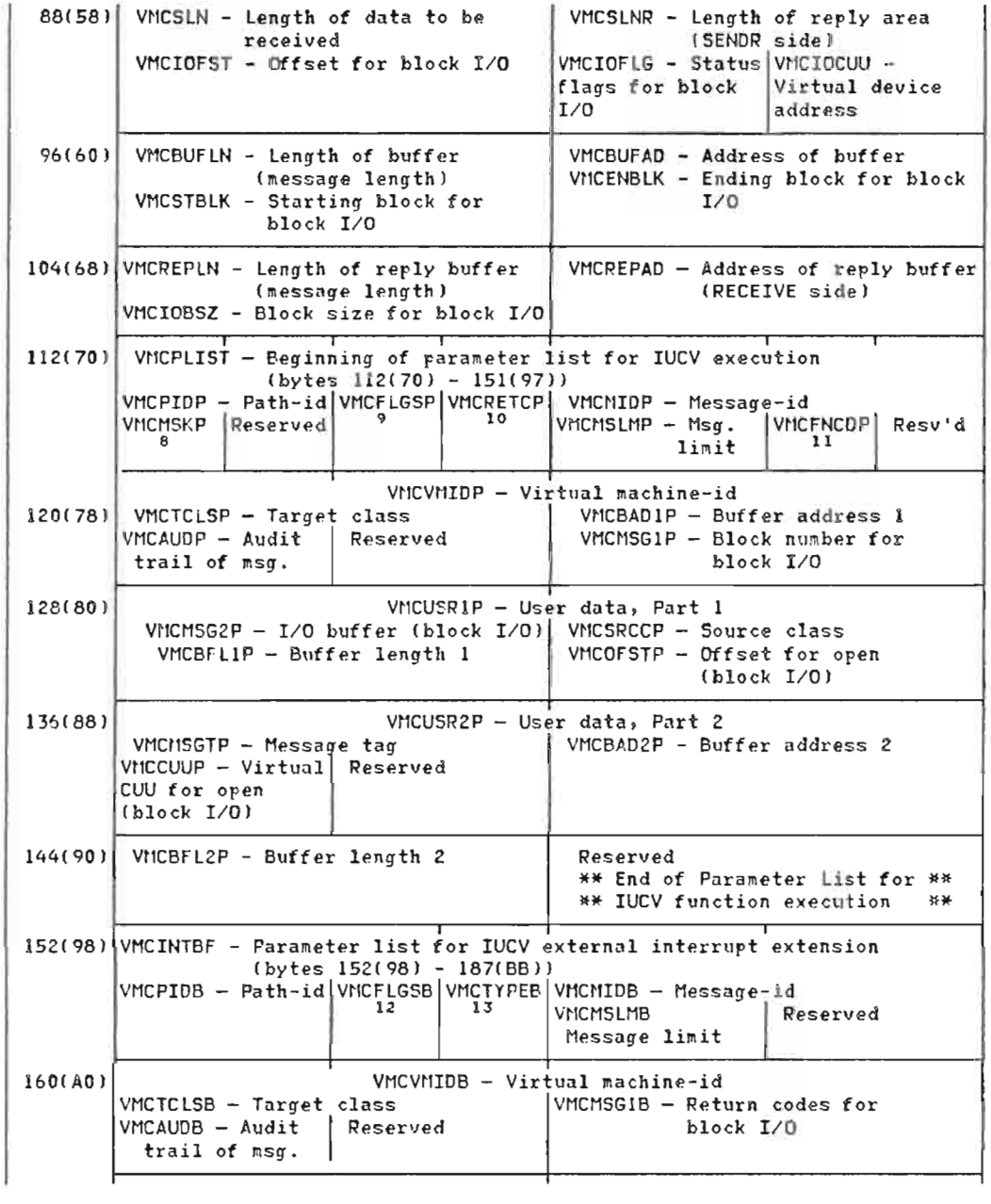
The VMCBLOCK contains the communication ECBs (event control blocks) to be posted, a flag lock for the COMPARE and SWAP logic, user data fields, control bytes used by ARICCOM, identification fields (path-id and message-id), pointers to and lengths of buffers (REPLY and SEND side), and the length of the VMCBLOCK itself.

Extensions of the control block contain fields for all of the data necessary to execute the IUCV functions (serves as a parameter list) and a data area in which is stored the IUCV external interrupt buffer data by the external interrupt handler.

Dec(Hex) VMCBLOCK

0 (0)	VMCCATCH – Eyecatcher							
8 (8)	VMCLENTH – Control block length				VMCLOCK – Compare word for external interrupt			
16(10)	VMCPTHID – Communication path id		VMCMSGLM – IUCV message limit (default equal to 1)		VMCMXCNS – Maximum number of connections		VMCTSKID – Task-id for resource manager	
24(18)	VMCCTLFL – Control bytes				VMCERRFL – Error fields			
	VMCFNCD [1]	VMCOPENF [2]	VMCMASK [3]	VMCLPFN	VCMRETCD [4]	VMCRESCD [5]	VMCSEVCD [6]	VMCREJCD [7]
32(20)	VMCVMID – VM-id of source virtual machine							
40(28)	VMCTOAP – VM-id of target machine							
48(30)	VMCDDNAM – DDNAME for block I/O							
56(38)	VMCUDTA1 – User data, Part 1							
64(40)	VMCUDAT2 – User data, Part 2							
72(48)	VMCMSGID – IUCV message-id				VMCCECB – CONNECT/ACCEPT ECB			
80(50)	VMCSECB – SEND ECB				VMCRECB – RECEIVE ECB VMCNOBLK – Number of blocks for block I/O			

[1] Communication Manager function request code.
[2] Comm. Manager IUCV option code.
[3] Comm. Manager IUCV SETCMASK byte.
[4] Comm. Manager return code. See page 227 for return code values and meanings.
[5] Comm. Manager reason code (see page 228)
[6] Comm. Manager IUCV SEVER return code.
[7] Comm. Manager IUCV REJECT return code.



8 Mask byte for SETCMASK

9 IUCV option flags:

```
VMCFPDTA 1... .... IPRMDATA
VMCFPQSC .1.. .... IPQUSCE
VMCFPPRT ..1. .... IPPRTY
VMCFPNRP ...1 .... IPNORPY
         .... x... Unused
VMCFPMID .... .1.. IPFGMID
VMCFPPID .... ..1. IPFGPID
VMCFPMCL .... ...1 IPFGMCL
```

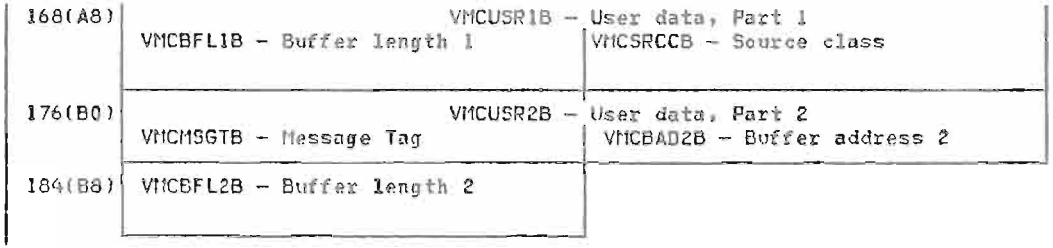
10 Return codes

11 Function code

12 IUCV option flags:

```
VMCFBDTA 1... .... IPRMDATA
VMCFBQSC .1.. .... IPQUSCE
VMCFBPRT ..1. .... IPPRTY
VMCFBNRP ...1 .... IPNORPY
         .... x... Unused
VMCFBMID .... .1.. IPFGMID
VMCFBPID .... ..1. IPFGPID
VMCFBMCL .... ...1 IPFGMCL
```

13 Type of external interrupt



Return Codes:

Note: 1000 will be added to all CP IUCV codes by CMS when returned from CP to CMS IUCV. ARICCOM will then subtract 900 from these codes and add 100 to all CP IUCV codes returned directly to ARICCOM from CP. This results in a "1xx" format for all CP IUCV return codes when put into the CPLIST, which clears up any ambiguities between CP and CMS return codes as they are represented below.

Name	Hex Val	Return From	Meaning
VMCREOK	00	All functions	Request OK
VMCINPID	01	PROBLEM ACCEPT, SENDR, RECEIVE, REPLY, RESETR, DISCONNECT PURGE, DISCONNECT ALL, CLOSE	Invalid path-id. Register 15 contains 12.
VMCPTHQS	02	SENDR	Path quiesced, no SENDs allowed. Register 15 contains 12.
VMCMLIMX	03	SENDR	Message limit exceeded. Register 15 contains 12.
VMCNPRY	04	SENDR	Priority messages not allowed on this path. Register 15 contains 12.
VMCBADRI	04	OPEN	High order byte of Register 1 is not zero. Register 15 contains 12.
VMCHSTPI	04	LOGON	User with this name previously issued HNDIUCV SET. Register 15 contains 12.
VMCBFTS	05	RECEIVE, REPLY	Receive or answer buffer too short to contain message. Register 15 contains 12.
VMCPX	06	RECEIVE, REPLY	Fetch/storage protection exception on send/answer buffer. Translated to CPLSNPX (121) on RECEIVE. Translated to CPLANPX (123) on REPLY. Register 15 contains 12.
VMCAX	07	RECEIVE, REPLY	Address exception on send or answer buffer. Translated to CPLSNAX (122) on RECEIVE. Translated to CPLANAX (124) on REPLY Register 15 contains 12.
VMCINVMP	08	RECEIVE, REPLY, RESETR, DISCONNECT PURGE, DISCONNECT ALL, CLOSE	Message-id found, message class or path-id invalid. Register 15 contain 12.
VMCNHNDS	08	LOGOFF, CONNECT SPECIFIC, ACCEPT, DISCONNECT PURGE, DISCONNECT ALL, OPEN, CLOSE	No HNDIUCV SET has been issued for this application. Register 15 contains 12.
VMCCLR	09	RECEIVE, REPLY	Message has been purged. Maps to CPLCCLR (X'11'). Register 15 contains 4.
VMCNEGML	0A	SENDR, REPLY	Message length is negative. Register 15 contains 12
VMCNOLOG	0B	CONNECT	Target communicator has not logged on. Maps to CPLNOLOG (X'0A'). Register 15 contains 4.

VMCDASD	0C	OPEN	DASD not reserved with the RESERVE command. Register 15 contains 12.
VMCNDCLB	0C	CONNECT	Target communicator has not declared buffer. Maps to CPLNIDN (X'0B'). Register 15 contains 4.
VMCUNPTH	0C	ACCEPT, DISCONNECT PURGE, DISCONNECT ALL, CLOSE	The user does not own the path. Register 15 contains 12.
VMCTMCR	0D	CONNECT, OPEN	Maximum number of CONNECTs for originator exceeded. Register 15 contains 12.
VMCMCRT	0E	CONNECT	Maximum number of CONNECTs for target exceeded. Maps to CPLNCNN (X'0C'). Register 15 contains 4.
VMCNOAF	0F	CONNECT	No authorization found. Register 15 contains 12.
VMCINVSN	10	CONNECT, OPEN	Invalid CP system service name. Register 15 contains 12.
VMCNAME	10	CONNECT, OPEN, ACCEPT, SEVER, LOGON, LOGOFF	Name parameter is not specified or its address is equal to zero. Register 15 contains 12.
VMCINVFC	11	CONNECT, OPEN	Invalid function code in IPFCNCD. Register 15 contains 12.
VMCMLMX	12	CONNECT, ACCEPT, SENDR, OPEN	Value in message limit exceeds 255. Register 15 contains 12.
VMCDBFPI	13	LOGON	Declare buffer previously issued. Register 15 contains 12.
VMCNCNOS	14	ACCEPT	Connection cannot be completed; originator severed. Maps to CPLABDC (X'07'). Register 15 contains 4.
VMCEXIT	14	LOGON	Exit parameter not specified or its address is equal to zero. Register 15 contains 12.
VMCPARM	18	CONNECT, OPEN, ACCEPT, SEVER	PARMLIST parameter not specified or its address is equal to zero. Register 15 contains 12.
VMCIPALL	1C	DISCONNECT PURGE, DISCONNECT ALL, CLOSE	An IUCV SEVER with the IPALL bit ON is not allowed. Register 15 contains 12.
VMCDDNMD	1C	OPEN	DDNAME not defined. Register 15 contains 12.
VMCNCMSU	20	LOGON	An IUCV DECLARE BUFFER has already been issued by a non-CMS IUCV user. CMS IUCV support cannot be initialized. Register 15 contains 12.
VMCDRERR	24	LOGON	Errors were encountered reading the directory for the virtual machine during IUCV initialization Register 15 contains 12.
VMCINVFN	28	LOGON, LOGOFF, CONNECT, OPEN, ACCEPT, CLOSE, DISCONNECT PURGE, DISCONNECT ALL	Unrecognized function name: not SET, REP, CLR, CONNECT, ACCEPT, SEVER. Register 15 contains 12.
VMCDSKNA	64	OPEN	Disk not attached. Register 15 contains 12.
VMCIOERR1	1xx	OPEN	An I/O error occurred, xx = return code from DIAG 18 (CKD devices). Register 15 contains 12.
VMCIOERR2	2xx	OPEN	An I/O error occurred, xx = return code from DIAG 20 (FBA devices). Register 15 contains 12.
VMCFRERR	2xx	LOGON, LOGOFF, DISCONNECT PURGE, DISCONNECT ALL, CLOSE	An error was encountered in freeing or freting CMS free storage. 'xx' = the return code from DMSFREE or DMSFRET. Register 15 contains 12.

Reason Codes:

Name	Hex Val	Meaning
VMCDEVND	01	Virtual device not defined; returned on OPEN.
VMCMRJCT	01	Message was rejected.
VMCDEVNS	02	Virtual device not supported; returned on OPEN.
VMCBSZNS	03	Block size not supported; returned on OPEN.
VMCPTHAE	04	IUCV path already exists; returned on OPEN.
VMCPRMSN	05	Connection not using PRMDATA = YES; returned on OPEN.

VMCRPLE	05	Reply too long for buffer.
VMCSVRDN	06	Path was severed (NORMAL DISCONNECT)
VMCSVRDA	07	Path was severed (ABNORMAL DISCONNECT)
VMCSPM	08	Single User Mode
VMCQSCE	17	Partner Issued a TERMQUIESCE
VMCSNPX	15	Protection exception on send buffer
VMCSNAX	16	Addressing exception on send buffer
VMCANPX	17	Protection exception on answer buffer
VMCANAX	18	Addressing exception on answer buffer
VMCRCPX	19	Protection exception on receive buffer
VMCRCAX	1A	Addressing exception on receive buffer
VMCRPPX	1B	Protection exception on reply buffer
VMCRPAX	1C	Addressing exception on reply buffer
VMCWRGDB	1D	Wrong DBNAME
VMCINVNT	FE	Invalid external interrupt occurred

VMH (VM COMMUNICATION BLOCK QUEUE HEADER)

The VMH (VM Communication Block Queue Header) is the primary (or global) control block for the pseudo-agent structure built by SQL/DS for its VM IUCV support. It contains pointers to the VMQ elements and to subqueues within the VMQ Elements chain (that is, to the chain of "in-use" pseudo-agents" and "waiting" pseudo-agents).

Additionally, VMH contains a similar set of pointers to the DCE chain of the real agents. These two sets of pointers are used to allocate and deallocate real agents to the pseudo-agents.

The VMH is pointed to by the DS2VMQHP field in the DS2CVT.

Dec(Hex) VMH

0 (0)	VMHEYE - Eyecatcher ("VMH")		
8 (8)	VMHLENG - Length of queue head		VMHAVMQ - Address of VMQ elements (pseudo-agents)
16(10)	VMH1APA - ↑ first available pseudo-agent		VMH1IUPA - ↑ first in-use pseudo-agent
24(18)	VMHLIUPA - ↑ last in-use psuedo-agent		VMH1ARA - ↑ first available real agent
32(20)	VMH1IURA - ↑ first in-use real agent		VMHLIURA - ↑ last in-use real agent
40(28)	VMH1WPA - ↑ first waiting pseudo-agent (RECB posted)		VMHLWPA - ↑ last waiting pseudo-agent (RECB posted)
48(30)	VMHCAPA - Count of available pseudo-agents		VMHCIUPA - Count of in-use pseudo-agents
56(38)	VMHCWTPA - Count of waiting pseudo-agents (RECB posted)		VMHCCNPA - Count of pseudo-agents connected to real agents
64(40)	VMHCARA - Count of available real agents	VMHCIURA - Count of in-use real agents	VMHMAXCN - Maximum number of IUCV connections
72(48)	Reserved (4 fullwords)		
			87(57)

VMQ (VM COMMUNICATION BLOCK QUEUE ELEMENT)

The VMQ (VM Communication Block Queue Element or "pseudo-agent") control blocks are allocated to users connected to SQL/DS via IUCV. Each VMQ Element immediate precedes its VMCBLOCK. These two control blocks are called a pseudo-agent. When a real agent becomes available, it is allocated to a "waiting" pseudo-agent. The pseudo-agents are all chained together (via VMQNEXT). Within this pseudo-agent queue are three subqueues.

- "In-use" pseudo-agent queue
- "Waiting" pseudo-agent queue (which is a subset of the "in-use" pseudo-agent queue)
- "Available" pseudo-agent queue. Whenever a user connects to SQL/DS, a pseudo-agent from the "next available" queue is allocated for that user, and placed on the "in-use" queue. When the user machine issues an IUCV SEND (with reply) request, that user's associated pseudo-agent is allocated to a real agent if one is available. If a real agent is not available is it placed on the "waiting" queue until a real agent becomes available.

At initialization time, VMQNEXT and VMQNAPA point to the same pseudo-agent. While VMQNEXT never changes, VMQNAPA will change during the course of program execution.

Dec(Hex) VMQ

0 (0)	VMQEYE - Eyecatcher ("VMQ")			
8 (8)	VMQLENG - Length of this block		VMQNEXT - ↑ next pseudo-agent (VMQ)	
16(10)	VMQUSRID - VM userid			
24(18)	VMQSQLID - SQL/DS connect id			
32(20)	VMQNAPA - ↑ next available pseudo-agent		VMQNWPA - ↑ next waiting pseudo-agent (RECB posted)	
40(28)	VMQPWPA - ↑ previous waiting pseudo-agent (RECB posted)		VMQNIUPA - ↑ next in-use pseudo-agent	
48(30)	VMQPIUPA - ↑ previous in-use pseudo-agent		VMQADRRA - ↑ real agent (DCE)	
56(38)	VMQADVMB - ↑ VMCBLOCK		VMQADYRS - ↑ YRSSCVT	
64(40)	VMQFLAGS 1	VMQRDAFL 2 / VMQRDASP 3 / VMQDSCFL 4	VMQTPMFL - save area for UNDOFLAG (TPMAP)	
72(48)				
80(50)	Reserved (4 fullwords)			95(5F)

```
1 VMQFLAGS
    VMQAVAIL  1... ....  Pseudo-agent is available.
    VMQINUSE  .1.. ....  Pseudo-agent is in-use but not
                         connected to real agent.
    VMQWAIT   ..1. ....  Pseudo-agent is waiting for a
                         real agent (RECB is posted).
    VMQALLOC  ...1 ....  Pseudo-agent is connected to a
                         real agent.
              .... xxxx  Reserved
2 VMQRDAFL - Save area for RDAFLAGS
3 VMQRDASP - Save area for RDASPEC
4 VMQDSCFL - Save area for DSCFLAGS
```

WSAELEM (WORKING STORAGE ELEMENT)

The WSAELEM is pointed to by field DSCWSASP or DSCWSACP of the DSCAREA. It is used for storage analysis and tracing of the working storage modules ARICWSG and ARICWSF.

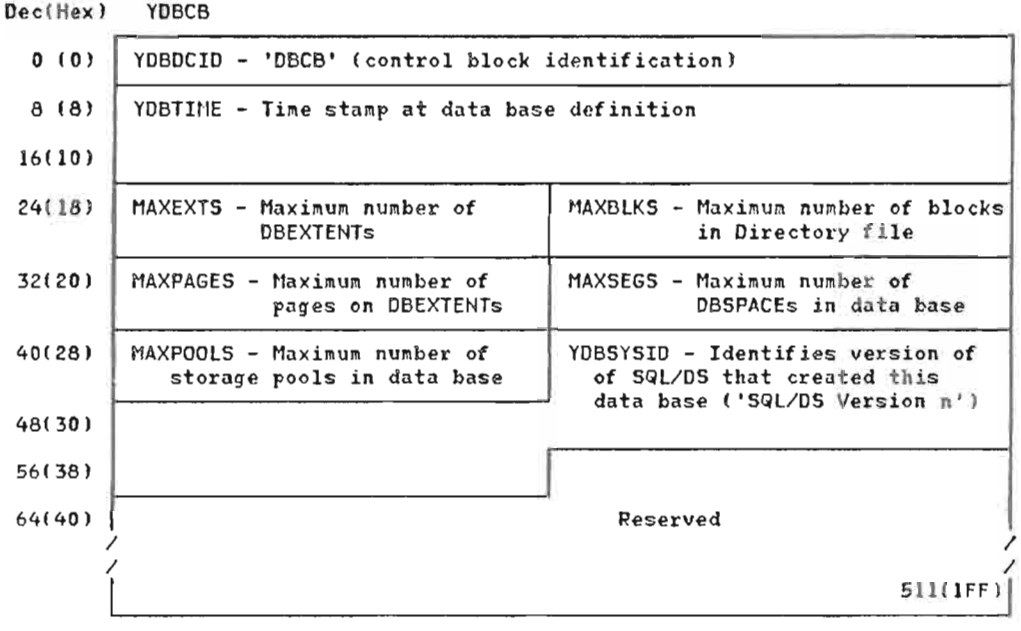
Dec(hex) WSAELEM

0 (0)	WSAMODN – Module name (name of module that called ARICWSG)	
8 (8)	WSAIR14 – Input register 14 (return point)	WSARTYPE – Call type (GET | FREE | ERR[1] | E-16[2])
16(10)	WSAAREAP – ↑ working storage area	WSAALEN – Length of working storage area
24(18)	WSAHIWAT – Working storage high water mark (bytes)	WSATOTC – Current allocated working storage

[1] Error detected by ARICWSG or ARICWSF.
[2] Error code 16 means that the wrong area is being freed (issued from ARICWSF).

YDBCB (DATA BASE CONTROL BLOCK)

The YDBCB contains information about the size of a particular data base. It is stored in the first block in the Directory file. It is read into virtual storage during DBSS initialization. The YDBCB is defined by the DBA during data base definition and is not changed thereafter. It sets the size for various control blocks in virtual storage and in the Directory file.



YRSSCVT (DBSS COMMUNICATION VECTOR TABLE)

This table contains global information about DBSS. All global resources (i.e., tables and control blocks) should be located via entries in this table. YRSSCVT is located in a commonly addressable area of storage. It is normally located via an entry in 'YTABLE1' called 'YRSSCVTP'. Each DBSS component can place information into this table or can store pointers to global tables in it.

The YRSSCVT is pointed to by:
- DS2CVT (DS2RSCVP, offset X'10')
- RDCVT (RDCYRSCV, offset X'14')
- YTABLE1 (YRSSCVTP, offset X'20')

(Also see the diagrams on pages 2 and 3.)

Alphabetic List of Field Names

ARBCUU	338(152)	
ARCDBLBF	320(140)	(flag bit)
ARCHCOM	320(140)	(flag bit)
ARCHDEF	320(140)	
ARCHEX	320(140)	(flag bit)
ARCHON	320(140)	(flag bit)
ARCHPCT	312(138)	
ARCHPTR	316(13C)	
ARCHST	320(140)	(flag bit)
ARCUU	321(141)	
AUTOCNT	332(14C)	
BLKALTP	112(70)	
CHECKPT	337(151)	
CHKCNTP	328(148)	
DBPSLN	355(163)	
DBPSWD	356(164)	
DBPSWORD	355(163)	
DS2START	225(E1)	
DUALLOG	336(150)	
FREELISP	240(F0)	
LOCWAITP	232(E8)	
LOGARCH	258(102)	
LOGCLEAN	308(134)	
LOGMODE	256(100)	
LOGSTATE	259(103)	
MASTEREP	128(80)	
MASTERP	88(58)	
MAXLOCKS	212(D4)	
MODMAPP	124(7C)	
NAMGATEP	228(E4)	
NBLKALT	164(A4)	
NBMAPBLK	156(9C)	
NCSCANS	340(154)	
NCUSERS	176(C4)	
NEXTID	208(D0)	
NLATCHES	216(D8)	
NLRBS	236(EC)	
NMASTER	148(94)	
NNAMGATE	224(E0)	
NPAGSCHK	324(144)	
NPBMAP	152(98)	
NSYSBLK	136(88)	
NTRANS	200(C8)	
NYTAB2	160(A0)	
OSEGBLK	140(8C)	
OSYSBLK	144(90)	
PBITMDP	120(78)	
PBITMP	116(74)	
PGCTRSP	176(B0)	
PLIRSSP	44(2C)	
PROGNAME	344(158)	
SEGATP	92(5C)	
SLATCHP	220(DC)	
SLOGARCH	280(118)	
SLOGBEG	272(110)	
SLOGBUF	288(120)	
SLOGCUSH	300(12C)	
SLOGEND	276(114)	
SLOGLTH	260(104)	
SLOGPAGE	284(11C)	
SLOGSCHK	268(10C)	
SLOGSEQN	264(108)	
SLOGWBUF	296(128)	
SLOGWRIT	292(124)	
TPMAPP	204(CC)	
VSAMACBB	80(50)	
YCHKPTTP	40(28)	
YCVTTIP	28(1C)	
YCVTTIS	32(20)	
YDATASIZ	180(B4)	
YDBCBP	84(54)	
YENDBUFF	184(B8)	
YMTAB1P	36(24)	
YNBCNT	52(34)	
YNBSCNT	52(34)	
YNBUCNT	54(36)	
YRSACTAG	188(BC)	
YRSBDPTR	80(50)	
YRSBOPEN	244(F4)	
YRSCLOGT	248(F8)	
YRSDBNAM	372(174)	
YRSDBNME	373(175)	
YRSDBNML	372(174)	
YRSDOPEN	244(F4)	
YRSDSCVP	16(10)	
YRSICTCH	0(0)	
YRSINAGT	192(C0)	
YRSLGBRC	304(130)	
YRSLRBMX	392(188)	
YRSLRBSC	388(184)	
YRSLRBSL	384(180)	
YRSOMBEG	364(16C)	
YRSOPSYS	352(160)	
YRSPATCH	12(C)	
YRSQDONE	244(F4)	(flag bit)
YRSRDCVP	20(14)	
YRSSCVTL	8(8)	
YRSSOPEN	244(F4)	(flag bit)
YRSSTART	257(101)	
YRSTRMAR	244(F4)	(flag bit)
YRSTRMFG	244(F4)	
YRSTRMIM	244(F4)	(flag bit)
YRSTRMQS	244(F4)	(flag bit)
YRSTRMST	244(F4)	(flag bit)
YRSVEXLP	76(4C)	
YRSVMCBD	72(48)	
YRSVMQBP	76(4C)	
YRSVMQVP	76(4C)	
YRSYTICH	24(18)	
YSCNTA	48(30)	
YSQSLEV	132(84)	
YTABA2	96(60)	
YTABA4	64(40)	
YTABNBB	172(AC)	
YTABNPB	168(A8)	
YTABPB	68(44)	
YT2EXTP	104(68)	
YT2POOLP	100(64)	
YT2SEGP	108(6C)	
YUCNTNAM	56(38)	

YUCNTVAL 60(3C)
YVSAMPTR 72(48)

Dec(Hex)	YRSSCVT	
0(0)	YRSICTCH - "YRSSCVT" (eyecatcher)	
8(8)	YRSSCVTL - YRSSCVT length	YRSPATCH - ↑ DBSS patch area
16(10)	YRSDSCVP - ↑ DS2CVT	YRSRDCVP - ↑ RDCVT
24(18)	YRSYT1CH - ↑ YTABLE1 chain	YCVTT1P - Address of YTABLE1 prototype
32(20)	YCVTT1S - Size of YTABLE1 without Stack	YMTAB1P - Address of Monitor's YTABLE1
40(28)	YCHKPTTP - "WHOAMI" of YCHKPT agent	PLIRSSP - ↑ ARIYM00 (PLIRSS)
48(30)	YSCNTA - Address of system counters table	YNBCNT - Numbers of counters YNBSCNT - System counters number | YNBUCNT - User counters number
56(38)	YUCNTNAM - Address of user counter names	YUCNTVAL - Address of user counter values
64(40)	YTABA4 - Address of YTABLE4 (buffer control information)	YTABPB - Address of page buffers
72(48)	YVSAMPTR - Address of VSAM declarations (for VSE) YRSVMCBD - Address of VMCBLOCK list for DDISK devices (for VM)	YRSVEXLP - ↑ VSAM exit list (for VSE) YRSVMQVP - Address of VMQ element table for I/O OPENS (VM)
80(50)	VSAMACBP - Address of BDISK ACB (for VSE) YRSBDPTR - Address of VMCBLOCK for BDISK device (for VM)	YDBCBP - Address of YDBCB (READ-only values)
88(58)	Address master record sections MASTERP - ↑ first part	SEGATP - ↑ DBSPACE attribute array
96(60)	Address of YTABLE2 sections YTABA2 - ↑ first part	YT2POOLP - ↑ Pool info array
104(68)	YT2EXTP - ↑ Extent array	YT2SEGP - ↑ DBSPACE array
112(70)	BLKALTP - Address of block alternation section	PBITMP - Address of summary bit map PBITMAP

Dec(Hex)	YRSSCVT (continued)	
120(78)	PBITMDP - Address of PBITMAPD	MODMAPP - Address of MODMAP partition
128(80)	MASTEREP - Address of last part of PMASTER record	YSOSLEV - Threshold percent for 'short-on-storage' warnings / Reserved
136(88)	NSYSBLK - Number of system blocks	OSEGBLK - Offset of DBSPACE blocks. First DBSPACE block is block number OSEGBLK+1
144(90)	OSYSBLK - Offset of system blocks	NMASTER - Master record number of blocks
152(98)	NPBMAP - Number of blocks to store PBITMAP	NBMAPBLK - Number of bit map blocks (per set)
160(A0)	NYTAB2 - Number of blocks to store YTABLE2	NBLKALT - Number of blocks to store block alternate vector
168(A8)	YTABNPB - Number of page buffers	YTABNBB - Number of block buffers
176(B0)	PGCTRSP - Address of PGCTRS	YDATASIZ - Length of DBSS shared data area
184(B8)	YENDBUFF - ↑ end of buffer pools	YRSACTAG - Number of agents still active
192(C0)	YRSINAGT - Number of in-doubt agents created	NCUSERS - Number of concurrent users
200(C8)	NTRANS - Number of concurrent user plus system	TPMAPP - ↑ transaction process map (see YTPMAP)
208(D0)	NEXTID - ID of next transaction to be started	MAXLOCKS - Maximum number of locks per transaction
216(D8)	NLATCHES - Number of S-latches in system	SLATCHP - ↑ array of S-latches
224(E0)	NNAMGATE - Number of hash table entries for named gates	NAMGATEP - ↑ name gate hash table
232(E8)	LOCWAITP - ↑ lock wait matrix	NLRBS - Number of lock request blocks

Dec(Hex) YRSSCVT (continued)

240(F0)	FREELISP - ↑ LRB free list			YRSTRMFG [1]	Reserved
248(F8)	YRSCLOGT - Cold log time stamp				
256(100)	LOGMODE [2]	YRSSTART [3]	LOGARCH [4]	LOGSTATE [5]	SLOGLTH - Number of bytes in log DBSPACE
264(108)	SLOGSEQN - Sequence number of next log page to be written				SLOGSCHK - Address of most recent system save record
272(110)	SLOGBEG - First used byte in log DBSPACE [6]				SLOGEND - First free byte in log DBSPACE
280(118)	SLOGARCH - First byte of log not yet archived				SLOGPAGE - Log page currently being written
288(120)	SLOGBUF - Address of buffer holding SLOGPAGE				SLOGWRIT - Next byte in log to be written to disk
296(128)	SLOGWBUF - Buffer holding page addressed by SLOGWBUF				SLOGCUSH - Number of pages for log cushion
304(130)	YRSLGBRC - Store log address of B checkpoint				LOGCLEAN - Zero or ID of process that is writing the good suffix of the log to disk (must be set with "Compare and Swap"). FFFF = crash and SYSTERR used by SYSREXIT.
312(138)	ARCHPCT - Log archive percentage				ARCHPTR - ↑ archive buffer
320(140)	ARCHDEF [7]	ARCUU - Character form of cuu (archive)			NPAGSCHK - Number of pages between checkpoints (in multi-partition/ multiple virtual machine mode)
328(148)	CHKCNTP - Count of pages remaining before a checkpoint (SYSMODE=M)				AUTOCNT - Number of LUWs between checkpoints (SYSMODE=S)
336(150)	DUALLOG [8]	CHECKPT [9]	ARBCUU - Binary form of cuu (archive)		NCSCANS - Number of scan SCB entries
344(158)	PROGNAME - Application program name				
352(160)	YRSOPSYS [11]	Reserved		DBPSLN - Length	DBPSWORD - Data base password field DBPSWD - Password (up to 8 characters) Length shown in DBPSLN

[1] See YRSSCVT Flags on page 239.
[2] N= No log; W= Write log
[3] SQL/DS start-up mode:
W= Perform normal SQL/DS start
C= Perform cold start
L= Reformat log
E= Add new data base DBEXTENT(s)
R= Restore data base from archive
S= Add new data base DBSPACE(s)

[4] Y= Archiving is enabled; don't let log wrap over unarchived records
N= No archiving; advance SLOGBEG at checkpoint to youngest alive begin work

[5] B'10000000'= error in read/write of primary log
B'01000000'= error in sec. log

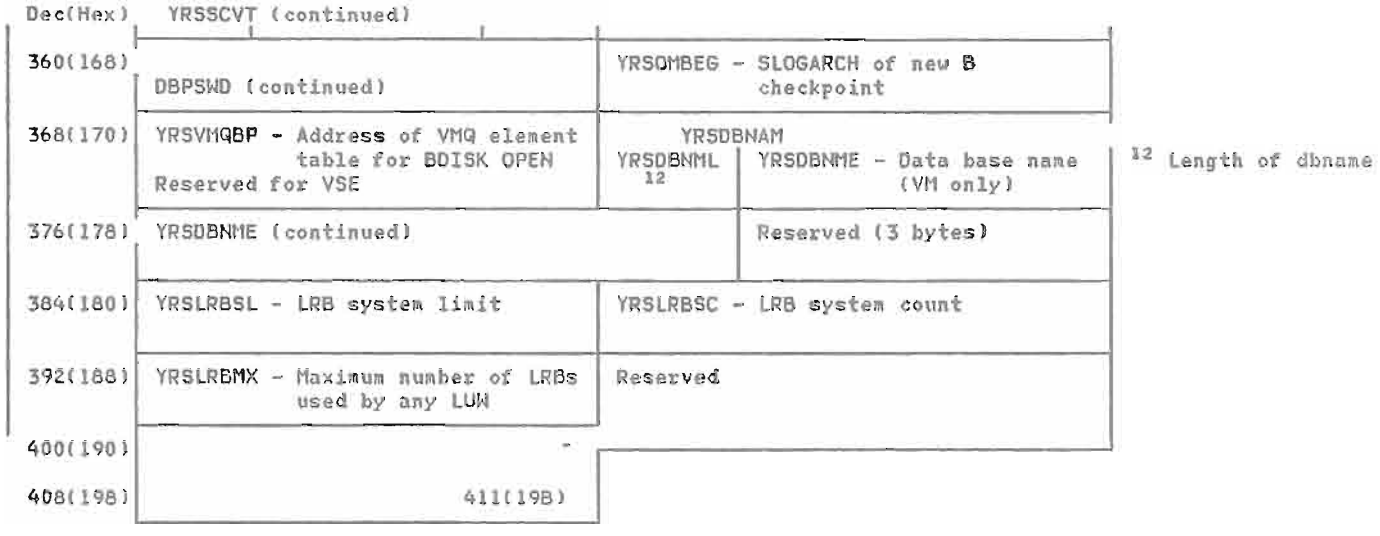
[6] The minimum of the address of the first record not archived, the address of the last save point, and the address of the begin work record of each active LUW. Bytes beyond this point are needed for archiving for online recovery

[7] Data base archive indicators (see page 239)

[8] N= No dual logging
Y= Dual logging

[9] Y= Checkpoint in progress

[11] Operating system indicators:
V for VM
D for VSE



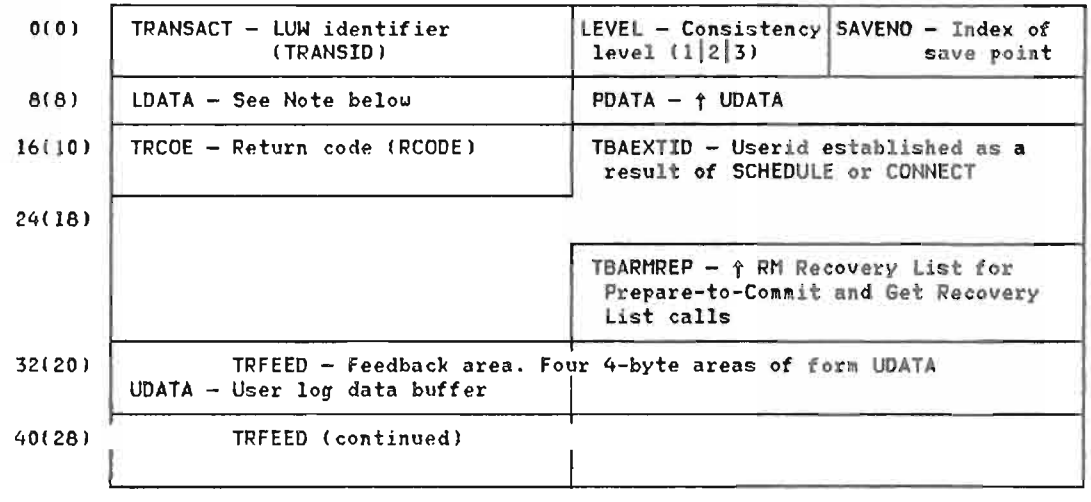
YRSSCVT Flags

Offset	Field Name	Bits	Meanings
320(140)	ARCHDEF		
	ARCHEX	1...	Log space or storage space is being exhausted.
	ARCHON	.1..	Archiving is taking place.
	ARCDBLBF	..1.	Double buffers exist for archiving.
	ARCHST	...1	Archiving posted.
	ARCHCOM	.... 1...	Archive requested by ARCHIVE command.
		xxx	Reserved.
244(F4)	YRSTRMFG		SQL/DS Quiescing Indicators:
	YRSTRMST	1...	SQL/DS shutdown started.
	YRSTRMIM	.1..	SQL/DS shutdown immediate.
	YRSTRMQS	..1.	SQL/DS shutdown quiesce.
	YRSTRMAR	...1	Quiesce with archive requested.
	YRSQDONE	.... 1...	SQL/DS quiesce completed.
	YRSOPEN	.... .1..	ACBs have been opened.
	YRSBOPEN	.... ..1.	B-disk has been opened.
	YRSDOPEN	1	D-disks have been opened.

YRSSTRAN (PARAMETER LIST FOR WORK COMPONENT DBSI CALLS)

The YRSSTRAN is the PLIST for any DBSI calls to the Work component. When the Work component receives a DBSI call, register 1 points to a field, which points to a parameter list, the first word of which points to YRSSTRAN. The pointer to YRSSTRAN is maintained in the RDATBASE field of RDAREA.

Dec(Hex) YRSSTRAN (TBASE)



Note: In general this is the length of user (RDS) data to log within a work-type LOGREC. Because RDS does not send user data, LDATA is set to zero by RDS. The exceptions to the general rule are:

- For CALL ARIYT14 (COMMIT) at the end of Catalog Generation, LDATA is -1. This forces a checkpoint before return to Catalog Generation.
- For CALL ARIYT06 (RESET), LDATA=0 means reset the storage pools if all areas in the secondary pools have been freed. Else call ARIYM00. LDATA=1 means call is from top of RDS - unconditionally reset all secondary storage pools.
- For CALL ARIYT28 (SCHEDULE), LDATA=0 means CONNECT; LDATA=1 means special SCHEDULE.

YTABLE1 (DBSSAREA)

YTABLE1 is an agent local control block used to drive the DBSS (and DSC) component of SQL/DS. The first YTABLE1 in the chain is pointed to by field YRSYT1CH of the YRSSCVT. Field YT1NEXT of a YTABLE1 points to the next YTABLE1 in the chain.

The YTABLE1 is pointed to by:

```
   DSCAREA (DSCYT1P,  offset X'18') ┐
   DCE     (DCEYT1P,  offset X'18') ┘ > 1
   RDAREA  (RDAYT1P,  offset X'1C') ┘
   YRSSCVT (YRSYT1CH, offset X'18') 2
   YTABLE1 (YT1NEXT,  offset X'20') 3
```

For the following notes, also refer to the diagrams on pages 2 and 3.

1 These pointers are only within an agent structure. That is, each of these control blocks is directly associated only with other control blocks within the same agent structure.

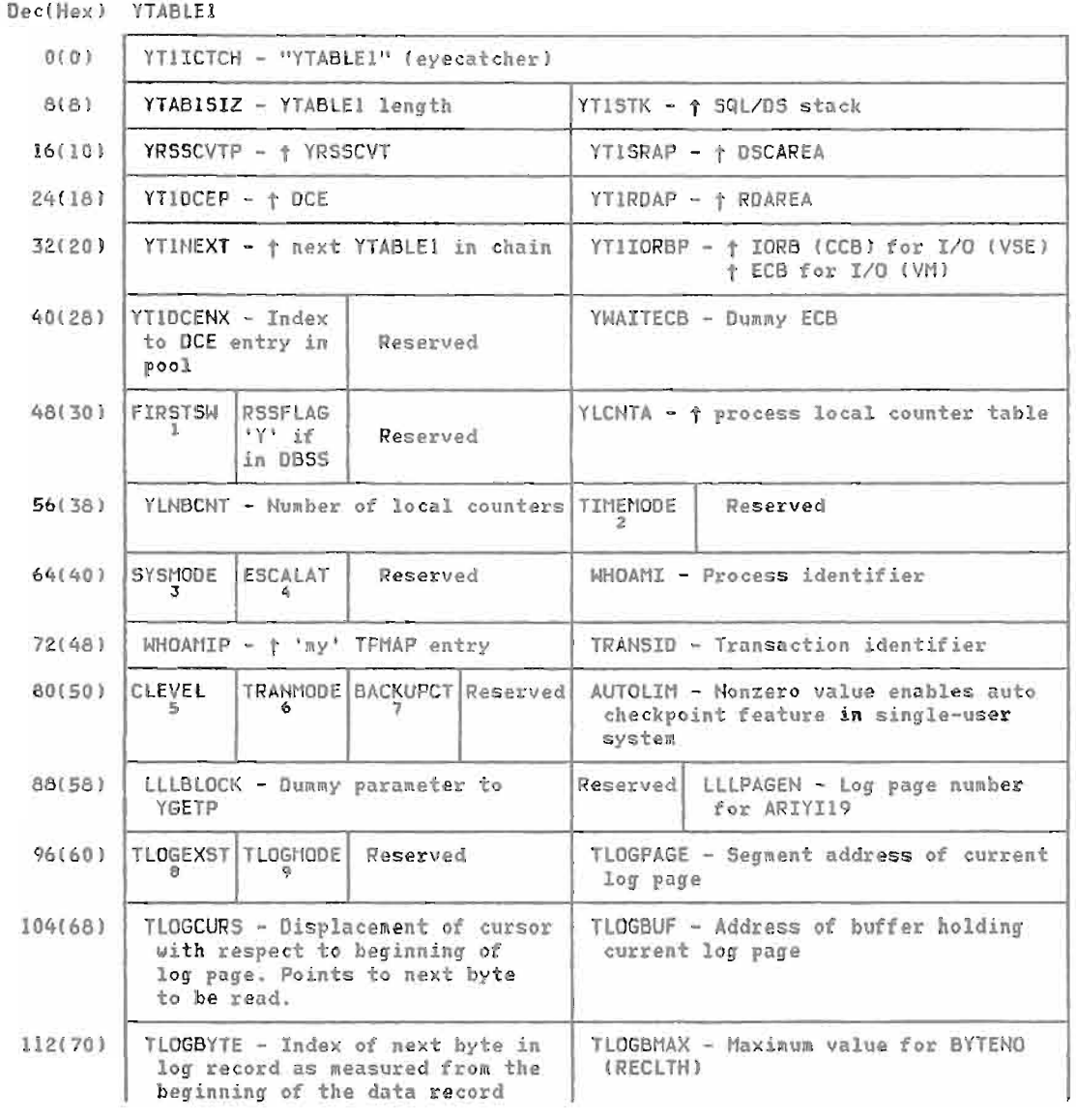
2 This points only to the first in the chain of YTABLEIs (that is, to the YTABLE1 for the Operator Agent structure).

3 This points to the next YTABLE1 in the chain (for the next agent structure).

Alphabetic List of Field Names

ATTNATTN	1189(4A5)	GLOCKREQ	155(9B)	ISCB	929(3A1)
ATTNMODE	1188(4A4)	GPAGE	1088(440)	ISCBMOVE	929(3A1) (flag bit)
AUTOLIM	84(54)	GSEGMENT	1086(43E)	ISCBSCAN	929(3A1) (flag bit)
BACKUPCT	82(52)	GTYPE	1084(43C)	ISCBUPD	929(3A1) (flag bit)
BUFFERS	308(134)	HBPTR	308(134)	ITID	930(3A2)
CALLNAME	244(F4)	HDRFLAG	372(174) (flag bit)	ITIDID	931(3A3)
CLEVEL	80(50)	HOLDIND	214(D6)	ITIDP	930(3A2)
DEBUGLV	1194(4AA)	HPPTR	312(138)	KPTR	196(C4)
DISPLKY	1078(436)	IBPTR	936(3A8)	LASTIID	374)176)
DISPLTO	1076(434)	IBUFF	936(3A8)	LASTLID	160(A0)
DOMPTR	296(128)	ICOMP	212(D4)	LASTPGLK	164(A4)
DPTR	200(C8)	IDISP	1006(3ED)	LASTRELN	158(9E)
DUPLELIM	1112(458)	IID	210(D2)	LASTREST	163(A3)
DURCODES	1092(444)	IIPC	1064(428)	LASTSEGN	156(9C)
EBPTR	344(158)	IKEY	408(198)	LASTSEST	162(A2)
ECLASS	362(16A)	IKEYGEN	667(19B)	LDMODE	1116(45C) (flag bit)
ELTH	360(168)	IKEYLTH	408(198)	LDRFLAGS	1116(45C)
ENTITY	360(168)	IKEYO	668(29C)	LDRMODE	1116(45C) (flag bit)
EPAGE	364(16C)	IKEYOLTH	668(29C)	LDRSCAN	1116(45C) (flag bit)
EPPTR	352(160)	IKEYOVAL	672(2A0)	LDRSTATE	1117(45D)
ESCALAT	65(41)	IKEYVAL	412(19C)	LID	208(D0)
ETIDID	367(16F)	ILOCK	927(39F)	LLLBLOCK	88(58)
ETIDS	364(16C)	IMPTR	300(12C)	LLLPAGEN	93(5D)
FASTFLAG	372(174) (flag bit)	IPAGES	1000(3E8)	LOGL	124(7C)
FDESCP	144(90)	IPFLAG	1000(3E8)	MODCBADR	136(88)
FEEDBACK	220(DC)	IPGFREE	1004(3EC)	MSRFLAGS	1124(464)
FIRSTSW	48(30)	IPN	1001(3E9)	MSRSTATE	1125(465)
GATE	1084(43C)	IPPTR	968(3C8)	NKEYDOM	190(BE)
GATEID	1088(440)	IRETCODE	1080(438)	NORDSPEC	1106(452)
GID	1091(443)	ISCANID	928(3A0)	NREQDOM	192(C0)

```
NSARGS     194(C2)
PAGECODE   1092(444)
PRESIID    934(3A6)
PRID       182(B6)
PSEGMENT   180(B4)
PTID       184(B8)
QUALF      215(D7)
RCODE      216(D8)
RCODFEED   216(D8)
RELN       150(96)
RELSTATE   153(99)
RID        174(AE)
RSSFLAG    49(31)
SCANFLAG   1075(433)
SCANID     188(BC)
SCANPAGE   1072(430)
SCBPTR     304(130)
SEGMATTR   154(9A)
SEGMENT    172(AC)
SEGN       148(94)
SEGSTATE   152(98)
SORTOPTR   1108(454)
SORTSEGM   1104(450)
SPTR       204(CC)
SVIXPAGE   1065(429)
SYSMODE    64(40)
TEMPHWM    1114(45A)
TID        176(B0)
TIDCODE    1096(448)
TIDID      179(B3)
TIDP       176(B0)
TIMEMODE   60(3C)
TLOGBMAX   116(74)
TLOGBUF    108(6C)
TLOGBYTE   112(70)
TLOGCURS   104(68)
TLOGEXST   96(60)
TLOGMODE   97(61)
TLOGPAGE   100(64)
TRACESW    1192(4A8)
TRANMODE   81(51)
TRANSID    76(4C)
UNLDMODE   1116(45C) (flag bit)
USERLTH    120(78)
VNUM       1068(42C)
VSAMRPLB   128(80)
WHOAMI     68(44)
WHOAMIP    72(48)
YBASE      172(AC)
YDCRFLAG   372(174) (flag bit)
YDEGREE    272(110)
YDOMREC    1128(468)
YEMAXPN    257(101)
YFLAGS     372(174)
YHMAXPN    252(FC)
YICRFLAG   372(174) (flag bit)
YIMAGREC   376(178)
YLCHILDP   285(11D)
YLCNEXT    294(126)
YLCNTA     52(34)
YLCODE     284(11C)
YLCTIDP    292(124)
YLINKREC   284(11C)
YLNBCNT    56(38)
YLPREFIX   271(10F)
YLPREL     290(122)
YLPSEG     288(120)
YLPTIDP    293(125)
YLRECL     286(11E)
YMASTREC   266(10A)
YMAXPN     253(FD)
YMDOMN     276(114)
YMIMN      278(116)
YMLINKCN   280(118)
YMLINKPN   282(11A)
YMRCODE    270(10E)
YMRTYPE    266(10A)
YNCLINKS   274(112)
YNIMAGES   273(111)
YNPLINKS   275(113)
YNRELS     260(104)
YNRELS2    261(105)
YNXTPREV   372(174) (flag bit)
YP         240(F0)
YRELNEXT   268(10C)
YRELTYPE   267(10B)
YRSSCVTP   16(10)
YSFLAGS    249(F9)
YSFREE     250(FA)
YSHEADER   248(F8)
YSNULL     251(FB)
YSRELN     262(106)
YSRELN2    264(108)
YSRTYPE    248(F8)
YTAB1SIZ   8(8)
YTCMPCOD   1082(43A)
YT1BUPTR   128(80)
YT1DCENX   40(28)
YT1DCEP    24(18)
YT1ICTCH   0(0)
YT1INDEX   408(198)
YT1IORBP   36(24)
YT1LRBMH   1220(4C4)
YT1MDCB1   140(8C)
YT1NEXT    32(20)
YT1OPCOD   1196(4AC)
YT1RDAP    28(1C)
YT1RESCD   1216(4C0)
YT1SCANP   236(E4)
YT1SRAP    20(14)
YT1STK     12(C)
YT1TRAC    1200(4B0)
YT1TRACA   1200(4B0)
YT1TRACB   1201(4B1)
YT1TRACC   1205(4B5)
YT1TRACD   1206(4B6)
YT1TRACI   1207(4B7)
YT1TRACK   1203(4B3)
YT1TRACL   1202(4B2)
YT1TRACS   1208(4B8)
YT1TRACT   1204(4B4)
YT1TRACX   1209(4B9)
YT1TRACZ   1210(4BA)
YT1TRCSP   1212(4BC)
YT1TRMAR   929(3A1)  (flag bit)
YT1VMCBU   132(84)
YVSAMRP    132(84)
YWAITECB   44(2C)
```



1 0 = TPMAP slot not assigned
1 = TPMAP assigned

2 'xxxxxx00'B = No timing
'xxxxxxx1'B = DBSI timing
'xxxxxx1x'B = Subsystem timing

3 M = Multiple User mode
S = Single User mode

4 Y = Try to escalate locks

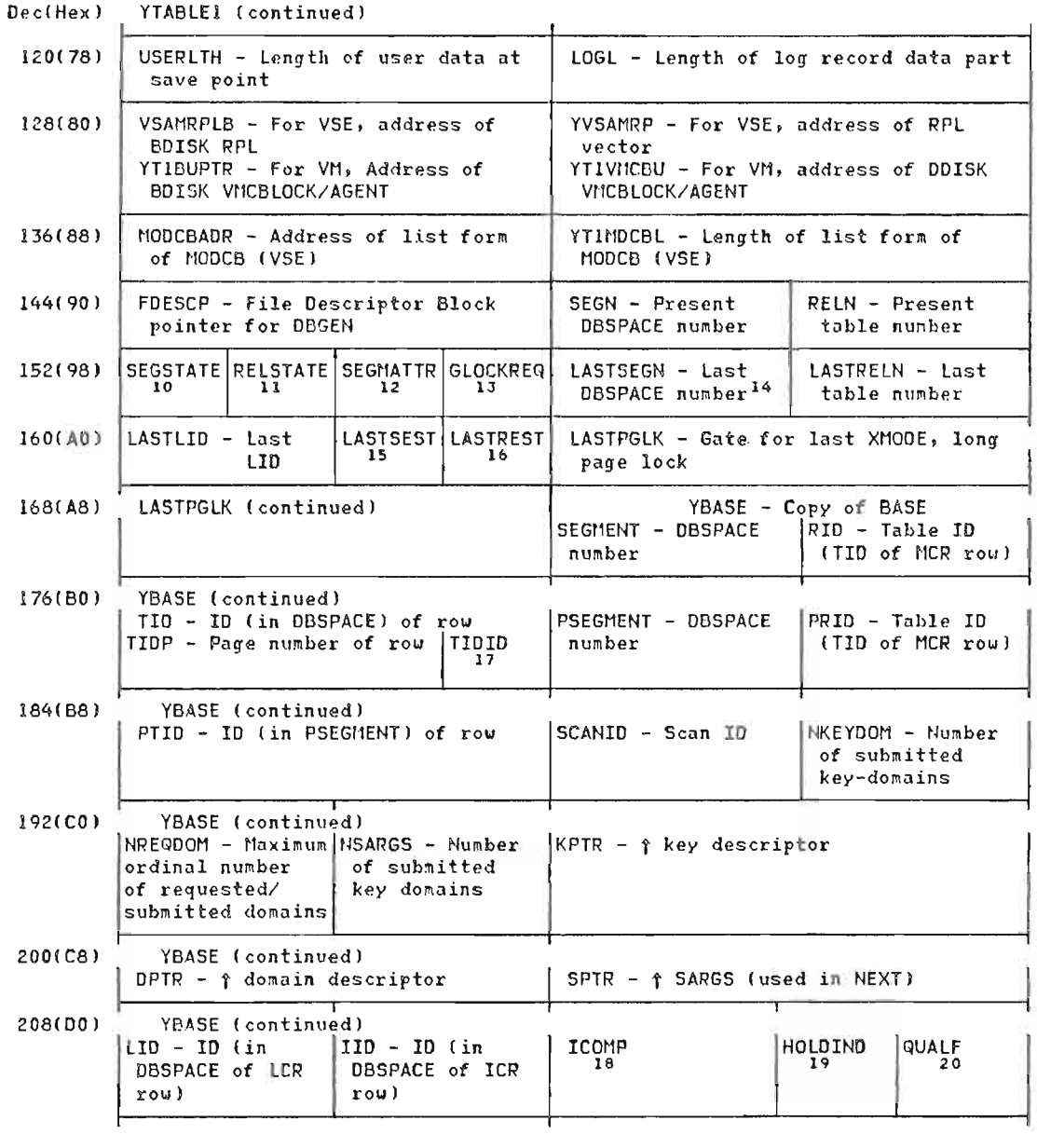
5 Consistency level: 1, 2, or 3

6 LUW mode:
F = Forward operation
B = Backward operation
R = System re-do
U = System un-do
N = LUW not started

7 Count of SI-backups executed in DBSI-call

8 N = No logging (yet) for LUW
Y = LUW logging has occurred

9 LUW current log access mode:
N = Neither YSTARTLG nor YSTARTRD
R = Reading YREADLOG, YMOVECUR, YSETCUR, YENDREAD
W = Writing YWRITELG, YENDWRIT



10 Present DBSPACE lock state

11 Present table lock state

12 DBSPACE attributes:

Bit	Value	Meaning
0	0	Public
1	0	Synchronous
2	0	Recoverable
3	0	Permanent
4	1	Open
5	0	Read-only
6	1	Available

13 Gross lock requirement

14 Set to 0 by ARIYT09 and during un-do and re-do

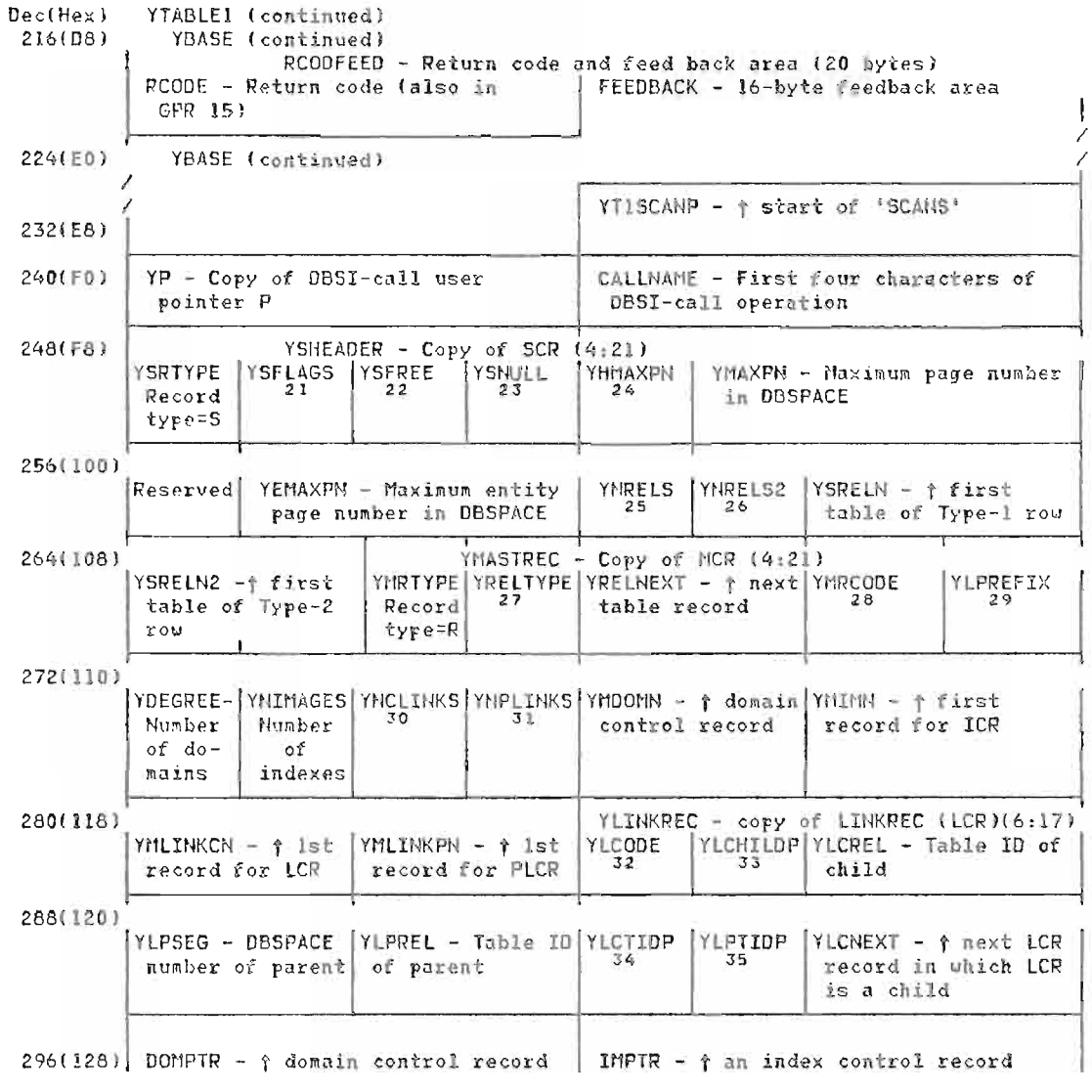
15 Last DBSPACE lock state

16 Last table lock state

17 ID number of row within page

18 Comparison operator using index or Type-2 table: M=Match, A=After, MA=Match/After, B=Before, MB=Match/Before, FI=First

19 Lock Hold indicator: H=Hold



requested; Any other character = otherwise

20 Call qualifier used in OPEN, NEXT, CONNECT: R=table, I=index, C=link on a child, P=link on a parent, B=Before, H=Here, A=After (B and A are used in CONNECT, H and A are used in NEXT)

21 DBSPACE flags:

Name	Bit	Value	Meaning
YTLOCK	0	1	TID locking
YLISTFLG	1	1	List DBSPACE
YSGLOCK	2	1	DBSPACE lock
	3-7		Unused

22 Percentage of free entity space at load

23 Character for null column in row

24 Max page number (3rd byte) in DBSPACE header

25 Number of Type-1 tables in DBSPACE

26 Number of Type-2 tables in DBSPACE

27 Table type: Bit 0=0 if Type-1

28 Table ordinal (stored in rows)

29 Size (bytes) of link prefix in row

30 Number of links in which table is child

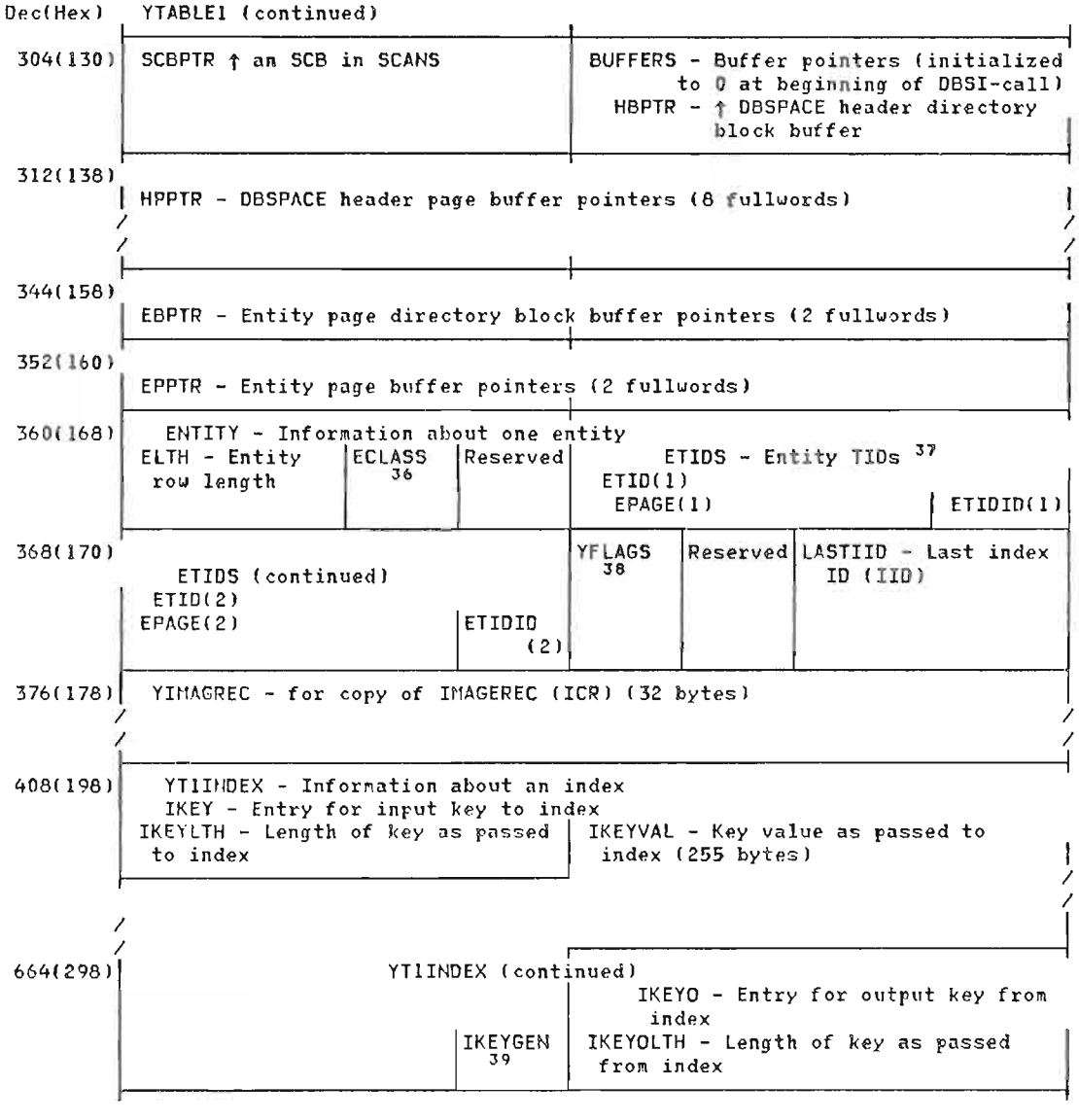
31 Number of links in which table is parent

32 Ordinal number of link

33 Length of child prefix

34 Index of first TID pointer in child vector of TID pointers. TID pointers are consecutive in this order: right twin, left twin, parent.

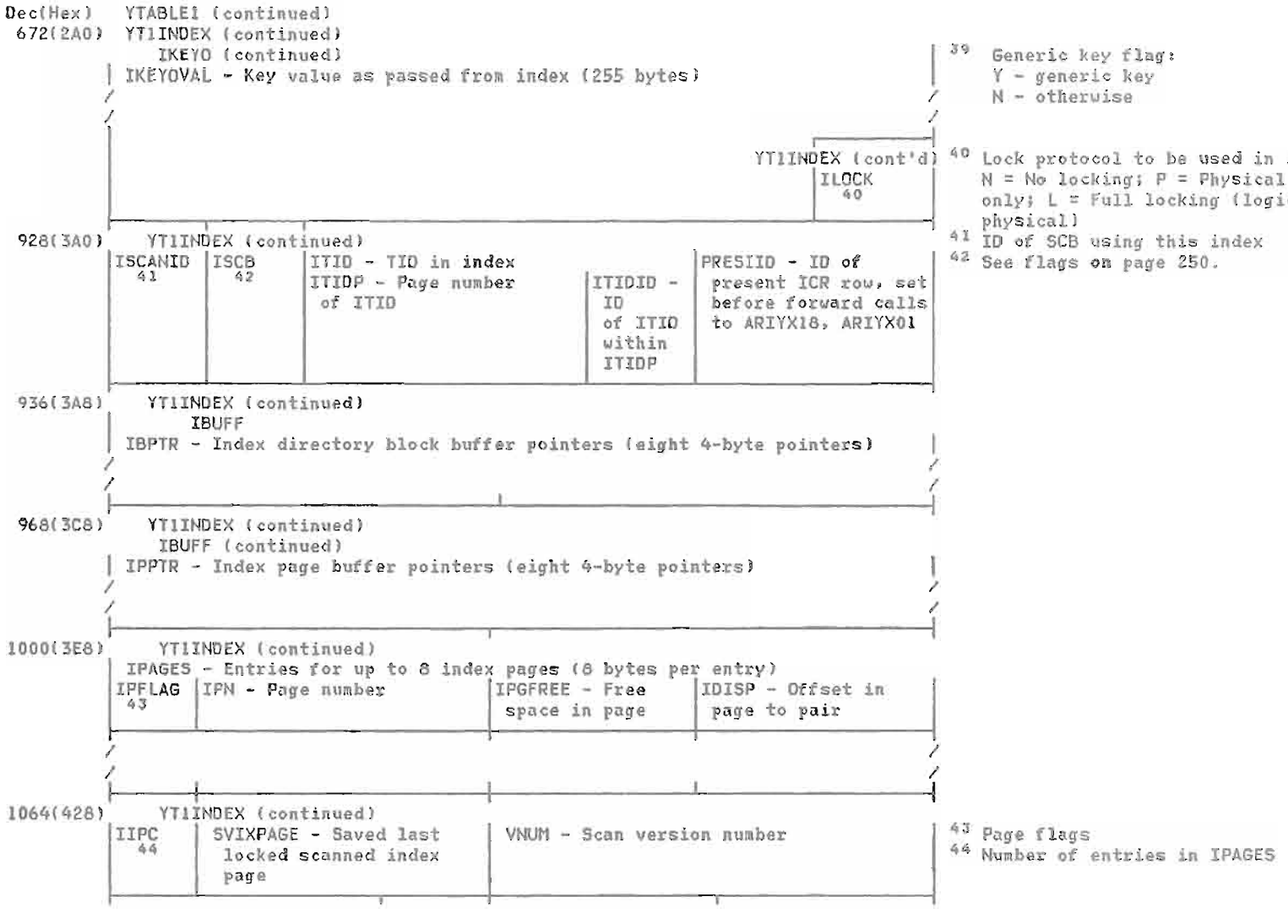
35 Index of first-child TID pointer in parent vector of TID pointers.

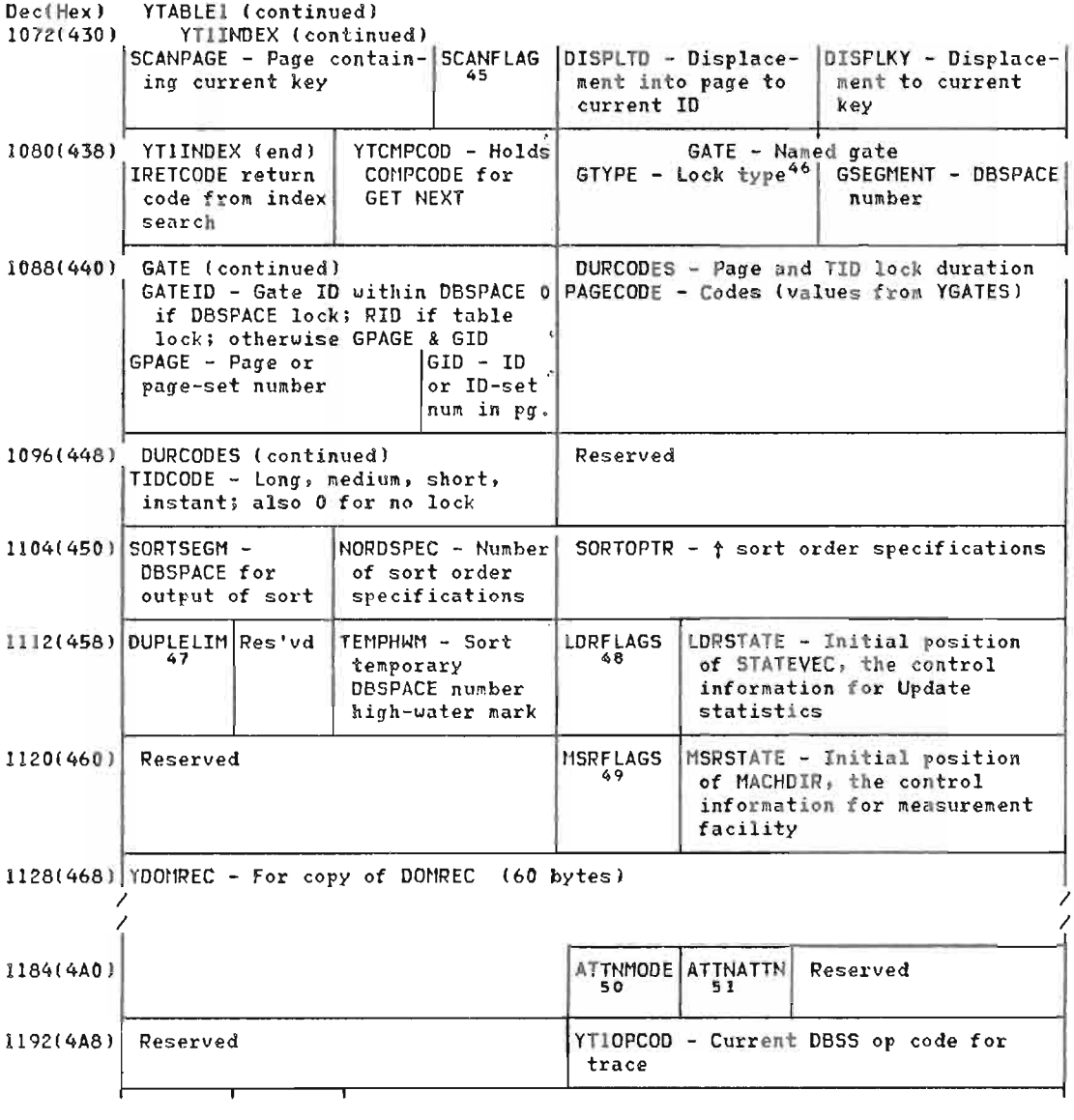


36 Entity page free space class (1 to 13)

37 Room for 2 TIDs of the form:
EPAGE 3 bytes for page for TID
ETIDID 1 byte ID of row within entity page

38 See flags on page 250.


Dec(Hex)
YTABLE1 (continued)
672(2A0)
YT1INDEX (continued)
IKEYO (continued)
IKEYOVAL - Key value as passed from index (255 bytes)
39 Generic key flag:
Y - generic key
N - otherwise
YT1INDEX (cont'd)
ILOCK
40
40 Lock protocol to be used in index:
N = No locking; P = Physical locking only; L = Full locking (logical + physical)
928(3A0)
YT1INDEX (continued)
ISCANID
41
ISCB
42
ITID - TID in index
ITIDP - Page number of ITID
ITIDID - ID of ITID within ITIDP
PRESIID - ID of present ICR row, set before forward calls to ARIYX18, ARIYX01
41 ID of SCB using this index
42 See flags on page 250.
936(3A8)
YT1INDEX (continued)
IBUFF
IBPTR - Index directory block buffer pointers (eight 4-byte pointers)
968(3C8)
YT1INDEX (continued)
IBUFF (continued)
IPPTR - Index page buffer pointers (eight 4-byte pointers)
1000(3E8)
YT1INDEX (continued)
IPAGES - Entries for up to 8 index pages (8 bytes per entry)
IPFLAG
43
IPN - Page number
IPGFREE - Free space in page
IDISP - Offset in page to pair
1064(428)
YT1INDEX (continued)
IIPC
44
SVIXPAGE - Saved last locked scanned index page
VNUM - Scan version number
43 Page flags
44 Number of entries in IPAGES



45 SCB flags:

Bit	Value	Meaning
0	0	SCB not "dirty"
1	1	Physical information bad
2	1	State is "BEFORE"
3	1	State is "EOF"

46 See flags on page 250.

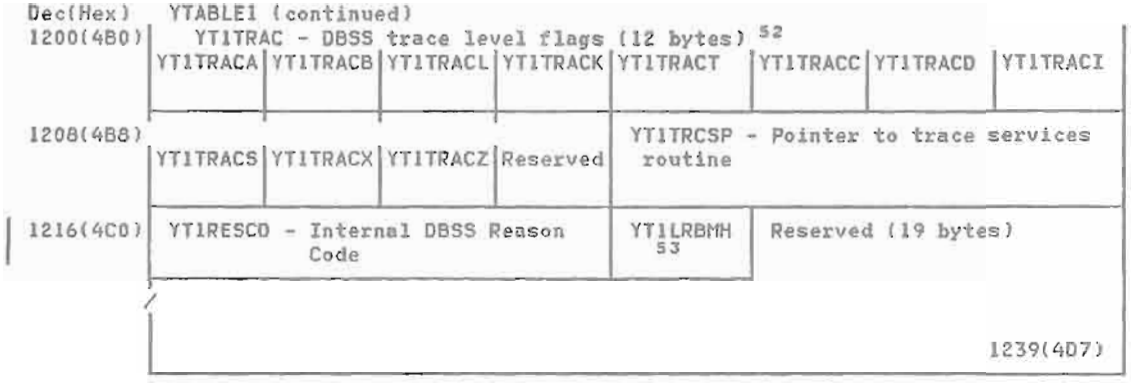
47 Option for duplicate elimination in sort

48 See flags on page 250.

49 MSRMODE: Bit 0 = 1 if measurement is on
Bits 1-7: Unused

50 'Y' = Attention handling enabled

51 User attention flag



52 YTITRACA - DBSS entry trace level
YTITRACB - DBSS exit trace level
YTITRACL - Log/Recovery subcomp trace level
YTITRACK - Lock mgmt subcomp trace level
YTITRACT - LUW mgmt subcomp trace level
YTITRACC - Data Control subcomp trace level
YTITRACD - Data Manipulation subcomp trace level
YTITRACI - Storage (I/O) subcomp trace level
YTITRACS - Sort subcomp trace level
YTITRACX - Index subcomp trace level
YTITRACZ - Update Statistics subcomp trace level
* - Reserved

53 LRB must have ARIYK22 set this to 'Y' prior to calling ARIYK41 for a lock, and reset it to 'N' after the call. When the value is 'Y', ARIYK41 allocates a lock even if the LRB system limit has been reached.

YTABLE1 Flags

Offset	Field Name	Bits	Meanings
372(174)	YFLAGS		Flags for Fast Path:
	YDOMREC	1...	YDOMREC has domain information.
	YIMAGREC	.1..	YIMAGREC has index information.
	FASTFLAG	..1.	Can do FAST NEXT.
	HDRFLAG	...1	Header page was accessed.
	YNXTPREV	.... 1...	NEXT or PREVIOUS call was made.
		xxx	Reserved.
929(3A1)	ISCB		Information on updates of index SCBs.
	ISCBMOVE	0...	SCBs to be moved if affected (forward DBSI-CALLs). Input to index.
	ISCBUPD	.0..	No SCB was moved in DBSI-CALL. Set by Index component in WRITE calls.
	ISCBSCAN	..0.	If no index scans
	*	...x xxxx	Reserved.

1084(43C) GTYPE			Lock Type:
First Byte:		0...	Entity or Index lock.
		1...	DBSPACE or Table lock..
		.0..	Interval (TID or Key) lock.
		.1..	Page-set lock.
		..0.	Entity lock.
		..1.	Index lock.
		...0	To be unlocked in BACKUP.
		...1	Not to be unlocked in BACKUP
			BACKUP (asynchronous DBSPACE).
		.... .1..	Special lock (for example, Data Base lock, BACKUP lock).
		.... ..1.	Temporary DBSPACE.
		.... x..x	Unused.
Second Byte:		xxxx xxxx	Unused.
1116(45C) LDRFLAGS			X'00'; no Update statistics activity. Any positional change of its substructure should be reflected in LDRMASK in DBSS.
	LDMODE	1...	Begin Update statistics activity
	UNLDMODE	.1..	End Update statistics activity
	LDRSCAN	..1.	ARIYZ11 does DBSPACE scan.
		..0.	ARIYZ11 does table scan.
	LDRMODE	...1	Update statistics facilities available.
		 xxxx	Reserved.

YTABLE1S

YTABLE1S is a truncated version of YTABLE1 (the agent local control block used to drive the DBSS (and DSC) component) for modules that reference only the first few fields of YTABLE1.

Dec(Hex) YTABLE1

0(0)	YT1ICTCH - "YTABLE1" (eyecatcher)			
8(8)	YTAB1SIZ - YTABLE1 length		YT1STK - ↑ SQL/DS stack	
16(10)	YRSSCVTP - ↑ DBSSCVT (YRSSCVT)		YT1SRAP - ↑ DSCAREA	
24(18)	YT1DCEP - ↑ DCE		YT1RDAP - ↑ Agent RDAREA	
32(20)	YT1NEXT - ↑ next YTABLE1 in chain		YT1IORBP - VM: ↑ ECB for I/O VSE: ↑ IORB (CCB) for I/O	
40(28)	YT1DCENX - Index to DCE entry in pool	Reserved	YWAITECB - Dummy ECB	
48(30)	FIRSTSW [1]	RSSFLAG 'Y' if in DBSS	Reserved	YLCNTA - ↑ process local counter table
56(38)	YLNBCNT - Number of local counters		TIMEMODE [2]	Reserved
64(40)	SYSMODE [3]	ESCALAT [4]	Reserved	WHOAMI - Process identifier
72(48)	WHOAMIP - ↑ 'my' TPMAP entry 75(4B)			

[1] 0 = TPMAP slot not assigned
1 = TPMAP assigned

[2] 'xxxxxx00'B = no timing
'xxxxxxx1'B = DBSI timing
'xxxxxx1x'B = Subsystem timing

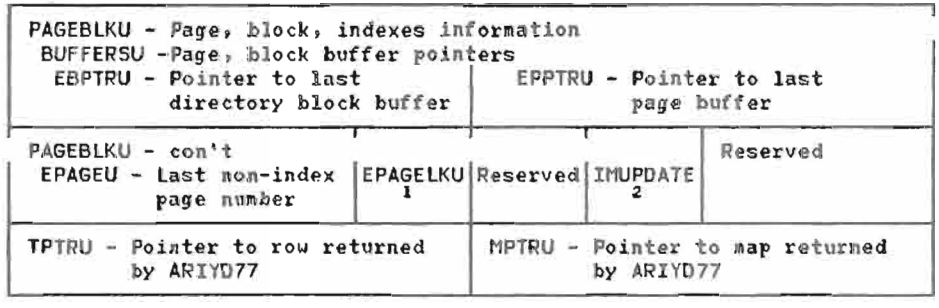
[3] M = Multiple User mode
S = Single User mode

[4] Try to escalate locks

YTABLE1U

YTABLE1U is an overlay to an area normally used by YBASE in YTABLE1. It is used during recovery operations. YTABLE1U describes the last non-index page/directory block that has been updated. It describes the lock protocol used in the recovery call.

Dec(Hex) YTABLE1U

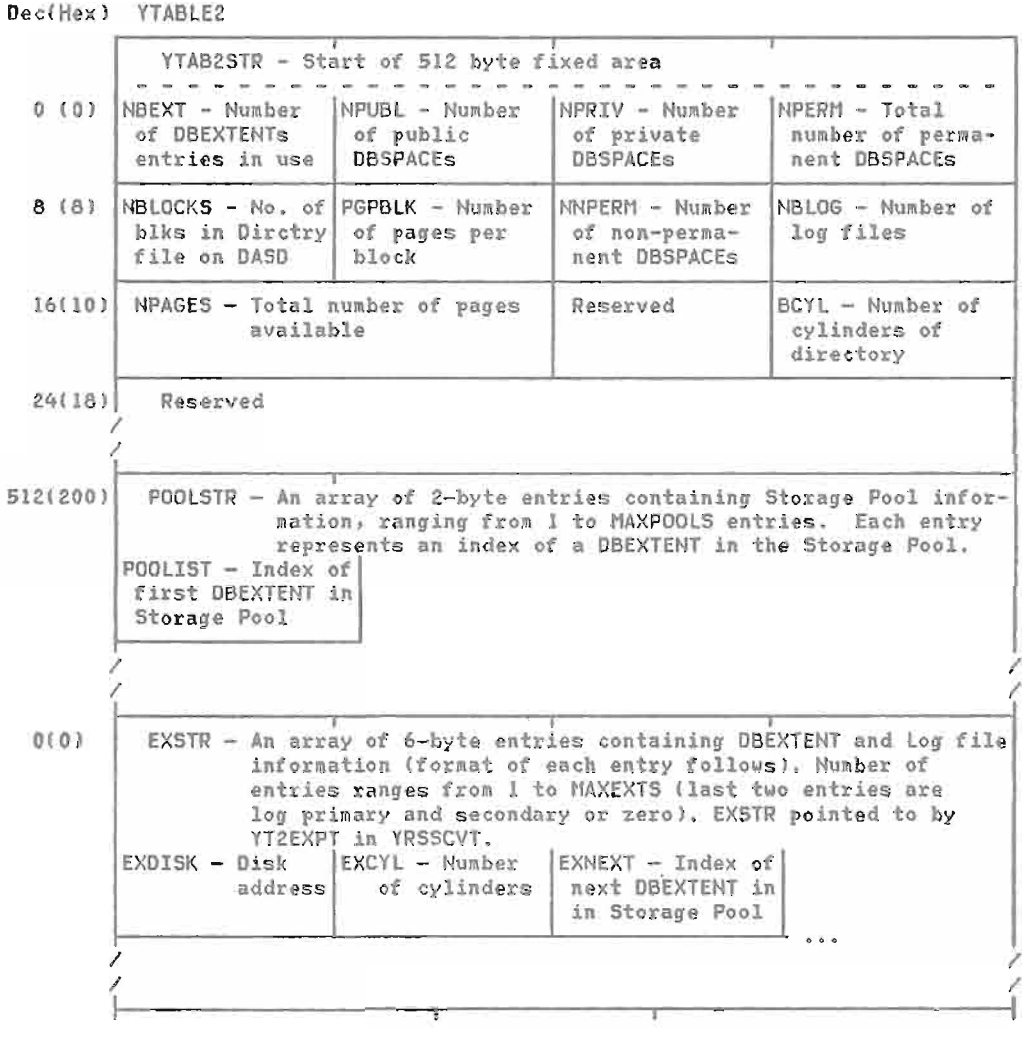


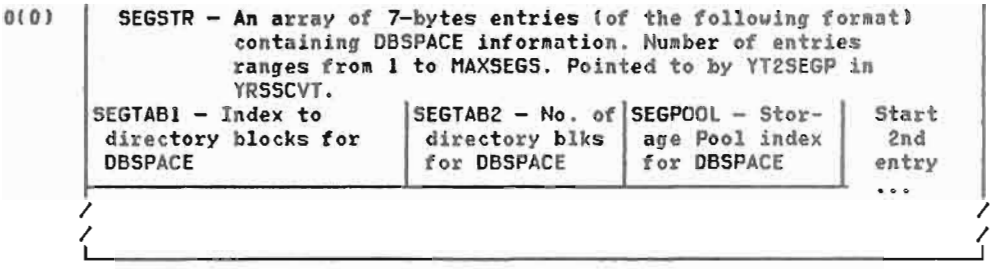
1 'P' if and only if an entity page lock is needed.
2 'Y' if and only if any index needs to be updated.

YTABLE2

YTABLE2 contains the Pool Table, Extent Table, DBSPACE Table, and some system variables. Two versions of YTABLE2 are stored in the Directory file on DASD. YT2BYTE in MASTER points to the current version.

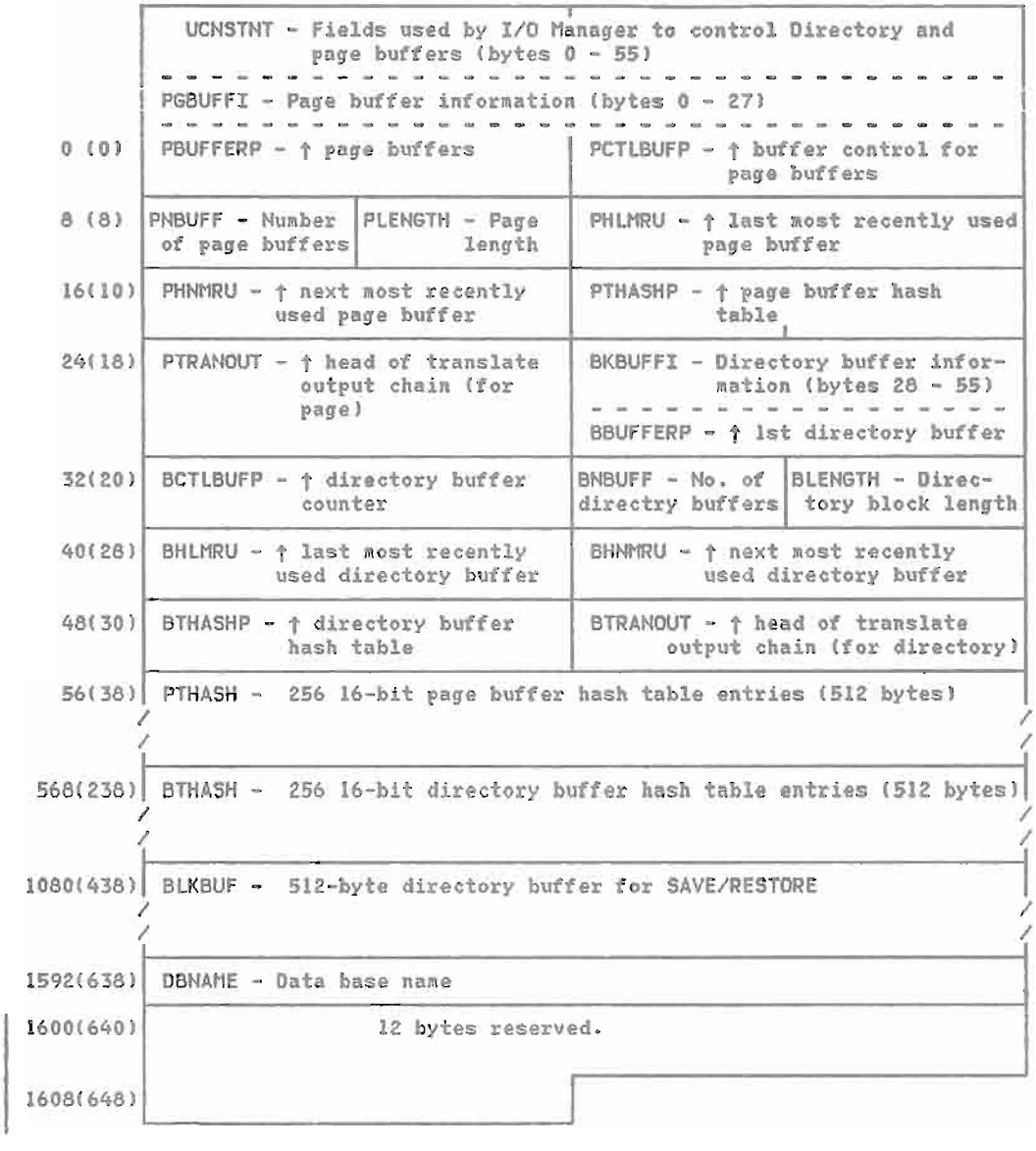
Portions of YTABLE2 vary in size according to the values of MAXPOOL, MAXSEG and MAXBLKS. Addressability to the various sections is set during DBSS initialization. The anchor pointers are in YRSSCVT (YTABA2 points to YTABLE2, YT2POOLP points to POOLSTR, YT2EXTP points to EXSTR, and YT2SEGP points to SEGSTR). All sections are fetched/stored together from/to the Directory file.





YTABLE4

YTABLE4 contains current information on buffers. It is kept in virtual storage only, not on Directory file. It is pointed to by field YTABA4 of the YRSSCVT.

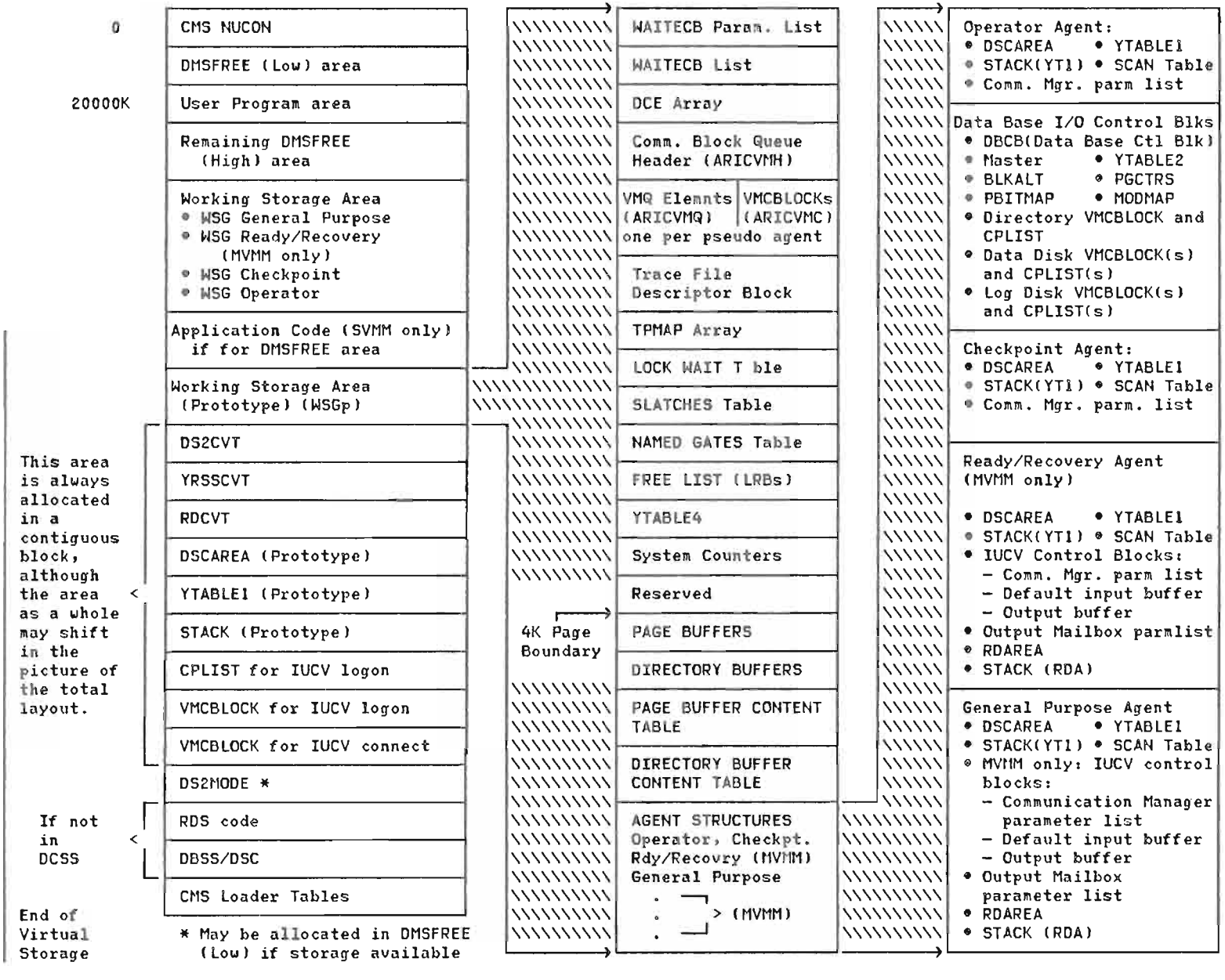


SECTION 6: DIAGNOSTIC AIDS

This section contains the following:

STORAGE LAYOUT AFTER INITIALIZATION - VSE

STORAGE LAYOUT AFTER INITIALIZATION - VM

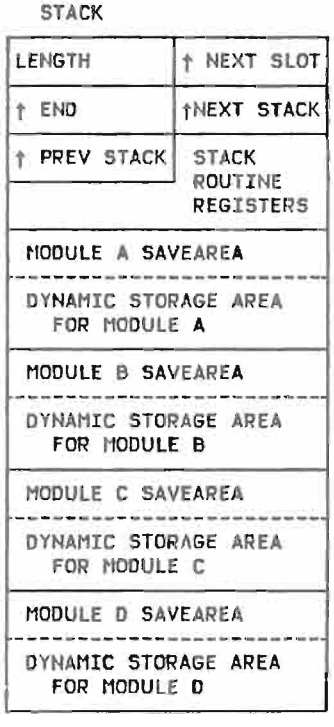


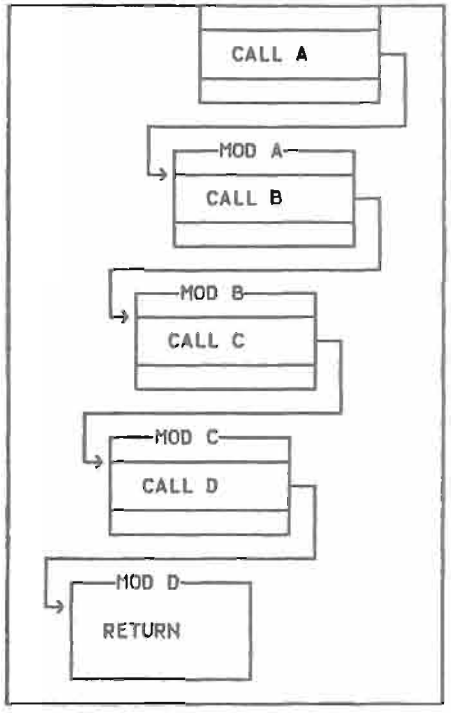
HOW TO FIND THE MODULE'S DYNAMIC STORAGE AREA (SAVE AREAS)

It is assumed that coding has been done using normal register conventions. It is also assumed that register 10 points to YTABLE1/RDAREA/RMLO. (Note: This does not apply for ISQL or Extract.) Due to the structure of the SQL/DS system-dependent routine (ARISYSD), it is recommended that you use the registers displayed in the SQL/DS "mini-dump".

```
R10 = † YTABLE1 / RDAREA / RMLO
      (DBSS/DSC)  (RDS)    (RM)

R13 = † SAVEAREA - at beginning of the
                   stack area allocated
                   to that module.
```

```
R10 = († YTABLE1) + 12(X'0C')     =    († STACK)
      († RDAREA) + 12(X'0C')
      († RMLO) + 12(X'0C')
```

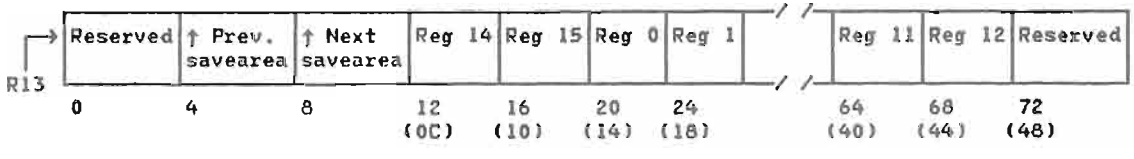




MODULE ENTRY:
Regs 14→12 will be saved in caller's savearea + 12.
Reg 13 is pointing to the savearea.

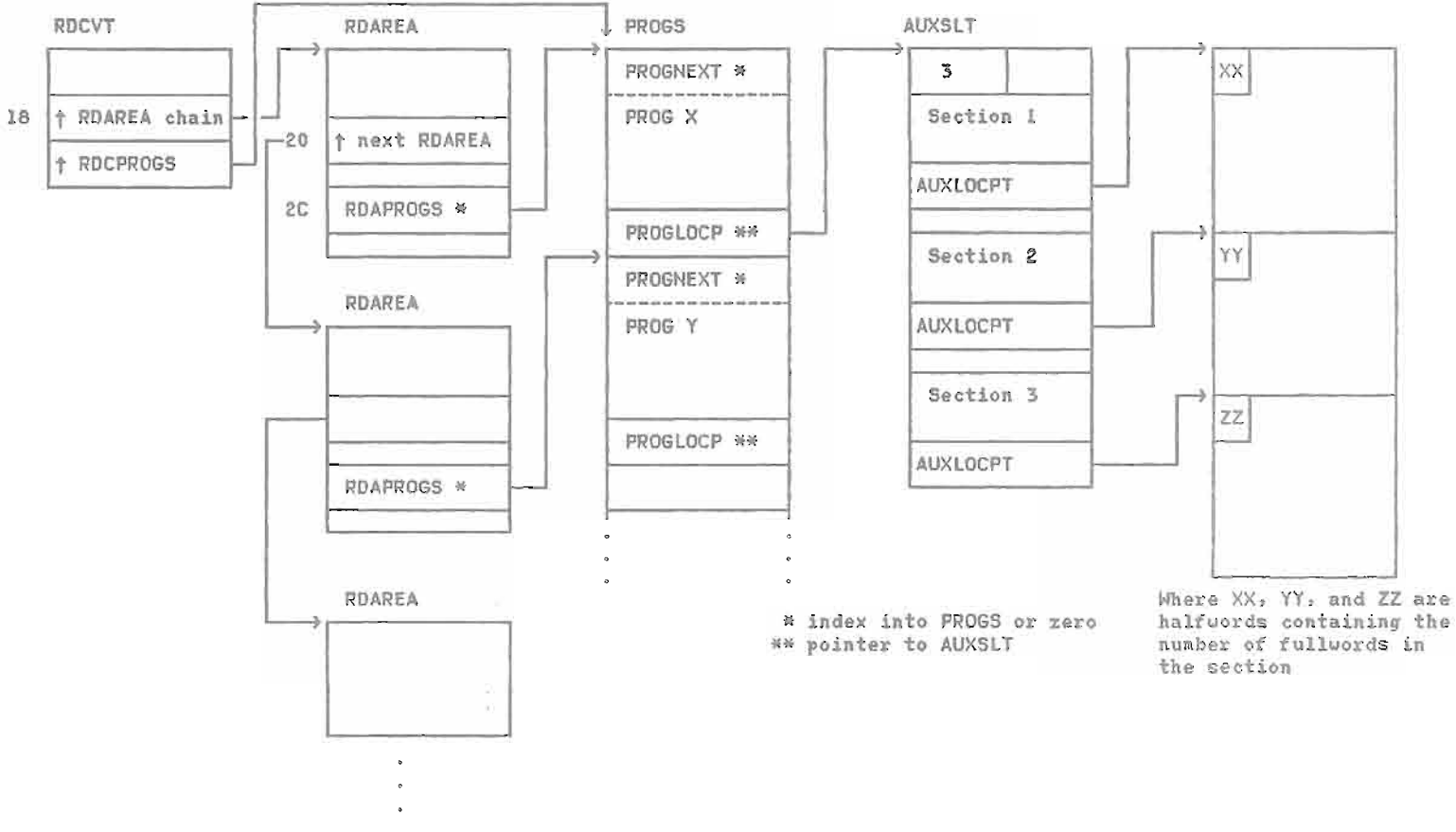
Reg 1 = Pointer to parameter list.

Note: Dynamic Storage is the storage that is required only during the invocation of a given module; that is, it is acquired from stack storage when the module is called (this is done in the module's prolog), and it is returned to stack storage when the module returns to its caller. When a module is invoked, it stores its caller's registers in the save area pointed to by register 13. It acquires stack storage, and the first 18 words are normally used as the save area (pointed to by register 13) for the "called" module.



HOW TO FIND AMOUNT OF GETVIS/DMSFREE STORAGE USED FOR ACCESS MODULES

Note: Offsets are in hexadecimal.

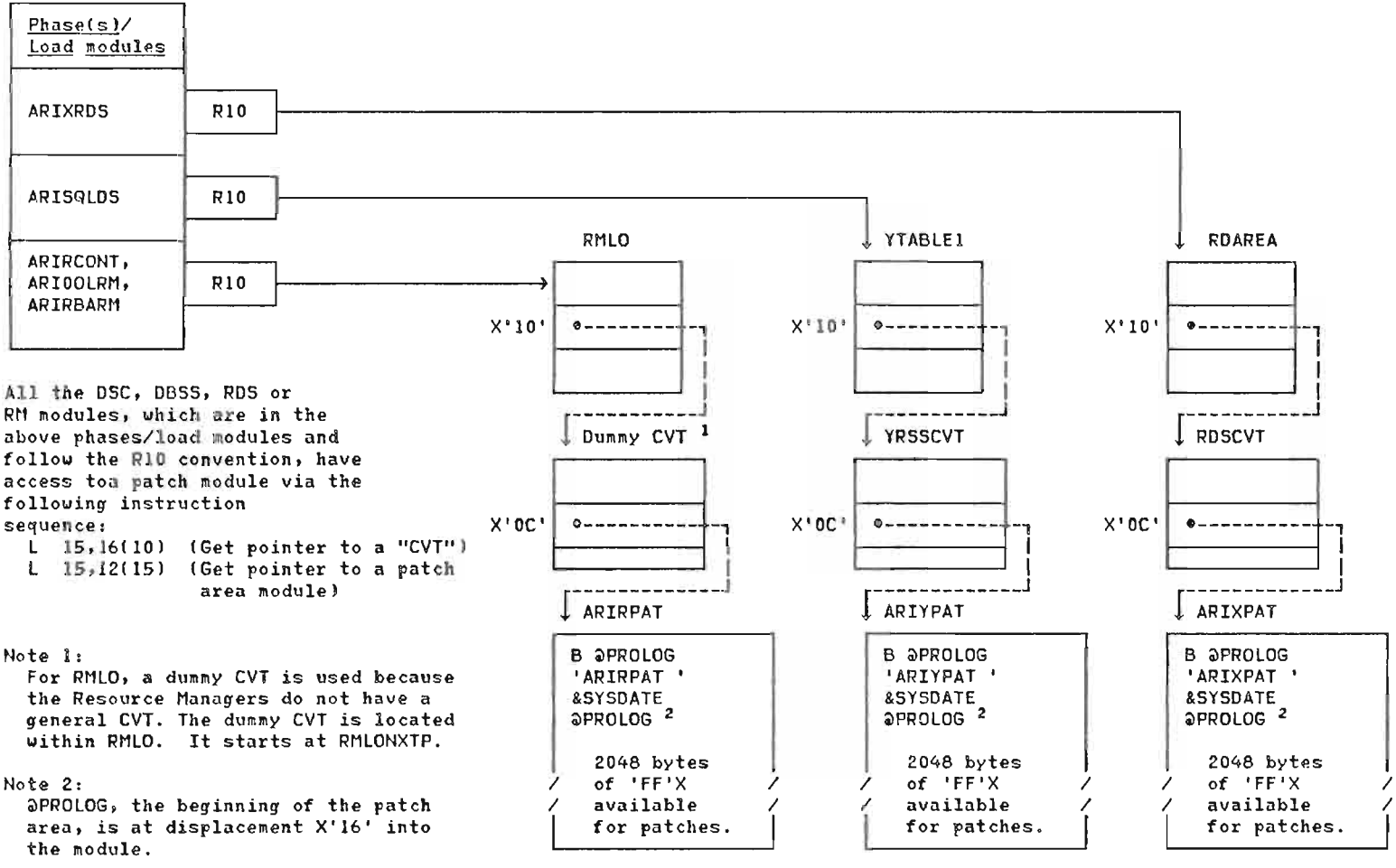


To find the amount of GETVIS/DMSFREE storage used by access modules:

1. Find the RDCVT.
2. Get pointer to the first/next RDAREA on the chain.
3. For each RDAREA, find the PROGS list. (Use index into PROGS (RDAPROGS), start of PROGS is in RDCPROGS.)
4. For each entry in the PROGS list, find the pointer to AUXSLT (PROGLOCP).
5. The AUXSLT is in GETVIS/DMSFREE storage. Size = (entries * 28) + 4.
6. For each entry in AUXSLT, get pointer to section.
7. The first halfword of each section contains the number of fullwords in the section, which is in GETVIS/DMSFREE storage. Size = (fullwords * 4).
8. Total GETVIS/DMSFREE storage allocated for access modules is sum of Steps 5 and 7 for each entry in PROGS for each RDAREA on chain.

SQL/DS PATCH AREA MODULES AND ACCESS

COMMON PATCH AREAS



All the DSC, DBSS, RDS or RM modules, which are in the above phases/load modules and follow the R10 convention, have access toa patch module via the following instruction sequence:

```
L   15,16(10)   (Get pointer to a "CVT")
L   15,12(15)   (Get pointer to a patch
                 area module)
```

Note 1:
For RMLO, a dummy CVT is used because the Resource Managers do not have a general CVT. The dummy CVT is located within RMLO. It starts at RMLONXTP.

Note 2:
@PROLOG, the beginning of the patch area, is at displacement X'16' into the module.

MODULES HAVING OWN PATCH AREAS

The following kinds of modules contain their own patch areas:

- Large modules (>4K bytes): 3%
- Complex modules: 5% or 50 bytes (minimum)
- Error-prone modules: 5% or 50 bytes (minimum)

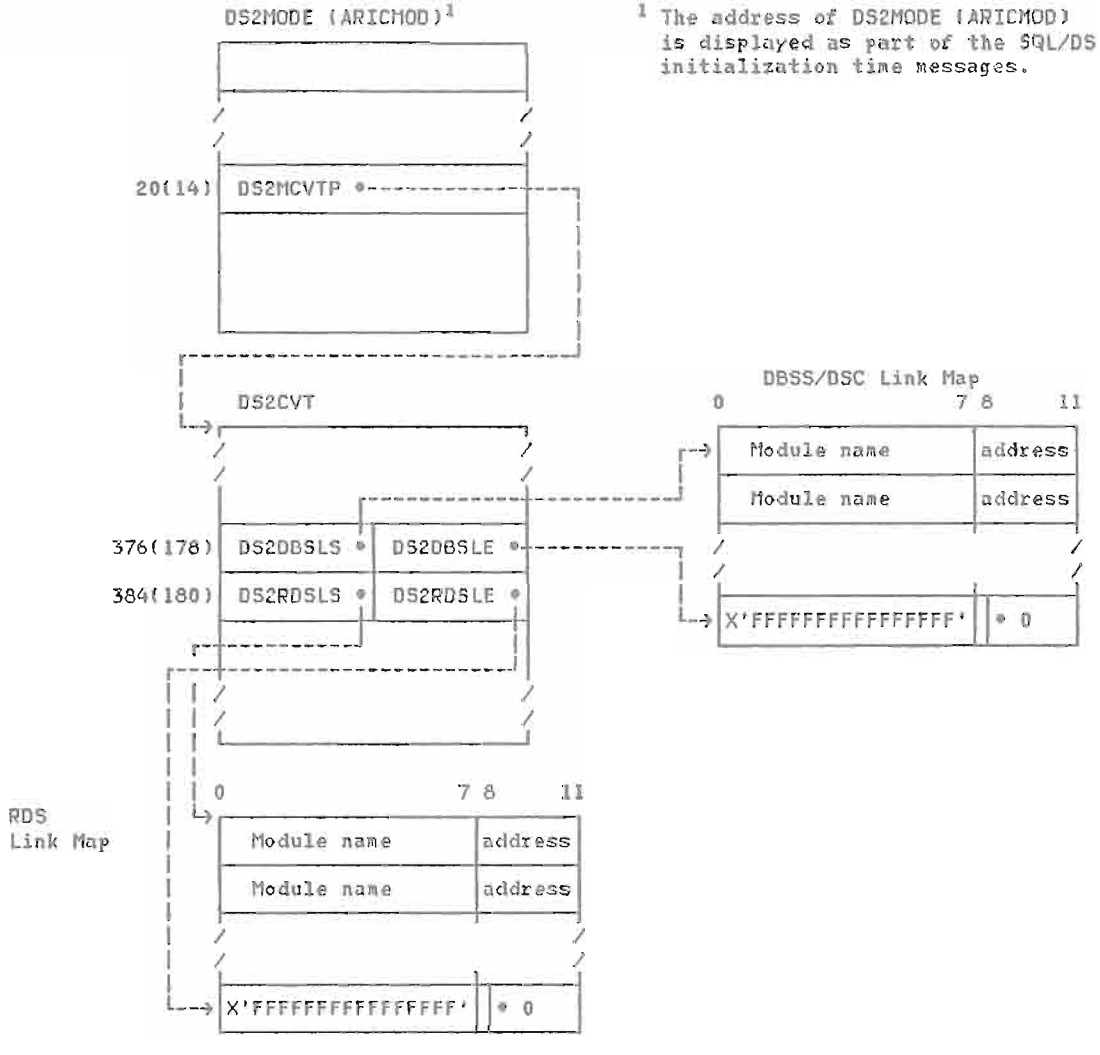
@PSPACE is the patch area label for each of these.

SQL/DS LINK MAPS AND ACCESS

The last module in the DBSS/DSC and RDS load modules contain the name and address of each module in the load module. When DUMPTYPE=P is specified (default), the link maps are the last two areas dumped. When DUMPTYPE=F is specified the link maps are at the end of each load module or phase.

Modules are ordered by their address (in the same order as the link edit book entries) allowing a given module to be located quickly via the address in register 15 in the save area chain.

See also module descriptions of ARIXLNK (RDS) and ARIYLNK (DBSS) in Volume 1.



[1] The address of DS2MODE (ARICMOD) is displayed as part of the SQL/DS initialization time messages.

Register 10 - Pointer to YTABLE1/RDAREA/RMLO (Bytes 0 - 7 contain "YTABLE1 ", or "RDAREA ", or "RMLO " as eyecatchers)

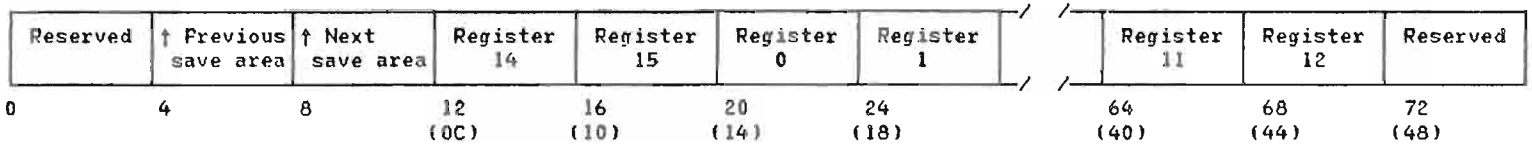
Register 13 - Pointer to save area (in stack) except for the Dispatcher, in which case it points to a DCE. (See the layout of the DCE in Section 5, Data Areas. Also see the discussion in Volume 1 under "Agent Handling and Communications" of the DSC portion of Section 2. The register 9 save area slot in the DCE contains the pointer to the save area of the dispatcher's caller, either ARICDWT or ARICWAT.)

Generally:
  Register 12 is used as a base register
  Register 11 is used as a data register
  Register 11 = Register 13
Due to the structure of the SQL/DS system-dependent routine (ARISYSD), it is recommended that you use the registers displayed in the SQL/DS "mini-dump".

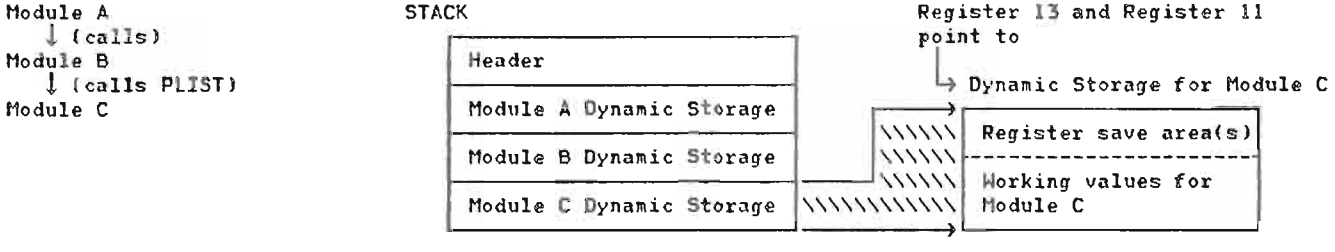
The save area is the start of the module's dynamic storage. Dynamic storage is found in the stack.

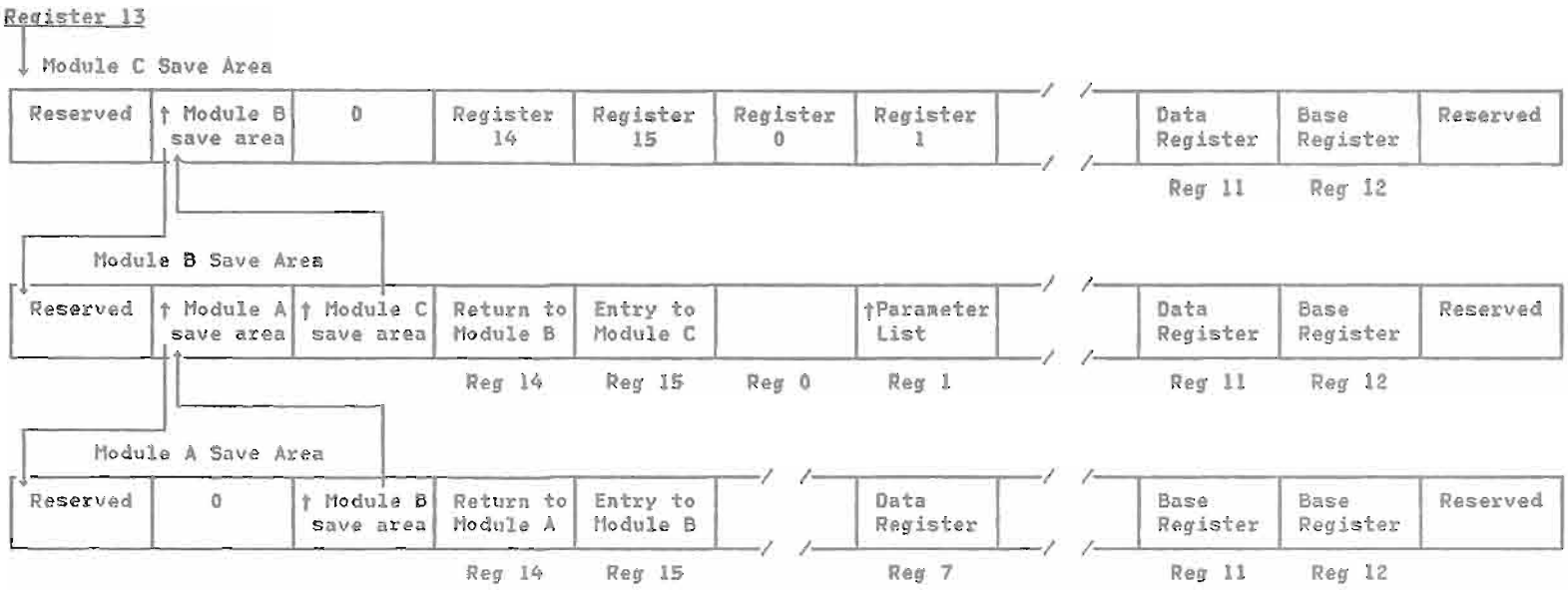
If register 10 points to a YTABLE1/RDAREA/RMLO, then register 10 + X'0C' points to STACK.

Register 13

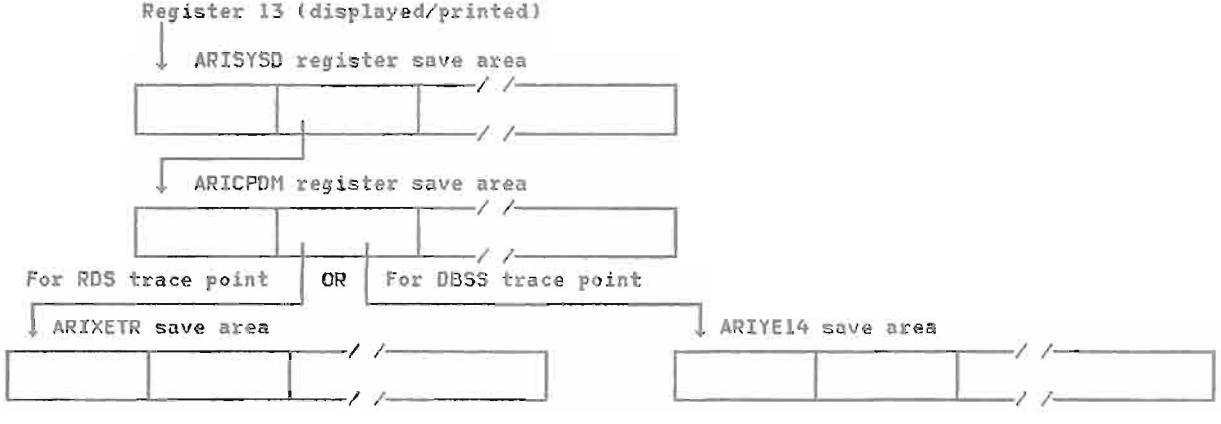


Assume a program check occurred in Module C, but Module A is the source of the problem. The dynamic storage for Module A can be found by "back tracking" through the save area chain (Register 13 is a pointer to the Module C save area).





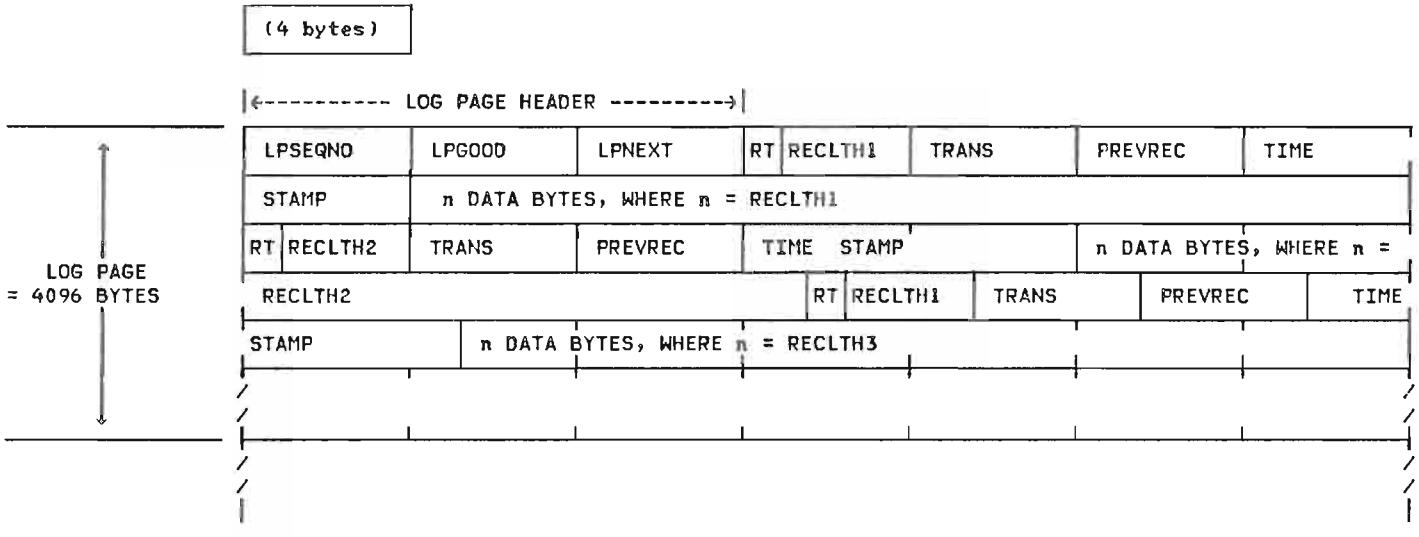
For dumps resulting from the DUMP option of the TRACE command, use the standard method of following register save areas to locate the general register values of the module containing the trace point that caused the snap dump. (For information relating to the Trace Facility, see the section "SQL/DS Trace Facility" starting on page 382.) The method is illustrated below:



The ARIXETR or ARIYE14 save area contains register values of the module containing the trace point (their values at the point of the trace call BALR).

LOG PAGE AND RECORD FORMAT

The following diagram shows the format of the log pages and records. Note that a record may overflow from one page to the next (that is, a single log record need not be entirely contained within the bounds of a single page).



Where:

LPSEQNO	4 BYTES	LOG PAGE SEQUENCE NUMBER
LPGOOD	4 BYTES	NUMBER OF VALID BYTES IN PAGE
LPNEXT	4 BYTES	POINTER TO NEXT LOG PAGE IN STORAGE
RT (RECTYPE)	1 BYTE	TYPE OF LOG RECORD:[1] 0: NOT YET WRITTEN 1-16: UPDATE-TYPE RECORDS 20: BEGIN LUW 22: COMMIT LUW 23: ABORT LUW 24: SYSTEM CHECKPOINT 25: PREPARE-TO-COMMIT
RECLTH	4 BYTES	LENGTH OF RECORD (EXCLUSIVE OF HEADER) IN BYTES.
TRANS	4 BYTES	LUW IDENTIFIER.
PREVREC	4 BYTES	DISPLACEMENT (IN BYTES) TO PREVIOUS LOG RECORD OF THIS LUW.
TIME STAMP	8 BYTES	TIME WHEN RECORD WAS STARTED.

Note 1: For further information on log record types, see the control blocks shown under "Log Pages and Records" and "Log Data for a DBSI Call as Passed to/from the Log Linkage" in the Data Areas section.

DBSS OP CODES

The OP codes are located in field:
YTIOPCPD (in YTABLEI) - It contains the current (or last) DBSS OPCODE.

Op Code	Meaning
LUW Codes	
1	Begin work
2	Commit work
3	Save work state (for rollback)
4	Rollback work to save point
Lock Codes	
6	Lock a DBSPACE, table, or row
7	Unlock a DBSPACE, table, or row
Data Manipulation Codes	
8	Close a scan
9	Connect a row in a link
10	Delete a row
11	Disconnect a row from a link
12	Fetch a row
13	Insert a row
14	Get the next row
15	Open a scan
16	Get the parent row in a link
17	Update a row
DBSPACE Codes	
22	Acquire a non-permanent DBSPACE
Data Control Codes	
23	Get a control record
24	Get the next control record
25	Insert the control record
26	Delete the control record
27	Update the control record
Initialization/Termination Codes	
28	Initialize DBSS within this process
30	Terminate DBSS within this process
Additional DBSPACE Codes	
31	Release a non-permanent DBSPACE
Operator Command Code	
32	Process the remote operator command
More LUW Codes	
33	Retrieve in-doubt list
34	Prepare to commit LUW
Sort Code	
35	Sort an object
More LUW Codes	
36	Schedule userid into DBSS and DSC
Update Statistics Codes	
37	Start update statistics
41	End update statistics

DBSS RETURN CODES

The return codes are found in Register 15 and also in parameters passed by the calling modules. The return codes are divided into three groups:

1) 0 - (successful) Op executed and no unusual conditions occurred.
2) positive - (warning) Op executed but one or more unusual condition occurred.
3) negative - (bad) Op not executed, data base not changed, or the logical unit of work has been backed up.

Warning Return Codes

Return Code	Name	Meaning
1	DIRTSCAN	Used in read call. For Fetch Row, Open Scan, or Get Parent, scan is 'on' a row but flagged dirty. For Get Next Row, scan is flagged dirty and may or may not be 'on' a row. Get Next Row resets the scan to 'clean' and resets the scan to 'on' (or 'EOF').
2	DOMTRUNC	Field truncation. On a read request, a requested domain is longer than the domain FREQLTH value and thus the returned domain has low order truncation.
4	EOF	End of file on a scanned set
4	TOF	Top of file on a scanned set. No operation currently sets this code.
8	KEYVIOL	Key violation on Get Next Row for an index scan. Submitted key lower than the current key as indicated by SCAN and QUALF. Scan is positioned 'on' first offending row.
16	NFORMSEG	DBSPACE requested by CFETCH, CNEXT exists but has no SCR record or was not opened
64	FORCEND	Automatic commit work is generated on deletion of DBSPACE, master or link control records
128	BADSARG	Record does not satisfy the search arguments

Return Code	Name	Meaning
4096	INCONSEG	NSEG or SEGMS(*) is inconsistent with the REL(*) supplied to 'STARTUS' call.

Bad Return Codes

Bad DBSS Call Parameters

Return Code	Name	Meaning
-1	ILOPCODE	Illegal OPCODE on DBSS call
-2	ILLEGALP	Illegal parameter (PARAMS) pointer on DBSS call (0, or not word/doubleword aligned).

Illegal parameters for BASE. Some codes also used by CBASE, SBASE.

Return Code	Name	Meaning
-3	ISEGMENT	Illegal DBSPACE number, missing SCR, or type conflict with operator
-4	ILLRID	Illegal table ID or illegal operation for list
-5	NFTID	TID not found or open of an empty table or list
-6	ILLPSEGM	Illegal parent DBSPACE number
-7	ILLPRID	Illegal PRID or illegal operation for list
-8	NFPTID	PTID not found
-9	ILSCANID	Illegal scan ID
-10	INKEYDOM	Illegal number of key fields
-11	ILLKPTR	Illegal address of KDOMAINS
-12	INREQDOM	Illegal number of DOMAINS fields
-13	ILLDPTR	Illegal address of DOMAINS
-14	INSARGS	Illegal number of search arguments
-15	ILLSPTR	Illegal address of search arguments
-16	ILLEGLID	Illegal link ID
-17	ILLEGIID	Illegal index ID
-18	ILLICOMP	Illegal comparison operator (ICOMP)
-19	ILLQUALF	Illegal call qualifier in BASE (QUALF); illegal call qualifier in CBASE (CQUALF)

Illegal Parameters in KDOMAINS

Return Code	Name	Meaning
-20	IKDOMPTR	Illegal key-field address
-21	IKDOMLTH	Illegal key-field length

Illegal Parameters in DOMAINS

Return Code	Name	Meaning
-22	IFLDPTR	Illegal field address
-23	IFACLTH	Illegal submitted field length

Illegal Parameters in Search Arguments

Return Code	Name	Meaning
-24	ISNUMBER	Illegal field number in search arguments
-25	ISPOINT	Illegal field address in search arguments
-26	ISLENGTH	Illegal field length in search arguments
-27	ISCOMP	Illegal comparison operator in search arguments
-28	ISBOP	Illegal boolean operator in search arguments

Exceptional Conditions on a Scan

Return Code	Name	Meaning
-29	EMPSCAN	Empty scan in SCANID
-30	SCANOW	Too many scans are opened by RDS
-31	UPDIRTY	Write call using a dirty scan
-32	NOTON	Call different from NEXT, CLOSE using a scan which is not 'on'

Exceptional Conditions on an Index or on a Row

Return Code	Name	Meaning
-33	NFKEY	Submitted key not found
-34	UNIQKEY	Violation of index key-uniqueness
-35	LONGKEY	Submitted key is too long
-36	LONGTPLE	Submitted row is too long
-37	ILLTYPE2	Illegal call for a Type 2 table. (Should never occur as SQL/DS does not have Type 2 tables)

Exceptional Conditions on a Link

Return Code	Name	Meaning
-38	CLINKED	Delete of a row illegally disconnecting some link; also connection of a row in a link in which it is already connected
-39	CNLINKED	Disconnect, or delete and disconnect of a non-linked child; link open on a non-linked child in a binary link; connect, insert and connect to a non-linked child in a binary link
-40	NOPARENT	Parent does not exist (in PARENT call)
-41	NOCHILD	Child does not exist (in OPEN call)

Illegal Parameters in CBASE and Auxiliary Control Structures

Return Code	Name	Meaning
-45	IQUALFSG	Illegal QUALFSEG
-46	IAUXPTR	Illegal auxiliary pointer

Return Code	Name	Meaning
-47	INPAGES	Illegal NIPAGES or NHPAGES
-48	ILLFREE	Illegal percentage of free space in SCR or ICR
-49	ILOCKPRT	Illegal lock protocol in SCR or ICR
-50	ILLNDOM	Illegal number of field entries in MCR or ICR
-51	ILLDOMS	Illegal field entries (lengths in MCR, field numbers in ICR)
-52	ILUNIQUE	Illegal unique key specification
-53	TOOMANYC	Too many control objects as a result of a CINSERT
-54	LISTSEGM	Operation illegal on a list DBSPACE. Only scan operations, Insert Row, and Sort are legal.
		Illegal Conditions Related to LUW Management
-60	NOSTART	Begin Work was not issued prior to a Save Work call.
-61	NOTEND	COMMIT or ROLLBACK not issued prior to Begin Work call (LUW active)
-62	NOLOG	Operation requires a log, but none exists (should not occur on BEGIN WORK or COMMIT as RDS does not support user log data)
-63	INVSCHED	SCHEDULE opcode issued for active LUW
-64	SYABORT	System requested LUW abort
-65	INLUW	Operation not valid while agent has active LUW
-66	OPABORT	LUW rolled back due to operator FORCE command
		Illegal Parameters in SBASE
-70	ILLEGALL	Illegal length of the DBSPACE list
-71	ILISTPTR	Illegal pointer to DBSPACE list
		Exceptional Conditions on a DBSPACE
-72	SGISOPEN	At least one DBSPACE already open
-73	SEMPTIED	Cannot be reopened until a synchronous save because CDELETE was issued
-74	NOAVAILS	No available temporary DBSPACE
-75	NOTAVSEG	DBSPACE not available or write operation with DBSPACE opened for read
-76	NOROOMH	No header room in DBSPACE in write call
-77	NOROOME	No entity room in DBSPACE in write call
-78	NOROOMI	No index room in DBSPACE in write call
		DBSS Errors and Verson Limitations
-80	RSSERROR	Error in DBSS code. Error ID in feedback area (Should not occur since DBSS terminates SQL/DS on any detected DBSS code error)
-81	IOERROR	DASD I/O error. DBSS always abends SQL/DS on I/O error. (Should not occur)
-82	NOTIMPL	Feature not yet implemented. (Should not occur)
-83	NOMEMORY	Unable to allocate needed main storage
		RETURN CODES USED ONLY IN SYSTEM WITH LOG
		Illegal Parameters in LUW Operators
-85	OUTSIDE	Pointer to user save data that is bad (Should never occur since RDS does not support user log data)
-86	TOOLONG	User wants to log more than 64K (Should never occur since RDS does not support user log data)
-87	BADSAVNO	Tried to back over non-existent save point
-89	OKBUTMOR	Read user log data but there was more. (Should never occur since RDS does not support user log data)
-90	INVIDLST	Output area too small for INDOUBT list
-91	SYBACKUP	System requested LUW backup
		RETURN CODES USED ONLY IN MULTI-USER SYSTEM
-92	CHECKING	Checkpoint in progress, try later
		Illegal Parameters Related to Locking

Return Code	Name	Meaning
-94	BADCLVL	Consistency level not 1, 2, or 3 (should not occur as RDS always uses consistency level 3)
-95	ILCKMODE	Lock mode not in (SHAR,SIX,EXCL)
		Exceptional Conditions Due to Locking
-97	LOCKLIM	Too many locks held by this LUW
-98	PRIVSEGM	Requested private DBSPACE is locked
		LUW Backout Due to Locking (Reached Save Point in FEEDBACK(1))
-99	DEADLOCK	LUW backout due to deadlock
-100	OVERLOAD	LUW backout due to too many locks being held system-wide
-101	DEADATSI	Deadlock backout to a DBSI boundary...internal retry failed
		RETURN CODES USED ONLY IN SINGLE-USER SYSTEM
-120	NOTMUL	Operation not allowed in single-user system issued by operator command OPCODE (OPCMMD)
		MORE RETURN CODES RELATED TO LUW MANAGEMENT
-123	ATTNBACK	OPCODE terminated due to asynchronous request to terminate LUW; caller should issue ROLLBACK. Can occur only if YTABLE1 flag ATTNATTN set to 'Y'. SQL/DS does not set ATTNATTN flag to 'Y'. Should not occur.
		RETURN CODES USED BY THE SORT SUBSYSTEM
-130	ILLSOPTR	Illegal address of SORTSPEC
-131	INORDSPC	Illegal number of sort order specifications
-132	ISORTSPC	Illegal sort order specification
-133	NOEXDOMN	Field number does not exist for input table-id
-134	IDUPELIM	Illegal duplicate elimination option
-135	INSLOOMS	Illegal number of SORTLIST fields
-136	NORMTPSG	No space in temporary DBSPACE for write call
-137	DUPFOUND	Duplicate found during sort/merge; stop due to user option
-138	ILLSLPTR	Illegal pointer to SORTLIST specification
-139	EMPTYSRT	Empty set of rows to sort: sort DBSPACE is unchanged
-140	ILTIDSPC	Illegal TID specification: input list has no TID's
-141	ILDOMREP	Illegal domain repetition: output list domain in SORTSPEC specified more than once
-142	ENCTPTLG	Encoded row too long: does not fit in an intermediate page
-143	ISORTERR	Internal sort error: number of rows sorted and merged different
		RETURN CODES USED BY THE UPDATE STATISTICS COMMANDS
-400	ILGMODE	Illegal update statistics mode
-401	ILNUMREL	Illegal number of specified tables
-402	DUPRELID	Duplicate table-id in input list
-403	IPATHTYP	Illegal path type
-404	ILPATHID	Illegal path ID
-405	MISMATCH	RELID and PATHID do not correspond
-406	NOTAROOT	First table-id on input list is not a root
-407	NOPARNT	No parent specified for binary link
-409	UNCLUSTI	Unclustered indexes exist
-410	INCONREL	NREL or RELS(*) inconsistent with that previously specified with 'STARTUS' call

SQL/DS REASON CODES

The SQL/DS reason codes are printed in message ARI042I. The SQL/DS reason codes are the VSE/Advanced Functions and VM/SP return codes that were returned to SQL/DS. Message ARI042I is usually accompanied by other SQL/DS messages; for example: ARI026E, ARI027E, ARI039E, ARI040E, ARI041E, etc. The return codes fall into three categories:

1. VSE/Advanced Functions and VM/SP System Services (such as CDLOAD, GETVIS, NUCXLOAD, or DMSFREE).

2. VSE/Advanced Functions and VM/SP Abnormal Termination Codes.

3. VSE/Advanced Functions Cross-Partition Communication (XPCC) and VM/SP Inter-User Communication Vehicle (IUCV) return codes.

VSE/ADVANCED FUNCTIONS AND VM/SP SYSTEM SERVICES RETURN CODES

The VSE/Advanced Functions System Services return codes are found in VSE/Advanced Functions Macro Reference, SC24-5211.

The VM/SP System Services return codes are found in the following manuals:

Virtual Machine/System Product: CMS Command and Macro Reference, SC19-6209.

Virtual Machine/System Product: System Programmer's Guide, SC19-6203.

Virtual Machine/System Product: System Messages and Codes, SC19-6204.

The accompanying SQL/DS messages should assist in determining the system service that was being invoked.

The return codes from the VSE/Advanced Functions abnormal termination codes can be found in the VSE/Advanced Functions Macro Reference Manual under the discussion of the STXIT AB macro instruction. For the SQL/DS library documentation of the SQL/DS reason codes and the VSE/Advanced Functions Cross-Partition Communication (XPCC) Return Codes, see the Messages and Codes manual as listed in the Preface.

SQL CODES - RDS CODES: DETECTING MODULES

The following list gives the SQL Code (SQLC) and RDS Code (RDSC) pairs with the module(s) that detect those codes. The list is arranged in numerical order by SQL code. (RDS code comes from the SQLERRD1 field in SQLCA.)

SQLC	RDSC	Detecting Module
+150	-325	ARIXA09
+200	-273	ARIXERD
-101	-001	ARIXC01 ARIXC10 ARIXC15
		ARIXC19 ARIXC31 ARIXC32
		ARIXC40 ARIXC41 ARIXC42
		ARIXC43
-101	-003	ARIXC18 ARIXC21
-101	-006	ARIXC18
-101	-100	ARIPSQC ARIXEDP ARIXOFP
-101	-101	ARIPSQC ARIXOML ARIXOFQ
-101	-102	ARIPSQC ARIXC37 ARIXOB2
		ARIXOFC ARIXOFT
-101	-103	ARIXC38 ARIXOB2 ARIXOFE
-101	-105	ARIXC37
-101	-107	ARIPSQC
-101	-501	ARIXPA1
-101	-820	ARIPSQA ARIPSQP
-102	-111	ARIPSQC
-102	-112	ARIPSQP
-102	-302	ARIXPA1
-102	-307	ARIXPA1
-102	-309	ARIXPA1
-103	-304	ARIXPA1
-104	-119	ARIPPRC
-104	-303	ARIXPA1
-104	-304	ARIXPA1
-104	-311	ARIXPA1
-104	-319	ARIXPA1
-104	-320	ARIXPA1
-104	-502	ARIXPA2
-104	-800	ARIPSQA
-104	-801	ARIPPRA ARIPPRA ARIPPRC
		ARIPPRP ARIPSQP
-104	-810	ARIPPRC ARIPPPC
-104	-999	ARIPPRC
-105	-305	ARIXPA1
-106	-308	ARIXPA1
-106	-321	ARIXPA1
-106	-801	ARIPPRA ARIPPRC ARIPPRP
-107	-301	ARIXPA1
-107	-306	ARIXPA1
-107	-860	ARIPPRA ARIPPRC ARIPPRP
-107	-999	ARIPPRC
-108	-307	ARIXPA2 ARIXPA3
-108	-860	ARIPPRA ARIPPRC ARIPPRP
-109	-120	ARIPPRC
-109	-808	ARIPPRA ARIPPRC ARIPPRP
-111	-150	ARIXOOS
-112	-130	ARIXOOS
-112	-155	ARIXOEX ARIXOOS
-112	-170	ARIXOEX
-113	-901	ARIPPRC
-115	-106	ARIXOW1
-115	-314	ARIXPA3
-117	-106	ARIXOIN
-117	-107	ARIXOIN
-117	-108	ARIXOIN
-118	-115	ARIXOGC
-118	-116	ARIXOGC
-118	-117	ARIXOGC
-119	-143	ARIXOVD
-119	-147	ARIXOVD
-119	-200	ARIXOGB
-120	-110	ARIXOEX
-120	-120	ARIXOEX
-120	-176	ARIXOEX
-121	-104	ARIXOFF
-121	-118	ARIXODM
-121	-119	ARIXODM
-124	-302	ARIXOGP
-125	-100	ARIXONV
-125	-303	ARIXOGP
-125	-305	ARIXOGP
-125	-318	ARIXPA3
-126	-304	ARIXOGP
-127	-105	ARIXOEX
-127	-135	ARIXOOS
-128	-127	ARIXOEX
-129	-501	ARIXPA2 ARIXPA3
-130	-312	ARIXPA1
-132	-315	ARIXPA3
-133	-115	ARIXOEX
-134	-100	ARIXOLF
-134	-103	ARIXOLF
-134	-105	ARIXOLF
-134	-106	ARIXOGP
-134	-120	ARIXOOS
-134	-215	ARIXOEX
-135	-105	ARIXOFF
-150	-140	ARIXOVC
-151	-210	ARIXOVC
-152	-240	ARIXOVC
-153	-130	ARIXOVD
-154	-100	ARIXOVD
-154	-110	ARIXOVD
-154	-120	ARIXOVD
-154	-140	ARIXI06 ARIXI13
-154	-150	ARIXOVC
-154	-220	ARIXOVC
-155	-180	ARIXOVC
-158	-200	ARIXOVD
-158	-210	ARIXOVD
-159	-312	ARIXPA2
-159	-813	ARIPPRA
-160	-313	ARIXPA2
-160	-869	ARIPPRA
-161	-810	ARIPPRA
-162	-296	ARIXERD
-162	-811	ARIPPRA
-163	-503	ARIXPA2
-164	-111	ARIPSQP
-164	-115	ARIPSQC
-164	-314	ARIXPA1
-165	-322	ARIXPA1
-166	-103	ARIPSQP
-166	-106	ARIPSQP
-166	-113	ARIPSQC
-166	-114	ARIPSQC ARIPSQP
-166	-315	ARIXPA1
-167	-111	ARIPSQC
-167	-113	ARIPSQP
-167	-321	ARIXPA1
-168	-311	ARIXPA2
-169	-112	ARIPSQC
-169	-115	ARIPSQC
-170	-319	ARIXPA1
-170	-101	ARIPSQP
-170	-102	ARIPSQP
-170	-104	ARIPSQP

SQLC	RDSC	Detecting Module
-170	-105	ARIPSQP
-170	-107	ARIPSQP
-170	-108	ARIPSQP
-170	-109	ARIPSQP
-170	-110	ARIPSQP
-170	-115	ARIPSQP ARIPSQC
-170	-320	ARIXPA1
-191	-200	ARIXIST
-198	-106	ARIFSQC
-201	-309	ARIXPA2
-201	-310	ARIXPA3
-202	-104	ARIXOCK
-202	-306	ARIPPRC
-203	-101	ARIXOCK
-204	+110	ARIXOCA
-204	-100	ARIXA09
-204	-120	ARIXA07 ARIXI06 ARIXI10 ARIXI13 ARIXI15
-204	-140	ARIXI03 ARIXI15
-204	-150	ARIXI33
-204	-170	ARIXI10
-204	-180	ARIXA07
-204	-190	ARIXI10
-204	-220	ARIXI10
-204	-240	ARIXI10
-204	-320	ARIXI10
-204	-330	ARIXA09
-204	-410	ARIXI10
-204	-510	ARIXOD1
-204	-570	ARIXI10
-205	-102	ARIXOCK
-205	-103	ARIXOCK
-208	-120	ARIXONV
-209	-110	ARIXONV
-210	-100	ARIPPRA ARIPPRC ARIPPRP
-210	-106	ARIXI10
-210	-272	ARIXERD
-210	-281	ARIXERD
-211	-320	ARIXPA2 ARIXPA3
-301	-001	GENCODE (ARIXCRE, ARIXCRH, ARIXCR5, ARIXCR6, ARIXCR7, ARIXCR8)
-301	-114	ARIPPRC
-301	-115	ARIPPRC
-301	-117	ARIXPRC
-302	-001	GENCODE (ARIXCRH, ARIXCR5, ARIXCR6, ARIXCR7, ARIXCR8)
-303	-001	GENCODE (ARIXCR2)
-303	-201	ARIPPRC
-304	-001	GENCODE (ARIXCR2, ARIXC36)
-305	-001	GENCODE (ARIXCR2, ARIXC36)
-306	-805	ARIXEPP
-306	-840	ARIPPRA ARIPPRC ARIPPRP
-306	-842	ARIPPRC
-307	-106	ARIPPRC
-307	-806	ARIPPRC ARIPPRP
-307	-807	ARIPPRA
-308	-107	ARIPPRC
-308	-804	ARIPPRA ARIPPRP
-309	-001	GENCODE (ARIXCRE, ARIXCRH, ARIXCR5, ARIXCR6, ARIXCR7, ARIXCR8)
-310	-100	ARIPPRC
-310	-102	ARIPPRC
-310	-103	ARIPPRC
-310	-105	ARIPPRC
-310	-116	ARIPPRC
-310	-841	ARIPPRA ARIPPRC ARIPPRP
-311	-001	GENCODE (ARIXCR7, ARIXCR8)
-312	-110	ARIXEDP
-312	-833	ARIXEPP
-313	-001	GENCODE (ARIXF97)
-314	-402	ARIPPRA ARIPPRC
-314	-403	ARIPPRA ARIPPRC ARIPPRP
-314	-404	ARIPLTP ARIPPRA ARIPPRC
-314	-405	ARIPLTP ARIPPRA ARIPPRC
-314	-406	ARIPLTP ARIPPRA ARIPPRC
-314	-407	ARIPLTP ARIPPRA ARIPPRC
-314	-408	ARIPLTP ARIPPRA ARIPPRC
-314	-409	ARIPLTP ARIPPRA ARIPPRC
-314	-410	ARIPLTP ARIPPRA ARIPPRC
-314	-411	ARIPPRA ARIPPRC ARIPPRP
-314	-412	ARIPPRA ARIPPRC ARIPPRP
-314	-413	ARIPLTP ARIPPRA ARIPPRC
-314	-414	ARIPLTP ARIPPRA ARIPPRC
-314	-415	ARIPLTP ARIPLTP
-314	-416	ARIPLTP ARIPPRC
-314	-417	ARIPLTP ARIPPRC
-314	-418	ARIPLTP
-314	-419	ARIPPRC
-314	-421	ARIPSQC
-314	-801	ARIPPRC
-315	-841	ARIPPRA ARIPPRC ARIPPRP
-316	-100	ARIPSQC
-317	-101	ARIPSQC
-317	-111	ARIPSQC
-317	-112	ARIPSQC
-318	-102	ARIPSQC
-319	-103	ARIPSQC
-320	-104	ARIPSQC
-321	-105	ARIPSQC
-322	-108	ARIPSQC
-323	-109	ARIPSQC
-324	-109	ARIXI10
-324	-812	ARIPPRA
-325	-814	ARIPPRA
-401	-104	ARIXOTF
-401	-105	ARIXOOS
-401	-110	ARIXOW2
-401	-125	ARIXOEX ARIXOOS
-401	-126	ARIXOOS
-401	-170	ARIXOOS
-401	-185	ARIXOEX
-401	-209	ARIXOW2
-401	-213	ARIXOW2
-401	-214	ARIXOW2
-401	-215	ARIXOW2
-401	-220	ARIXOEX
-401	-225	ARIXOEX
-401	-235	ARIXOEX
-401	-244	ARIXOEX
-401	-245	ARIXOEX
-401	-258	ARIXOEX
-401	-259	ARIXOEX
-401	-260	ARIXOEX
-401	-261	ARIXOEX
-401	-262	ARIXOEX
-401	-263	ARIXOEX
-401	-264	ARIXOEX
-401	-265	ARIXOEX
-402	-100	ARIXOEX
-402	-110	ARIXOOS
-402	-115	ARIXOOS
-402	-135	ARIXOEX
-402	-145	ARIXOEX
-402	-165	ARIXOOS
-402	-180	ARIXOEX
-403	-160	ARIXOOS
-403	-175	ARIXOEX
-404	-109	ARIXOTF
-404	-110	ARIXOTF
-405	-101	ARIXOTF ARIXOTF ARIXOTF
-406	-001	GENCODE (ARIXC36)
-407	-001	GENCODE (ARIXCRE, ARIXCR5, ARIXCR6, ARIXCR7, ARIXCR8, ARIXC04, ARIXC35)
-407	-103	ARIXOFF
-407	-114	ARIXODM

SQLC	RDSC	Detecting Module
-408	-102	ARIXOFF
-408	-106	ARIXOFF
-408	-107	ARIXOFF
-408	-108	ARIXOFF
-408	-109	ARIXOFF
-408	-233	ARIXOOS
-409	-145	ARIXOOS
-409	-165	ARIXOEX
-410	-100	ARIXOTF ARIXOTF ARIXOTF
-411	-112	ARIXOTF
-411	-113	ARIXOTF
-411	-140	ARIXOEX
-411	-210	ARIXOEX ARIXOOS
-411	-319	ARIXPA2
-412	-109	ARIXOW2
-413	-001	GENCODE (ARIXC36)
-414	-104	ARIXOW2
-415	-103	ARIXOSO
-416	-103	ARIXOSO
-417	-100	ARIXOW2
-417	-105	ARIXOW2
-417	-251	ARIXOEX
-417	-252	ARIXOEX
-418	-121	ARIXOOS
-419	-190	ARIXOOS
-419	-260	ARIXOEX
-421	-105	ARIXOTF
-421	-207	ARIXOW2
-421	-257	ARIXOEX
-422	-208	ARIXOEX
-422	-210	ARIXOW2
-422	-211	ARIXOW2
-423	-212	ARIXOW2
-501	-001	GENCODE (ARIXF03)
-501	-100	ARIXECL
-502	-001	GENCODE (ARIXF03)
-503	-001	GENCODE (ARIXF47)
-504	-001	GENCODE (ARIXCR4)
-504	-822	ARIPPRA ARIPPRC ARIPPRP
-505	-820	ARIPPRA ARIPPRC ARIPPRP
-507	-001	GENCODE (ARIXF21, ARIXF47)
-508	-001	GENCODE (ARIXF20, ARIXF21, ARIXF26, ARIXF47)
-509	-001	(ARIXF21, ARIXF47)
-510	-001	GENCODE (ARIXF21, ARIXF47)
-511	-107	ARIXOOP
-511	-110	ARIXOGP
-512	-240	ARIXERD
-512	-250	ARIXERD
-513	-800	ARIPPRA ARIPPRC ARIPPRP
-515	-165	ARIXOOP
-516	-001	ARIXC38
-517	-274	ARIXERD
-518	-278	ARIXERD
-551	-100	ARIXOIN ARIXOSC
-551	-103	ARIXOOP
-551	-105	ARIXI09
-551	-106	ARIXI09
-551	-107	ARIXI09
-551	-108	ARIXI10
-551	-110	ARIXOOP
-551	-112	ARIXI09
-551	-114	ARIXI09
-551	-115	ARIXOOP
-551	-120	ARIXOOP
-551	-125	ARIXOOP
-551	-127	ARIXOOP
-551	-131	ARIXOOP
-551	-140	ARIXOOP
-551	-150	ARIXI09 ARIXOVD
-551	-160	ARIXOOP
-551	-161	ARIXOOP
-551	-200	ARIXA07 ARIXOOP
-551	-201	ARIXOOP
-551	-913	ARIXELX ARIXERD
-552	-102	ARIXOOP
-552	-116	ARIXOOP
-552	-126	ARIXOOP
-552	-130	ARIXOOP
-552	-170	ARIXA07
-552	-185	ARIXI07
-552	-200	ARIXA09
-554	-110	ARIXA07
-555	-130	ARIXA09
-555	-210	ARIXA09
-555	-350	ARIXA09
-556	-140	ARIXA09
-556	-220	ARIXA09
-556	-231	ARIXA09
-556	-235	ARIXA09
-556	-360	ARIXA09
-557	-120	ARIXA09
-557	-140	ARIXA07
-557	-160	ARIXA07
-557	-190	ARIXA09
-557	-210	ARIXA07
-558	-232	ARIXA09
-560	-090	ARIXA01
-560	-130	ARIXA01
-561	-110	ARIXA01
-561	-163	ARIXA07
-562	-161	ARIXA07
-563	-301	ARIXERD
-564	-080	ARIXA01
-564	-270	ARIXA07
-565	-280	ARIXERD
-566	-810	ARIXELX
-601	-100	ARIXI06 ARIXI08
-601	-102	ARIXI09
-601	-107	ARIXI08
-601	-110	ARIXI07
-601	-119	ARIXI09
-601	-123	ARIXI09
-601	-130	ARIXI01 ARIXI07
-601	-280	ARIXI13
-602	-100	ARIXI07 ARIXI09
-602	-150	ARIXI09
-602	-200	ARIXI06 ARIXI13 ARIXOVD
-602	-240	ARIXI07
-603	-210	ARIXI06
-604	-316	ARIXPA3
-604	-317	ARIXPA1 ARIXPA3
-606	-101	ARIXI04
-608	-150	ARIXI07
-608	-170	ARIXI07
-608	-300	ARIXI33
-609	-160	ARIXI03
-610	-190	ARIXI07 ARIXI13
-612	-310	ARIXI07
-614	-170	ARIXI06
-614	-180	ARIXI06
-618	-947	ARIXESX
-619	-996	ARIXOD2
-619	-997	ARIXOD2
-619	-998	ARIXOD1
-619	-999	ARIXOD2
-657	-110	ARIXI33
-701	-077	DBSS
-702	-078	DBSS
-703	-076	DBSS
-704	-150	ARIXI01
-704	-170	ARIXI01
-705	-100	ARIXI01 ARIXI03
-705	-110	ARIXI03
-705	-120	ARIXI01
-705	-130	ARIXI03
-705	-170	ARIXI03
-705	-185	ARIXI03
-705	-190	ARIXI06
-705	-210	ARIXI01
-705	-220	ARIXI01
-705	-318	ARIXPA1
-708	-135	ARIXI15

SQLC	RDSC	Detecting Module
-708	-139	ARIXI03
-708	-310	ARIXI10
-709	-340	ARIXI10
-710	-098	DBSS
-712	-001	GENCODE (ARIXCRA, ARIXCRB)
-751	-060	DBSS
-751	-100	ARIXI12
-752	-095	ARIXA01
-752	-101	ARIXI02
-752	-135	ARIXA01
-754	-062	DBSS
-756	-271	ARIXERD
-757	-291	ARIXERD
-758	-107	ARIXI10
-758	-284	ARIXERD
-759	-956	ARIXESX
-760	-293	ARIXERD
-761	-957	ARIXESX
-802	-907	ARICABE ARIXEBR
-802	-908	ARICABE ARIXEBR
-802	-909	ARICABE ARIXEBR
-802	-910	ARICABE ARIXEBR
-802	-911	ARICABE ARIXEBR
-802	-912	ARICABE ARIXEBR
-802	-913	ARICABE ARIXEBR
-802	-914	ARICABE ARIXEBR
-802	-915	ARICABE ARIXEBR
-803	-034	DBSS
-804	-000	ARIRORM ARIRBRM ARIRVRM
-804	-001	GENCODE (ARIXF02)
-804	-270	ARIXERD
-805	-807	ARIXELX
-805	-820	ARIXELX
-806	-808	ARIXELX
-806	-821	ARIXELX
-807	-210	ARIXERD
-807	-800	ARIXERD
-807	-820	ARIXERD
-809	-036	DBSS
-810	-001	GENCODE (ARIXF24)
-813	-160	ARIXI15
-814	-170	ARIXI15
-820	-121	ARIPPRC
-820	-122	ARIPPRC
-820	-123	ARIPPRC
-821	-100	ARIXEDP
-823	-151	ARIXOOP
-824	000	ARIRORM ARIRVIR ARIRVRM
-825	000	ARIXEOC
-826	-276	ARIXERD
-826	-277	ARIXERD
-826	-285	ARIXERD
-826	-286	ARIXERD
-826	-287	ARIXERD
-826	-288	ARIXERD
-826	-289	ARIXERD
-826	-290	ARIXERD
-827	-283	ARIXERD
-827	-823	ARIXELX
-827	-950	ARIXESX
-828	-282	ARIXERD
-829	-949	ARIXESX
-830	-811	ARIXELX
-831	-279	ARIXERD
-832	-292	ARIXERD
-832	-294	ARIXERD
-901	-001	ARIXC02 ARIXC03 ARIXC05 ARIXC06 ARIXC07 ARIXC14 ARIXC18 ARIXC20 ARIXC21 ARIXC22 ARIXC23 ARIXC24 ARIXC27 ARIXC28 ARIXC29 ARIXC30 ARIXC33 ARIXC34 ARIXC36 ARIXC39 ARIXC45 ARIXC47 ARIXC48 ARIXC51 ARIXF62 GENCODE (ARIXCRA, ARIXCRB, ARIXCRC, ARIXCRE, ARIXCR0, ARIXCR5, ARIXCR6, ARIXCR7, ARIXF19, ARIXF20, ARIXF21, ARIXF25, ARIXF27, ARIXF29, ARIXF30, ARIXF48, ARIXF49, ARIXF57, ARIXF62, ARIXF75, ARIXF77, ARIXF81)
-901	-002	ARIXC02 ARIXC03 ARIXC05 ARIXC06 ARIXC18 ARIXC20 ARIXC21 ARIXC24 ARIXC27 ARIXC28 ARIXC29 ARIXC30 ARIXC32 ARIXC36 ARIXC39 ARIXC45 ARIXC48 GENCODE (ARIXCR2, ARIXF62)
-901	-003	ARIXC02 ARIXC05 ARIXC06 ARIXC36 GENCODE (ARIXF81)
-901	-004	ARIXC05 ARIXC06 ARIXC18 ARIXC36
-901	-005	ARIXC05 ARIXC18 ARIXC36
-901	-006	ARIXC05
-901	-007	ARIXC05
-901	-008	ARIXC05
-901	-009	ARIXC05
-901	-010	ARIXC05
-901	-050	ARIXA02
-901	-100	ARIXEAB ARIXEBR ARIXEDP ARIXEFB ARIXEPH ARIXEPP ARIXERP ARIXEUH ARIXI10 ARIXI14 ARIXI15 ARIXOBY ARIXOCK ARIXOCY ARIXOGA ARIXOIS ARIXOMS ARIXOOP ARIXOSO ARIXOVC ARIXSUT
-901	-101	ARIXC37 ARIXEDP ARIXEPH ARIXEPP ARIXI09 ARIXI20 ARIXOFC ARIXOFT ARIXOGA ARIXOIN ARIXOIS ARIXOLF ARIXOMD ARIXOMS ARIXOOP ARIXOSO ARIXOSR ARIXOW2
-901	-102	ARIXEDP ARIXERP ARIXI20 ARIXOFE ARIXOGA ARIXOIN ARIXOLC ARIXOLF ARIXOMA ARIXOMD ARIXOML ARIXOMS ARIXOQ1 ARIXOSO ARIXOSR ARIXOSU ARIXOTF ARIXOW1 ARIXOW2
-901	-103	ARIXI20 ARIXOGA ARIXOIN ARIXOMA ARIXOML ARIXOMS ARIXOQ1 ARIXOSU ARIXOTF ARIXOW2 ARIXO1S
-901	-104	ARIXEAB ARIXC38 ARIXI09 ARIXI20 ARIXOGA ARIXOIN ARIXOIS ARIXOLF ARIXOMA ARIXOML ARIXOMS ARIXOOP ARIXOQ1 ARIXOSR ARIXOSU ARIXOW1
-901	-105	ARIPPRC ARIXI04 ARIXI10 ARIXI20 ARIXOIN ARIXOMA ARIXOMS ARIXOOP ARIXOQ1
-901	-106	ARIXOMA ARIXOMS ARIXOOP ARIXOTF ARIXOW2
-901	-107	ARIXOTF ARIXOW1 ARIXOW2
-901	-108	ARIPPRC ARIXI09 ARIXI20 ARIXOTF ARIXOW2
-901	-109	ARIXI09 ARIXI20
-901	-110	ARIXI09 ARIXI10 ARIXI15 ARIXI20 ARIXOIN ARIXOVC
-901	-111	ARIPPRC ARIXI09 ARIXI20 ARIXOCT ARIXODM ARIXOFR ARIXOGC ARIXOTF ARIXOW2
-901	-112	ARIXI20 ARIXOCT ARIXODM ARIXOFR ARIXOW2
-901	-113	ARIXI09 ARIXI19 ARIXI20 ARIXOCT ARIXOSL ARIXOW2
-901	-115	ARIXI20 ARIXODM
-901	-116	ARIXI09 ARIXODM ARIXOST ARIXOW1

SQLC	RDSC	Detecting Module
-901	-117	ARIXI09 ARIXOW1
-901	-118	ARIXOB2
-901	-120	ARIXEDP ARIXOCK ARIXODF
		ARIXOD1 ARIXOGC ARIXOVC
-901	-121	ARIXI09 ARIXOGC ARIXOOP
-901	-130	ARIXEDP ARIXODF ARIXOD1
		ARIXOEX ARIXONV ARIXOTF
		ARIXOVC
-901	-140	ARIXODF ARIXOOS ARIXOVD
-901	-142	ARIXODF
-901	-148	ARIXODF
-901	-150	ARIXI06 ARIXODF ARIXOEX
		ARIXOOP
-901	-152	ARIXODF
-901	-155	ARIXIST ARIXOOS
-901	-160	ARIXI06 ARIXOEX ARIXOVC
		ARIXOVD
-901	-167	ARIXIST
-901	-170	ARIXI10 ARIXOEX ARIXOVC
		ARIXOVD
-901	-171	ARIXIST
-901	-175	ARIXOOS
-901	-180	ARIXI10 ARIXOOS ARIXOVD
-901	-185	ARIXOOS
-901	-190	ARIXOEX ARIXOVC ARIXOVD
-901	-195	ARIXOEX
-901	-200	ARIXI04 ARIXOEX ARIXOMS
		ARIXOTF ARIXOVC
-901	-201	ARIXI04 ARIXI19 ARIXOMS
-901	-202	ARIXOMS
-901	-203	ARIXOMS
-901	-205	ARIXOEX
-901	-210	ARIXI07
-901	-220	ARIXA07 ARIXERD
-901	-230	ARIXERD ARIXI07 ARIXOEX
		ARIXOOS ARIXOOS ARIXOVC
-901	-231	ARIXOOS
-901	-240	ARIXOEX
-901	-250	ARIXOEX ARIXOVC
-901	-255	ARIXOEX
-901	-260	ARIXOVC ARIXOV1
-901	-270	ARIXOV1
-901	-275	ARIXERD
-901	-280	ARIXOV1
-901	-290	ARIXOV1
-901	-300	ARIXEPP ARIXOV1
-901	-310	ARIXOVC
-901	-320	ARIXOVC
-901	-330	ARIXOVC ARIXOV1
-901	-340	ARIXOVC ARIXOV1
-901	-350	ARIXOVC

SQLC	RDSC	Detecting Module
-901	-360	ARIXOAF ARIXOCK ARIXOML
		ARIXOML ARIXOOP ARIXOSR
		ARIXOSU ARIXOVC
-901	-450	ARIXOGP
-901	-451	ARIXOGP
-901	-452	ARIXOGP
-901	-460	ARIXI10
-901	-560	ARIXI10
-901	-815	ARIPPRC
-901	-830	ARIXEPP
-901	-831	ARIXEPP
-901	-832	ARIXEPP
-901	-900	ARIXELX
-901	-901	ARICABE ARIXEBR
-901	-902	ARICABE ARIXEBR
-901	-903	ARICABE ARIXEBR
-901	-904	ARICABE ARIXEBR
-901	-905	ARICABE ARIXEBR
-901	-906	ARICABE ARIXEBR
-901	-914	ARIXELX
-901	-990	ARIXOD1
-902	-1000	DBSS
-903	-064	DBSS
-903	-091	DBSS
-903	-109	ARIXEAD
-911	-099	DBSS
-911	-101	DBSS
-912	-100	DBSS
-914	-000	ARIRBRM ARIRVRM
-914	-123	DBSS
-915	-097	DBSS
-915	-180	ARIXI15
-916	-***	DBSS ARIRBRM ARIRVRM
-921	-100	ARIXOCK
-930	-001	ARIXC13 ARIXC37 GENCODE (ARIXCR2, ARIXCR3)
-930	-002	ARIXC13 ARIXC19
-930	-003	ARIXC13
-930	-004	ARIXC21 GENCODE (ARIXCR2, ARIXCR3)
-930	-100	ARIXEAD ARIXEDP ARIXELX ARIXEPP ARIXERD ARIXIST ARIXI33 ARIXPA1
-930	-101	ARIXELX ARIXESX ARIXOCS ARIXOTS
-930	-102	ARIXEAD ARIXEAB ARIXEDS ARIXELX ARIXIST
-930	-103	ARIXEAD ARIXELX ARIXESX
-930	-104	ARIXELX
-930	-105	ARIXELX
-930	-106	ARIXELX ARIXESX
-930	-110	ARIXEAD ARIXOD2

SQLC	RDSC	Detecting Module
-930	-444	ARIXOVD
-930	-450	ARIXOGP
-930	-451	ARIXOGP
-931	-001	GENCODE (ARIXCR2, ARIXCR3)
-931	-002	ARIXC14
-931	-003	ARIXC14 ARIXC19
-931	-004	ARIXC14
-931	-005	ARIXC21 ARIXC37 GENCODE (ARIXCR3)
-931	-006	ARIXCR2
-931	-100	ARIXEDP ARIXEFB ARIXEPP ARIXERD ARIXPA1
-931	-101	ARIXEDS ARIXESX
-931	-102	ARIXESX
-931	-104	ARIXEAD ARIXESX
-931	-105	ARIXEAD ARIXESX
-931	-106	ARIXEAD
-931	-107	ARIXEAD ARIXELX
-931	-108	ARIXEAD ARIXELX
-931	-109	ARIXELX
-931	-110	ARIXELX
-931	-111	ARIXEAD ARIXELX
-931	-112	ARIXEAD
-931	-295	ARIXERD
-931	-399	ARIXERD
-931	-445	ARIXOVD
-931	-446	ARIXOVD
-931	-447	ARIXOVD
-931	-448	ARIXOVD
-931	-449	ARIXOVD
-931	-450	ARIXI33
-931	-452	ARIXOGP
-931	-453	ARIXOGP
-931	-600	ARIXI33
-931	-650	ARIXI33
-932	-004	ARIRBRM ARIRORM ARIRVIR ARIRVRM ARIRVST
-932	-008	ARIRBRM ARIRORM ARIRVIR ARIRVRM ARIRVST
-932	-012	ARIRBRM ARIRORM ARIRVIR ARIRVRM ARIRVST
-932	-016	ARIRBRM ARIRORM ARIRVIR ARIRVRM ARIRVST
-933	-004	ARIRBRM ARIRORM ARIRVIR ARIRVRM
-933	-008	ARIRBRM ARIRORM ARIRVIR ARIRVRM
-933	-012	ARIRBRM ARIRORM ARIRVIR ARIRVRM
-933	-016	ARIRBRM ARIRORM ARIRVIR ARIRVRM

SQLC	RDSC	Detecting Module	SQLC	RDSC	Detecting Module	SQLC	RDSC	Detecting Module
-934	-000	ARIPRDI ARIRIRM ARIRVIR ARIRVST						
-935	-000	ARIRBRM ARIRORM ARIRVRM						
-936	-000	ARIRENA						
-937	-000	ARIRORM						
-938	-000	ARIRIRM ARIRVIR						
-939	-000	ARIRVIR						
-940	-000	ARIRBRM ARIRVRM						
-941	-004	ARIRBRM ARIRVRM						
-941	-008	ARIRBRM ARIRVRM						
-941	-012	ARIRBRM ARIRVRM						
-941	-016	ARIRBRM ARIRVRM						
-942	000	ARIRBRM ARIRORM ARIRVRM						
-946	-892	ARIXESX						
-946	-894	ARIXESX						
-946	-953	ARIXESX						
*	-101	ARIXEAD						
*	-812	ARIXELX						
*	-822	ARIXELX						
*	-914	ARIXESX						
*	-948	ARIXESX						
*	-951	ARIXESX						
*	-952	ARIXESX						
*	-954	ARIXESX						
*	-955	ARIXESX						

* SQLCODE depends upon DBSS error

MODULES DETECTING SQL CODES AND RDS CODES

The following list gives the modules that may detect the pairs of SQL Codes (SQC) and RDS Codes (RDC). The list is arranged i
alphameric sequence by module names. (RDS code comes from the SQLERRD1 field in SQLCA.)

Module	SQLC	RDSC
ARICABE	-802	-907
	-802	-908
	-802	-909
	-802	-910
	-802	-911
	-802	-912
	-802	-913
	-802	-914
	-802	-915
	-901	-901
	-901	-902
	-901	-903
	-901	-904
	-901	-905
	-901	-906
ARIPLTP	-314	-404
	-314	-405
	-314	-406
	-314	-407
	-314	-408
	-314	-409
	-314	-410
	-314	-413
	-314	-414
	-314	-415
	-314	-416
	-314	-417
	-314	-418
ARIPPRA	-104	-801
	-106	-801
	-107	-860
	-108	-860
	-109	-808
	-159	-813
	-160	-809
	-161	-810
	-162	-811
	-210	-100
	-306	-840
	-306	-842
	-307	-806
	-307	-807
	-308	-804
	-310	-841
	-314	-402
	-314	-403
	-314	-404
	-314	-405
	-314	-406
	-314	-407
	-314	-408
	-314	-409
	-314	-410
	-314	-411
	-314	-412
	-314	-413
	-314	-414
	-315	-841
	-324	-812
	-325	-814
	-504	-822
	-505	-820
	-513	-800
ARIPPRC	-104	-801
	-104	-119
	-104	-810
	-104	-999
	-106	-801
	-107	-860
	-107	-999
	-108	-860
	-109	-808
	-109	-120
	-113	-901
	-202	-306
	-210	-100
	-301	-114
	-301	-115
	-303	-201
	-306	-840
	-306	-842
	-307	-106
	-307	-806
	-308	-107
	-310	-100
	-310	-102
	-310	-103
	-310	-105
	-310	-116
	-310	-841
	-314	-403
	-314	-404
	-314	-405
	-314	-406
	-314	-407
	-314	-408
	-314	-409
	-314	-410
	-314	-411
	-314	-412
	-314	-413
	-314	-414
	-314	-415
	-314	-416
	-314	-417
	-314	-419
	-315	-841
	-504	-822
	-505	-820
	-513	-800
	-820	-121
	-820	-122
	-820	-123
	-901	-105
	-901	-108
	-901	-111
	-901	-815

Module	SQLC	RDSC
ARIPPRP	-104	-801
	-106	-801
	-107	-860
	-108	-860
	-109	-808
	-210	-100
	-306	-840
	-307	-806
	-308	-804
	-310	-841
	-314	-801
	-314	-803
	-314	-804
	-314	-805
	-314	-806
	-314	-807
	-314	-808
	-314	-809
	-314	-810
	-314	-811
	-314	-812
	-314	-813
	-314	-814
	-314	-815
	-314	-816
	-314	-817
	-314	-818
	-315	-841
	-504	-822
	-505	-820
	-513	-800
ARIPRDI	-934	-000
ARIPSQA	-101	-820
	-104	-800

Module	SQLC	RDSC
ARIPSQC	-101	-100
	-101	-101
	-101	-102
	-102	-111
	-164	-115
	-166	-113
	-166	-114
	-167	-111
	-169	-112
	-169	-115
	-170	-115
	-314	-421
	-317	-111
	-317	-112
	-322	-108
	-323	-109
	-822	-103
	-822	-104
	-822	-106
	-822	-107
ARIPSQP	-101	-820
	-102	-112
	-104	-801
	-164	-111
	-166	-103
	-166	-106
	-166	-114
	-167	-113
	-170	-101
	-170	-102
	-170	-104
	-170	-105
	-170	-107
	-170	-108
	-170	-109
	-170	-110
	-170	-115
ARIRBRM	-932	-004
	-932	-008
	-932	-012
	-932	-016
	-933	-004
	-933	-008
	-933	-012
	-933	-016
	-935	-000
	-938	-000
	-939	-000

Module	SQLC	RDSC
	-940	-000
	-941	-004
	-941	-008
	-941	-012
	-941	-016
ARIRENA	-936	-000
ARIRIRM	-934	-000
	-941	-004
	-941	-008
	-941	-012
	-941	-016
ARIRORM	-804	-000
	-932	-004
	-932	-008
	-932	-012
	-932	-016
	-933	-004
	-933	-008
	-933	-012
	-933	-016
	-935	-000
	-937	-000
	-939	-000
	-942	-000
ARIRVIR	-824	-000
	-914	-000
	-932	-004
	-932	-008
	-932	-012
	-932	-016
	-933	-004
	-933	-008
	-933	-012
	-933	-016
	-934	-000
	-938	-000
	-939	-000
ARIRVRM	-804	-000
	-824	-000
	-914	-000
	-916	-***
	-932	-004
	-932	-008
	-932	-012

* SQLCODE depends upon DBSS error

Module	SQLC	RDSC
	-932	-016
	-933	-004
	-933	-008
	-933	-012
	-933	-016
	-935	-000
	-940	-000
	-941	-004
	-941	-008
	-941	-012
	-941	-016
ARIRVST	-932	-004
	-932	-008
	-932	-012
	-932	-016
	-934	-000
ARIXA01	-560	-090
	-560	-130
	-561	-110
ARIXA02	-901	-050
ARIXA07	-204	-120
	-204	-180
	-551	-200
	-552	-170
	-554	-110
	-557	-140
	-557	-160
	-557	-210
	-561	-163
	-562	-161
	-901	-220
ARIXA09	+150	-325
	-204	-100
	-204	-330
	-552	-200
	-555	-130
	-555	-210
	-555	-350
	-556	-140
	-556	-220
	-556	-231
	-556	-235
	-556	-360
	-557	-120
ARIXC01	-101	-001
ARIXC02	-901	-001
	-901	-002
	-901	-003
ARIXC03	-901	-001
	-901	-002
ARIXC05	-901	-001
	-901	-002
	-901	-003
	-901	-004
	-901	-005
	-901	-006
	-901	-007
	-901	-008
	-901	-009
	-901	-010
ARIXC06	-901	-001
	-901	-002
	-901	-003
	-901	-004
ARIXC07	-901	-001
ARIXC10	-101	-001
ARIXC13	-930	-001
	-930	-002
	-930	-003
ARIXC14	-901	-001
	-931	-002
	-931	-003
	-931	-004
ARIXC15	-101	-001
ARIXC18	-101	-003
	-101	-006
	-901	-001
	-901	-002
	-901	-004
	-901	-005
ARIXC19	-101	-001
	-930	-002
	-931	-003
ARIXC20	-901	-001
	-901	-002
ARIXC21	-101	-003
	-901	-001
	-901	-002
	-930	-004
	-931	-005
ARIXC22	-901	-001
ARIXC23	-901	-001
ARIXC24	-901	-002
	-901	-001
ARIXC27	-901	-001
	-901	-002
ARIXC28	-901	-001
	-901	-002
ARIXC29	-901	-001
	-901	-002
ARIXC30	-901	-002
	-901	-001
ARIXC31	-101	-001
ARIXC32	-101	-001
	-901	-002
ARIXC33	-901	-001
ARIXC34	-901	-001
ARIXC36	-901	-001
	-901	-002
	-901	-003
	-901	-004
	-901	-005

* SQLCODE depends upon DBSS error

Module	SQLC	RDSC
ARIXC37	-101	-102
	-101	-105
	-901	-101
	-930	-001
	-931	-005
ARIXC38	-101	-103
	-516	-001
	-901	-104
ARIXC39	-901	-001
	-901	-002
ARIXC40	-101	-001
ARIXC41	-101	-001
ARIXC42	-101	-001
ARIXC43	-101	-001
ARIXC45	-901	-001
	-901	-002
ARIXC47	-901	-001
ARIXC48	-901	-001
	-901	-002
ARIXC51	-901	-001
ARIXEAB	-901	-100
	-901	-104
	-930	-102
ARIXEAD	-903	-109
	-930	-100
	-930	-102
	-930	-103
	-930	-110
	-931	-104
	-931	-105
	-931	-106
	-931	-107
	-931	-108
	-931	-111
	-931	-112
	*	-101

Module	SQLC	RDSC
ARIXEBR	-802	-907
	-802	-908
	-802	-909
	-802	-910
	-802	-911
	-802	-912
	-802	-913
	-802	-914
	-802	-915
	-901	-100
	-901	-901
	-901	-902
	-901	-903
	-901	-904
	-901	-905
	-901	-906
ARIXEDR	-826	-100
	-826	-101
	*	-102
	*	-103
	*	-104
	*	-105
	*	-106
	*	-107
	*	-108
ARIXECL	-501	-100
ARIXEDP	-101	-100
	-312	-110
	-821	-100
	-901	-100
	-901	-101
	-901	-102
	-901	-120
	-901	-130
	-930	-100
	-931	-100
ARIXEDS	-930	-102
	-931	-101
ARIXEFB	-901	-100
	-931	-100

Module	SQLC	RDSC
ARIXELX	-551	-913
	-566	-810
	-805	-807
	-805	-820
	-806	-808
	-806	-821
	-827	-823
	-830	-811
	-901	-900
	-901	-914
	-930	-100
	-930	-101
	-930	-102
	-930	-103
	-930	-104
	-930	-105
	-930	-106
	-931	-107
	-931	-108
	-931	-109
	-931	-110
	-931	-111
	*	-812
	*	-822
ARIXEPH	-901	-101
ARIXEPP	-306	-805
	-312	-833
	-901	-100
	-901	-101
	-901	-300
	-901	-830
	-901	-831
	-901	-832
	-930	-100
	-931	-100
ARIXERD	+200	-273
	-162	-296
	-210	-272
	-210	-281
	-512	-240
	-512	-250
	-517	-274
	-518	-278
	-551	-913
	-563	-301
	-565	-280
	-756	-271

* SQLCODE depends upon DBSS error

Module	SQLC	RDSC
	-757	-291
	-758	-284
	-760	-293
	-804	-270
	-807	-210
	-807	-800
	-807	-820
	-826	-276
	-826	-277
	-826	-285
	-826	-286
	-826	-287
	-826	-288
	-826	-289
	-826	-290
	-827	-283
	-828	-282
	-831	-279
	-832	-292
	-832	-294
	-901	-220
	-901	-230
	-901	-275
	-930	-100
	-931	-100
	-931	-295
	-931	-399
ARIXEPH	-901	-100
ARIXEUH	-901	-100
ARIXERP	-901	-100
	-901	-101

Module	SQLC	RDSC
ARIXESX	-618	-947
	-759	-956
	-761	-957
	-827	-950
	-829	-949
	-930	-101
	-930	-103
	-930	-106
	-931	-101
	-931	-102
	-931	-104
	-931	-105
	-946	-892
	-946	-893
	-946	-894
	-946	-953
	*	-914
	*	-948
	*	-951
	*	-952
	*	-954
	*	-955
ARIXF02	-804	-001
ARIXF20	-508	-001
ARIXF21	-507	-001
	-508	-001
	-509	-001
	-510	-001
ARIXF47	-503	-001
	-507	-001
	-508	-001
	-509	-001
	-510	-001
ARIXF62	-901	-001
	-901	-002
	-901	-003
ARIXF81	-901	-001
	-901	-002
	-901	-003
ARIXF97	-313	-001

Module	SQLC	RDSC
ARIXI01	-601	-130
	-704	-150
	-704	-170
	-705	-100
	-705	-120
	-705	-210
	-705	-220
ARIXIST	-191	-200
	-901	-155
	-901	-167
	-901	-171
	-930	-100
	-930	-102
ARIXI02	-752	-101
ARIXI03	-204	-140
	-609	-160
	-705	-100
	-705	-110
	-705	-130
	-705	-170
	-705	-185
	-708	-139
ARIXI04	-606	-101
	-901	-105
	-901	-200
	-901	-201
ARIXI06	-154	-140
	-204	-120
	-601	-100
	-602	-200
	-614	-170
	-614	-180
	-705	-190
	-803	-210
	-901	-150
	-901	-160
ARIXI07	-552	-185
	-601	-110
	-601	-130
	-602	-100
	-602	-240
	-608	-150
	-608	-170

* SQLCODE depends upon DBSS error

Module	SQLC	RDSC
	-610	-190
	-612	-310
	-901	-210
	-901	-230
ARIXI08	-601	-100
	-601	-107
ARIXI09	-551	-105
	-551	-106
	-551	-107
	-551	-112
	-551	-114
	-551	-150
	-601	-102
	-601	-119
	-601	-123
	-602	-100
	-602	-150
	-901	-101
	-901	-104
	-901	-108
	-901	-109
	-901	-110
	-901	-111
	-901	-113
	-901	-116
	-901	-117
	-901	-121
ARIXI10	-204	-120
	-204	-170
	-204	-190
	-204	-220
	-204	-240
	-204	-320
	-204	-410
	-204	-570
	-210	-106
	-324	-109
	-551	-108
	-708	-310
	-709	-340
	-758	-107
	-901	-100
	-901	-105
	-901	-110
	-901	-170

Module	SQLC	RDSC
	-901	-180
	-901	-460
	-901	-560
ARIXI12	-751	-100
ARIXI13	-154	-140
	-204	-120
	-601	-280
	-602	-200
	-611	-190
ARIXI14	-901	-100
ARIXI15	-204	-120
	-204	-140
	-813	-160
	-814	-170
	-901	-100
	-901	-110
	-915	-180
ARIXI19	-901	-113
	-901	-201
ARIXI20	-901	-101
	-901	-102
	-901	-103
	-901	-104
	-901	-105
	-901	-108
	-901	-109
	-901	-110
	-901	-111
	-901	-112
	-901	-113
	-901	-114
	-901	-115
ARIXI33	-204	-150
	-608	-300
	-657	-250
	-930	-100
	-931	-450
	-931	-600
	-931	-650
ARIXOAF	-901	-360

Module	SQLC	RDSC
ARIXOBY	-901	-100
ARIXOB2	-101	-102
	-101	-103
	-901	-118
ARIXOCA	-204	-110
ARIXOCK	-202	-104
	-203	-101
	-205	-102
	-205	-103
	-901	-100
	-901	-120
	-901	-360
	-921	-100
ARIXOCS	-930	-101
ARIXOCT	-901	-111
	-901	-112
	-901	-113
ARIXOCY	-901	-100
ARIXODF	-901	-120
	-901	-130
	-901	-140
	-901	-142
	-901	-148
	-901	-150
	-901	-152
ARIXODM	-121	-118
	-121	-119
	-407	-114
	-901	-111
	-901	-112
	-901	-115
	-901	-116
ARIXOD1	-204	-510
	-619	-998
	-901	-120
	-901	-990
	-930	-100

* SQLCODE depends upon DBSS error

Module	SQLC	RDSC
ARIXOD2	-619	-996
	-619	-997
	-619	-999
	-901	-130
	-930	-110
ARIXOEX	-112	-155
	-112	-170
	-120	-110
	-120	-120
	-120	-176
	-127	-105
	-128	-127
	-133	-115
	-134	-215
	-401	-125
	-401	-185
	-401	-220
	-401	-225
	-401	-235
	-401	-244
	-401	-245
	-401	-258
	-401	-259
	-401	-260
	-401	-261
	-401	-262
	-401	-263
	-401	-264
	-401	-265
	-402	-100
	-402	-135
	-402	-145
	-402	-180
	-403	-175
	-409	-165
	-411	-140
	-411	-210
	-417	-251
	-417	-252
	-421	-257
	-422	-208
	-901	-150
	-901	-160
	-901	-170
	-901	-130
	-901	-190
	-901	-195
	-901	-200
	-901	-205
ARIXOFC	-101	-102
	-901	-101
ARIXOFE	-101	-103
	-901	-102
ARIXOFF	-121	-104
	-135	-105
	-407	-103
	-408	-102
	-408	-106
	-408	-107
	-408	-108
	-408	-109
ARIXOFP	-101	-100
ARIXOFQ	-101	-101
ARIXOFR	-901	-111
	-901	-112
ARIXOFT	-101	-102
	-901	-101
ARIXOGA	-901	-101
	-901	-100
	-901	-102
	-901	-103
	-901	-104
ARIXOGB	-119	-200
ARIXOGC	-118	-115
	-118	-116
	-118	-117
	-901	-111
	-901	-120
	-901	-121
ARIXOGP	-124	-302
	-125	-303
	-125	-305
	-126	-304
	-134	-106
	-511	-110
	-901	-450
	-901	-451
	-901	-452
	-930	-450
	-930	-451
	-931	-452
	-931	-453
ARIXOIN	-117	-106
	-117	-107
	-117	-108
	-551	-100
	-901	-110
	-901	-101
	-901	-102
	-901	-103
	-901	-104
	-901	-105
ARIXOIS	-901	-100
	-901	-101
	-901	-104
ARIXOLC	-901	-102
ARIXOLF	-134	-100
	-134	-103
	-134	-105
	-901	-104
	-901	-101
	-901	-102
ARIXOMA	-901	-102
	-901	-103
	-901	-104
	-901	-105
	-901	-106
ARIXOMD	-901	-101
	-901	-102
ARIXOML	-101	-101
	-901	-102
	-901	-103
	-901	-104
	-901	-360
	-901	-360

* SQLCODE depends upon DBSS error

Module	SQLC	RDSC
ARIXOMS	-901	-105
	-901	-100
	-901	-101
	-901	-102
	-901	-103
	-901	-104
	-901	-106
	-901	-200
	-901	-201
	-901	-202
	-901	-203
ARIXONV	-125	-100
	-208	-120
	-209	-110
	-901	-130
ARIXOOP	-511	-107
	-551	-103
	-551	-110
	-551	-115
	-551	-120
	-551	-125
	-551	-127
	-551	-131
	-551	-140
	-551	-160
	-551	-161
	-551	-200
	-551	-201
	-552	-102
	-552	-116
	-552	-126
	-552	-130
	-901	-100
	-901	-121
	-901	-150
	-823	-151
	-901	-101
	-901	-104
	-901	-105
	-901	-106
	-901	-360
ARIXOOS	-111	-150
	-112	-130
	-112	-155
	-127	-135
	-134	-120
	-401	-105
	-401	-125
	-401	-126
	-401	-170
	-402	-110
	-402	-115
	-402	-165
	-403	-160
	-408	-233
	-409	-145
	-411	-210
	-418	-121
	-901	-140
	-901	-155
	-901	-175
	-901	-180
	-901	-185
	-901	-230
	-901	-231
ARIXOQ1	-901	-102
	-901	-103
	-901	-104
	-901	-105
ARIXOSC	-551	-100
ARIXOSL	-901	-113
ARIXOSO	-415	-103
	-416	-103
	-901	-100
	-901	-100
	-901	-101
	-901	-102
ARIXOSR	-901	-101
	-901	-102
	-901	-104
	-901	-360
ARIXOST	-901	-116
ARIXOSU	-901	-102
	-901	-103
	-901	-104
	-901	-360
ARIXOS3	-101	-101
ARIXOS4	-101	-101
ARIXOTF	-401	-104
	-404	-109
	-404	-110
	-405	-101
	-405	-101
	-405	-101
	-410	-100
	-410	-100
	-410	-100
	-411	-112
	-411	-113
	-421	-105
	-901	-102
	-901	-103
	-901	-106
	-901	-107
	-901	-108
	-901	-111
	-901	-130
	-901	-200
ARIXOTS	-930	-101
ARIXOVC	-150	-140
	-151	-210
	-152	-240
	-154	-150
	-154	-220
	-155	-180
	-901	-100
	-901	-110
	-901	-120
	-901	-130
	-901	-160
	-901	-170
	-901	-190
	-901	-200
	-901	-230
	-901	-250
	-901	-260
	-901	-310
	-901	-320
	-901	-330
	-901	-340
	-901	-350

* SQLCODE depends upon DBSS error

Module	SQLC	RDSC
	-901	-360
ARIXOVD	-119	-143
	-119	-147
	-153	-130
	-154	-100
	-154	-110
	-154	-120
	-158	-200
	-158	-210
	-551	-150
	-602	-200
	-901	-140
	-901	-160
	-901	-170
	-901	-180
	-901	-190
	-930	-444
	-931	-445
	-931	-446
	-931	-447
	-931	-448
	-931	-449
ARIXOV1	-901	-260
	-901	-270
	-901	-280
	-901	-290
	-901	-300
	-901	-330
	-901	-340
ARIXOW1	-115	-106
	-901	-107
	-901	-116
	-901	-102
	-901	-104
	-901	-117
ARIXOW2	-401	-110
	-401	-209
	-401	-213
	-401	-214
	-401	-215
	-412	-109
	-414	-104
	-417	-100
	-417	-105
	-421	-207

Module	SQLC	RDSC
	-422	-210
	-422	-211
	-423	-212
	-901	-102
	-901	-108
	-901	-111
	-901	-113
	-901	-101
	-901	-103
	-901	-106
	-901	-107
	-901	-112
ARIXO1S	-901	-103
ARIXPA1	-101	-501
	-102	-302
	-102	-307
	-102	-309
	-103	-304
	-104	-303
	-104	-304
	-104	-311
	-105	-305
	-106	-308
	-107	-301
	-107	-306
	-130	-312
	-164	-314
	-165	-322
	-166	-315
	-167	-321
	-170	-319
	-170	-320
	-604	-317
	-705	-318
	-930	-100
	-931	-100
ARIXPA2	-104	-502
	-108	-307
	-129	-501
	-159	-312
	-160	-313
	-163	-503
	-168	-311
	-201	-309
	-211	-320
	-411	-319

Module	SQLC	RDSC
ARIXPA3	-108	-307
	-115	-314
	-125	-318
	-129	-501
	-132	-315
	-201	-310
	-211	-320
	-604	-316
	-604	-317
ARIXPRC	-301	-117
ARIXSUT	-901	-100
DBSS	-701	-077
	-702	-078
	-703	-076
	-710	-098
	-751	-060
	-754	-062
	-803	-034
	-809	-036
	-902	-1000
	-903	-064
	-903	-091
	-911	-099
	-911	-101
	-912	-100
	-914	-123
	-915	-097
	-916	-***
GENCODE		
(ARIXCRA)	-712	-001
	-901	-001
(ARIXCRB)	-712	-001
	-901	-001
(ARIXCRC)	-901	-001
(ARIXCRE)	-301	-001
	-309	-001
	-407	-001
	-901	-001
(ARIXCRH)	-301	-001
	-302	-001
	-309	-001
(ARIXCRO)	-901	-001

* SQLCODE depends upon DBSS error

Module	SQLC	RDSC
(ARIXCR2)	-303	-001
	-304	-001
	-305	-001
	-901	-002
	-930	-001
	-930	-004
	-931	-001
	-931	-006
(ARIXCR3)	-930	-001
	-930	-004
	-931	-001
	-931	-005
(ARIXCR4)	-504	-001
(ARIXCR5)	-301	-001
	-302	-001
	-309	-001
	-407	-001
	-901	-001
(ARIXCR6)	-301	-001
	-309	-001
	-302	-001
	-407	-001
	-901	-001
(ARIXCR7)	-301	-001
	-302	-001
	-309	-001
	-311	-001
	-407	-001
	-901	-001

Module	SQLC	RDSC
(ARIXCR8)	-301	-001
	-302	-001
	-309	-001
	-311	-001
	-407	-001
(ARIXC04)	-407	-001
(ARIXC35)	-407	-001
(ARIXC36)	-304	-001
	-305	-001
	-406	-001
	-413	-001
(ARIXF02)	-804	-001
(ARIXF03)	-501	-001
	-502	-001
(ARIXF19)	-901	-001
(ARIXF20)	-508	-001
	-901	-001
(ARIXF21)	-507	-001
	-508	-001
	-509	-001
	-510	-001
	-901	-001

Module	SQLC	RDSC
(ARIXF24)	-810	-001
(ARIXF25)	-901	-001
(ARIXF26)	-508	-001
(ARIXF27)	-901	-001
(ARIXF29)	-901	-001
(ARIXF30)	-901	-001
(ARIXF47)	-503	-001
	-507	-001
	-508	-001
	-509	-001
	-510	-001
(ARIXF48)	-901	-001
(ARIXF49)	-901	-001
(ARIXF57)	-901	-001
(ARIXF62)	-901	-001
	-901	-002
(ARIXF75)	-901	-001
(ARIXF77)	-901	-001
(ARIXF81)	-901	-001
	-901	-003
(ARIXF97)	-313	-001

* SQLCODE depends upon DBSS error

MESSAGE AND MODULE CROSS REFERENCES

This subsection consists of two major lists: "Message IDs in Sequence - Issuing Modules" and "Modules in Sequence - Messages Issued". These refer to the modules causing the messages to be printed. The modules containing the messages (both message ID and text) are:

ARIM000: Messages ARI000 - ARI099
ARIM200: Messages ARI200 - ARI299
ARIRMSG: Messages ARI400 - ARI499
ARIM500: Messages ARI500 - ARI519
ARIM520: Messages ARI520 - ARI535
ARIPMSM: Messages ARI536 - ARI599
ARI0MSGC: EXEC message ARI043I
ARI6MSGC: EXEC messages ARI600 - ARI699
ARIM800: Messages ARI800 - ARI899
ARISDBM: Messages ARI900 - ARI999
*ARIIMG1: Messages ARI7000 - ARI7019
*ARIIMG2: Messages ARI7020 - ARI7039
*ARIIMG3: Messages ARI7040 - ARI7059
*ARIIMG4: Messages ARI7060 - ARI7079
*ARIIMG5: Messages ARI7080 - ARI7099
*ARIIMG6: Messages ARI7100 - ARI7119
*ARIIMG7: Messages ARI7120 - ARI7139
*ARIIMG8: Messages ARI7140 - ARI7159
*ARIIMG9: Messages ARI7160 - ARI7179
*ARIIMG10: Messages ARI7180 - ARI7199
*ARIIMG11: Messages ARI7200 - ARI7299
*ARIIMG12: Messages ARI7300 - ARI7349
*ARIIMG13: Messages ARI7350 - ARI7449
*ARIIMG14: Messages ARI7450 - ARI7549
*ARIIMG15: Messages ARI7550 - ARI7599
*ARIIMG16: Messages ARI7600 - ARI7699
*ARIIMG17: Messages ARI7700 - ARI7799
*ARIIMG18: Messages ARI7865 - ARI7924
*ARIIMG19: Messages ARI7925 - ARI7999
*ARIISTM: Status areas and messages not issued via message services

* These are macros contained in module ARIM700 or ARIISTX

MESSAGE IDS IN SEQUENCE - ISSUING MODULES

Message ID	Issuing Module(s)
ARI001E	ARICCRA
	ARICIP1
	ARICMUD
	ARICSTK
	ARISDBR
	ARISFDB
	ARISFM1
	ARISIIO
	ARIYC11
	ARIYD69
	ARIYD81
	ARIYI22
	ARIYI36
	ARIYK12
	ARIYL08
	ARIYL13
	ARIYL21
	ARIYL23
	ARIYS08
	ARIYX04
	ARIYZ00
ARI002E	ARICPRM
	ARISPFM
ARI003E	ARICPRM
ARI004I	ARICPRM
ARI005E	ARICPRM
ARI006E	ARICPRM
ARI007E	ARICPRM
ARI008E	ARICPRM
ARI009E	ARICPRM
ARI010E	ARICPRM
ARI011E	ARICPRM
ARI012I	ARICPRM
ARI013E	ARICPRM
ARI014E	ARICPRM
ARI015I	ARICPRM
ARI016I	ARICPRM
ARI017E	ARICPRM
ARI018I	ARICPRM
ARI019E	ARICPRM
ARI025I	ARICIP1
	ARICSPM
ARI026E	ARICIP1
	ARICSPM

Message ID	Issuing Module(s)
ARI027E	ARICCLA
	ARICENA
	ARICIP1
	ARICIP2
	ARICMUD
	ARICOMB
	ARICSHT
ARI028I	ARICSHT
ARI029I	ARICCLA
	ARICSHT
ARI030I	ARICSHT
ARI031E	ARICSHT
ARI032I	ARICTRM
ARI033E	ARICTRM
ARI034E	ARICTRM
ARI035I	ARICABE
ARI036E	ARICABE
ARI037E	ARICABE
ARI038E	ARICABE
ARI039E	ARIYM02
ARI040E	ARICABE(C
	ARICIP1
	ARIYM02
ARI041E	ARIYM02
ARI042I	ARICIP1
	ARICTRM
	ARISYSD (EP
	ARISYSDG)
ARI043I	ARICIP1
	ARICTRM
	ARISYSD (EP
	ARISYSDG)
	CMS EXECs:
	SQLADBEX
	SQLADBSP
	SQLGENLD
	SQLINIT
	SQLLOG
	SQLSTART
ARI044D	ARICABE
ARI045I	ARIYM01
ARI046I	ARICSPM
ARI047E	ARICIP1
	ARICSPM
ARI053E	ARIXERO
ARI054E	ARIXERO
ARI055A	ARICTRI
ARI056I	ARICTRI

Message ID	Issuing Module(s)
ARI057E	ARIIMSG
	ARIIRWI
	ARIYM04
	ARISERR
ARI058E	ARIYM05
	ARIMFMT
	ARIIFOR
ARI059E	ARISYSD (EP
	ARISYSDG)
ARI060I	ARIYM01
ARI061E	ARIYM10
ARI062A	ARIYM10
ARI063I	ARIYM10
ARI064E	ARIYM10
ARI065I	ARIYM10
ARI066E	ARIYM10
ARI067E	ARIYM11
	ARIYT01
ARI068E	ARISYSD(D (EP
	ARISYSD5)
ARI069E	ARISYSE (EP
	ARISYSD5)
ARI070A	ARISYSE (EP
	ARISYSD5)
ARI072D	ARISYSE (EP
	ARISYSD5)
	ARISYSIN (CMS
	EXEC)
ARI081E	ARIYE14
	ARIXETR
ARI082E	ARICTRC
ARI083I	ARICTRC
ARI084D	ARICTRC
ARI085E	ARICTRC
ARI086E	ARICTRC
	ARICTRI
ARI087D	ARICTRC
ARI088D	ARICTRC
ARI089E	ARICTRC
ARI090D	ARICTRC
ARI091D	ARICTRC
ARI093I	ARICTRC
ARI094E	ARICTRC
	ARICTRI
ARI095I	ARICTRC
ARI096I	ARICTRC
ARI097I	ARICTRC
ARI098E	ARICTRC
ARI099I	ARICTRC

Message ID	Issuing Module(s)
ARI200E	ARISPFM
	ARIYL14
ARI201E	ARIYI20
	ARIYI29
ARI202I	ARIYI29
	ARIYI51
ARI203E	ARIYI14
	ARIYI19
ARI204D	ARIYT00
ARI220E	ARIYT18
ARI221E	ARIYT18
ARI222E	ARIYT18
ARI223I	ARIYT18
ARI224E	ARIYT18
ARI225E	ARIYT18
ARI226I	ARIYT18
ARI227E	ARIYT18
ARI228E	ARIYT18
ARI229E	ARIYT18
ARI230I	ARIYT16
ARI231E	ARIYT29
ARI232E	ARIYT29
ARI248E	ARIYZ19
ARI249E	ARIYZ19
ARI276E	ARIYL01
ARI280A	ARIYL13
ARI281I	ARIYL13
ARI282I	ARIYL13
ARI283I	ARIYL13
ARI284I	ARIYL15
ARI285I	ARIYL00
ARI286I	ARIYL21
ARI287I	ARIYL24
ARI288I	ARIYL24
ARI289I	ARIYL21
ARI290I	ARIYL22
ARI291I	ARIYL22
ARI292I	ARIYL23
ARI293I	ARIYL23
ARI294A	ARIYL21
	ARIYL22
	ARIYL23
ARI295A	ARIYL21
ARI296A	ARIYL21
	ARIYL22
	ARIYL23
ARI297A	ARIYL21
	ARIYL23

Message ID	Issuing Module(s)
ARI298E	ARIYL22
	ARIYL23
ARI299A	ARIYL23
ARI400E	ARIRENA
	ARIRDIS
ARI401I	ARIRENA
ARI402E	ARIRENA
ARI403E	ARIRENA
	ARIRDIS
ARI404E	ARIRENA
	ARIRDIS
ARI405E	ARIRENA
ARI406E	ARIRENA
	ARIRDIS
ARI407E	ARIRENA
	ARIRDIS
ARI408E	ARIRENA
ARI410I	ARIRENA
ARI411I	ARIRDIS
ARI412I	ARIRDIS
ARI413I	ARIRDIS
ARI414A	ARIRDIS
ARI415D	ARIBRM
ARI416E	ARIBRM
ARI417I	ARIRBRM
ARI418A	ARIRENA
ARI419A	ARIRENA
ARI420E	ARIRENA
ARI421I	ARIRDIS
ARI422E	ARIRORM
ARI423A	ARIRENA
ARI424I	ARIRENA
	ARIRMSG
	ARIRORM
ARI425I	ARIRDIS
ARI426I	ARIRDIS
ARI427A	ARIRDIS
ARI428E	ARIRVRM
ARI429E	ARIRVRM
ARI430I	ARIRDIS
ARI431E	ARIRDIS
ARI432E	ARIRDIS
ARI433E	ARIRDIS
ARI434E	ARIRDIS
ARI436E	ARIRORM
	ARIRDIS
ARI437E	ARIRENA
ARI438E	ARIRORM
	ARIRDIS

Message ID	Issuing Module(s)
ARI1450I	ARIRORMD
ARI1500I	ARIDSQLA
	ARIIPSQ
ARI1501I	ARIDSQLA
	ARIIPSQ
ARI1502I	ARIDSQLA
	ARIIPSQ
ARI1503E	ARIDSQLA
	ARIIBIN
	ARIICAN
	ARIICI2D
	ARIICMD
	ARIIERS
	ARIIHLP
	ARIILST
	ARIIPQY
	ARIIPSQ
	ARIIREC
	ARIIRNM
	ARIIRUN
	ARIISCC
	ARIISTR
	ARIIST2
	ARIPPRA
	ARIPPRP
	ARIPLTP
	ARIPPRC
	ARIPSQA
	ARIPSQC
	ARIPSQP
	ARIPTXC
ARI1504I	ARIDSQLA
	ARIIBIN
	ARIICAN
	ARIICI2D
	ARIICMD
	ARIIERS
	ARIIHLP
	ARIILST
	ARIIPQY
	ARIIPSQ
	ARIIREC
	ARIIRNM
	ARIIRUN
	ARIISCC
	ARIISTR
	ARIIST2
	ARIPLTP
	ARIPPRC
	ARIPSQA
	ARIPSQC
	ARIPSQP
	ARIPTXC
	ARIRORMD
	ARIPSQP
	ARIPTXC
ARI505I	ARIDSQLA
	ARIIBIN
	ARIICAN
	ARIICI2D
	ARIICMD
	ARIIERS
	ARIIHLP
	ARIILST
	ARIIPQY
	ARIIPSQ
	ARIIREC
	ARIIRNM
	ARIIRUN
	ARIISCC
	ARIISTR
	ARIIST2
	ARIPLTP
	ARIPPRC
	ARIPSQA
	ARIPSQC
	ARIPSQP
	ARIPTXC
ARI511E	ARIDSQLA
ARI520A	ARISIST
	ARISMAI
ARI521A	ARISDFR
	ARISIST
	ARISMAI
	ARISUPD
ARI522A	ARISDFL
	ARISDFM
	ARISDFR
	ARISEDI
	ARISFIL
	ARISINI
	ARISIST
	ARISIST2
	ARISMAI
	ARISSET
	ARISUPD
ARI523A	ARISDFL
	ARISDFM
	ARISDFR
	ARISEGB
	ARISFIL
	ARISINI
	ARISIST
	ARISIST2
	ARISMAI
	ARISUPD
ARI524A	ARISCOL
	ARISDFL
	ARISDFM
	ARISFIL
	ARISIST
	ARISIST2
	ARISUPD
ARI525A	ARISCNV
ARI526A	ARISMAI
ARI527A	ARISMAI
ARI528I	ARISMAI
ARI529I	ARISMAI
ARI530I	ARISMAI
ARI536I	ARIPPRF
ARI537E	ARIPPRF
	ARIPSQF
ARI537E	ARIPPRF
	ARIPSQA
	ARIPSQC
	ARIPSQF
	ARIPSQP
ARI539I	ARIPSCF
	ARIPSCN
ARI540E	ARIPPRF
	ARIPSQA
	ARIPSQC
	ARIPSQF
	ARIPSQP
ARI541E	ARIPSQA
	ARIPSQC
	ARIPSQF
	ARIPSQP
ARI542E	ARIPPRF
	ARIPSQA
	ARIPSQC
	ARIPSQP
ARI545E	ARIPSSF
ARI546I	ARIPPRF
ARI547I	ARIPPRA
	ARIPPRC
	ARIPPRF
	ARIPPRP
ARI548E	ARIPSSF
ARI549E	ARIPSSF
ARI550E	ARIPSSF
ARI551I	ARIPSCF
	ARIPSCN
ARI552I	ARIPSCF
	ARIPSCN
ARI553E	ARIPSSF
ARI554E	ARIPSSF
ARI555E	ARIPSSF
ARI556E	ARIPSSF
ARI557E	ARIPDCF
ARI558I	ARIPSQF
ARI559E	ARIPSQF
	ARIPSSF
ARI560I	ARIPPRA
	ARIPPRC
	ARIPPRF
	ARIPPRP
ARI561I	ARIPPRA
	ARIPPRC
	ARIPPRF
	ARIPPRP
ARI562I	ARIPPIFA
ARI563I	ARIPPRF
ARI564E	ARIPPRF
ARI565E	ARIPPRF
ARI566E	ARIPSSF
ARI567E	ARIPDCF
ARI568E	ARIPSQF
ARI569E	ARIPSQF
	ARIPSSF
ARI570I	ARIPPIFA
	ARIPPRA
	ARIPPRC
	ARIPPRP
ARI571I	ARIPPIFA
	ARIPPRA
	ARIPPRC
	ARIPPRP
ARI572E	ARIPPRA
	ARIPPRC
	ARIPPRP
	ARIPSSF
ARI574E	ARIPDCF
	ARIPPRA
	ARIPPRP
	ARIPSQF
ARI575E	ARIPPRA
	ARIPPRC
	ARIPPRP
	ARIPSSF
ARI576E	ARIPSQC
ARI577E	ARIPPRC
	ARIPPRF
ARI578I	ARIPPRF
ARI579I	ARIPPIFA
	ARIPPRA
	ARIPPRC
	ARIPPRF
	ARIPPRP
ARI580E	ARIPPRA
	ARIPPRC
	ARIPPRF
	ARIPPRP
	ARIPSCF
	ARIPSCN
ARI581E	ARIPSCF
	ARIPSCN
	ARIPSQF
ARI582E	ARIPPRA
	ARIPPRC
	ARIPPRP
ARI583E	ARIPSCF
	ARIPSCN
ARI584E	ARIPPRA
	ARIPPRC
	ARIPPRF
	ARIPPRP
	ARIPSQF
	ARIPTXA
	ARIPTXC
	ARIPTXF
	ARIPTXP
ARI586I	ARIPPRA
	ARIPPRC
	ARIPPRF
	ARIPPRP
ARI587I	ARIPPIFA
	ARIPPRA
	ARIPPRC
	ARIPPRP
ARI588E	ARIPSCF
	ARIPSCN
ARI589E	ARIPMSF
	ARIPMSH
ARI590E	ARIPPRA
	ARIPPRP
	ARIPSQF
ARI591E	ARIPPRA
	ARIPPRP
	ARIPSQF
ARI592E	ARIPPRA

Message ID	Issuing Module(s)
ARI593E	ARIPPRA
ARI594E	ARIPSQC
ARI595E	ARIPSQC
ARI596E	ARIPSQC
ARI597E	ARIPPRA
	ARIPPRP
	ARIPSQF
ARI598I	ARIPPRA
	ARIPPRC
	ARIPPRF
	ARIPPRP
ARI599E	ARIPPRC
Start CMS EXEC Messages	
ARI600A	SQLGENLD
	SQLINIT
ARI601E	SQLADBEX
	SQLADBSP
	SQLDBGEN
	SQLINIT
	SQLLOG
	SQLSTART
ARI602D	SQLGENLD
ARI603I	SQLDBID
ARI604A	SQLGENLD
ARI605E	SQLGENLD
	SQLLOG
ARI606E	SQLDBINS
	SQLGENLD
	SQLINIT
ARI607E	ARISPDFC
	SQLADBEX
	SQLDBGEN
	SQLDBINS
	SQLGENLD
	SQLLOG
	SQLSTART
ARI608E	SQLGENLD
ARI609E	SQLSTART
ARI611E	SQLSTART
ARI612E	SQLPREP
	SQLSTART
ARI613E	ARISAVES
	ARISDBMA
	ARISSERV
	SQLADBEX
	SQLADBSP
	SQLDBGEN
	SQLDBINS
	SQLLOG

Message ID	Issuing Module(s)
	SQLPREP
	SQLSTART
ARI614A	SQLADBEX
ARI615D	SQLINIT
ARI616E	SQLADBEX
	SQLADBSP
	SQLDBGEN
	SQLGENLD
	SQLLOG
	SQLSTART
ARI617E	SQLSTART
ARI618I	SQLINIT
ARI619E	SQLINIT
ARI620I	SQLGENLD
	SQLDBINS
	SQLLOG
ARI621E	SQLADBEX
	SQLDBGEN
	SQLLOG
ARI622I	SQLSTART
ARI624A	SQLGENLD
ARI625D	SQLGENLD
ARI626E	SQLDBINS
	SQLGENLD
	SQLINIT
ARI627E	ARISAVES
	ARISDBMA
	SQLADBEX
	SQLADBSP
	SQLDBGEN
	SQLDBINS
	SQLINIT
	SQLLOG
	SQLSTART
ARI628E	ARISAVES
	ARISDBMA
	SQLADBEX
	SQLADBSP
	SQLDBGEN
	SQLDBINS
	SQLINIT

Message ID	Issuing Module(s)
	SQLLOG
	SQLPREP
	SQLSTART
ARI630E	SQLINIT
	SQLSTART
ARI630I	SQLSTART
ARI631E	ARISPROD
	SQLADBEX
	SQLDBGEN
	SQLDBINS
	SQLLOG
	SQLSTART
ARI632D	SQLDBGEN
ARI633D	SQLDBGEN
ARI634D	SQLDBGEN
ARI635D	SQLDBGEN
ARI636D	SQLDBGEN
ARI637D	SQLDBGEN
ARI638D	SQLADBEX
	SQLADBSP
	SQLDBGEN
ARI639E	ARISPDFC
	SQLADBEX
	SQLADBSP
	SQLDBGEN
	SQLINIT
	SQLLOG
	SQLSTART
ARI640E	ARICEMGC
ARI641E	ARICEMGC
ARI642E	ARICEMGC
ARI643E	ARICEMGC
ARI644E	SQLADBEX
	SQLADBSP
	SQLDBGEN
	SQLLOG
ARI645E	SQLADBEX
	SQLADBSP
	SQLLOG
ARI646D	SQLADBSP
ARI647D	SQLADBEX
	SQLDBGEN
	SQLLOG
ARI648A	SQLADBEX
	SQLADBSP
ARI649A	SQLADBSP
ARI650E	ARISPDFC
	SQLINIT
ARI651E	ARISAVES
	ARISDBMA
	SQLDBGEN

Message ID	Issuing Module(s)
	SQLDBINS
	SQLGENLD
	SQLLOG
ARI652E	ARISAVES
	ARISLKIT
	ARISPDFC
	ARISPROD
	ARISSERV
	ARISSLKE
	SQLDBID
	SQLDBINS
ARI653E	ARISAVES
	ARISDBMA
	ARISLKIT
	ARISPROD
	ARISQLLD
	ARISSERV
	ARISSLKE
	SQLDBINS
	SQLINIT
	SQLSTART
ARI657I	ARISEMSG
ARI658E	ARISEMSG
	ARISESCP
	ARISLKIT
	ARISPROD
	ARISQLLD
	ARISSLKE
ARI659I	ARISESCP
ARI660I	ARISESCP
ARI661E	ARISAVES
	ARISDBMA
	SQLDBINS
ARI662I	ARISEMSG
ARI663I	SQLPREP
	SQLSTART
ARI664E	ARISDBMA
	SQLDBINS
ARI666E	ARISDBMA
ARI667E	ARISSLKE
ARI668E	ARISSLKE
ARI669E	ARISSLKE
ARI670E	ARISLKIT
ARI671E	ARISLKIT
ARI672E	ARISLKIT
ARI673I	ARISAVES
ARI675E	SQLPREP
ARI676E	SQLPREP
ARI677I	SQLPREP
ARI678E	SQLPREP

Message ID	Issuing Module(s)
ARI679E	SQLPREP
ARI680E	SQLPREP
ARI681E	SQLPREP
ARI682E	SQLLOG
ARI683E	SQLLOG
ARI684I	SQLLOG
ARI685I	SQLLOG
ARI686E	SQLLOG
ARI687E	SQLLOG
ARI692E	SQLISTRT
ARI693E	SQLISTRT
ARI694I	SQLISTRT
ARI696E	SQLISTRT
ARI697E	SQLISTRT
ARI698E	SQLISTRT
ARI699E	SQLISTRT
End CMS EXEC Messages	
ARI7000E	ARIICICD
ARI7001E	ARIICICD
ARI7020E	ARIISMG
ARI7021E	ARIISMG
ARI7022E	ARIIMSG
ARI7040I	ARIIDBS
ARI7041D	ARIIVFYC
	ARIIVFYD
ARI7042E	ARIICAN
ARI7043I	ARIICAN
ARI7044I	ARIICICD
ARI7045E	ARIICAN
ARI7046D	ARIICICD
ARI7047I	ARIICICD
	ARIIVFYC
	ARIIVFYD
ARI7048I	ARIICICD
	ARIIVFYC
	ARIIVFYD
ARI7049E	ARIISQL
ARI7050I	ARIIVFYC
ARI7051E	ARIIRWI
ARI7060I	ARIIRWI
ARI7061I	ARIIRWI
ARI7062E	ARIIRWI
ARI7063I	ARIIRWI
ARI7064I	ARIIRWI
ARI7065I	ARIIRWI
ARI7066I	ARIIRWI
ARI7068I	ARIIRWI
ARI7069I	ARIIHLD
ARI7080A	ARIIDBS
ARI7081A	ARIICI2D

Message ID	Issuing Module(s)
ARI7082A	ARIICI2D
ARI7083A	ARIICI2D
ARI7084I	ARIICI2D
ARI7085E	ARIICI2D
ARI7086E	ARIICI2D
ARI7087E	ARIICI2D
ARI7101E	ARIITIOC
ARI7102I	ARIIREAC
ARI7119E	ARIISYSC
ARI7120E	ARIISUB
ARI7121E	ARIISUB
ARI7123E	ARIISUB
ARI7124E	ARIISUB
ARI7125E	ARIISUB
ARI7126E	ARIISUB
ARI7127I	ARIIFOL
ARI7128E	ARIISUB
ARI7180E	ARIITRC
ARI7181I	ARIITRC
ARI7182E	ARIITRC
ARI7183E	ARIITRC
ARI7184I	ARIITRC
ARI7185I	ARIITRC
ARI7186I	ARIITRC
ARI7300E	ARIIBIN
ARI7301E	ARIIBIN
ARI7302E	ARIIBIN
ARI7303E	ARIIBIN
ARI7304E	ARIIBIN
ARI7305E	ARIIBIN
ARI7306E	ARIIBIN
ARI7307A	ARIIBIN
ARI7308A	ARIIBIN
ARI7309E	ARIIBIN
ARI7310E	ARIIBIN
ARI7311E	ARIIBIN
ARI7312I	ARIIBIN
ARI7313I	ARIIBIN
ARI7314I	ARIIBIN
ARI7315E	ARIIBIN
ARI7316E	ARIIBIN
ARI7317I	ARIIBIN
ARI7318E	ARIIBIN
ARI7319I	ARIIBIN
ARI7320I	ARIIBIN
ARI7321E	ARIIBIN
ARI7322I	ARIIBIN
ARI7323I	ARIIBIN
ARI7324E	ARIIBIN
ARI7325I	ARIIBIN
ARI7326A	ARIIBIN
ARI7327E	ARIIBIN
ARI7328E	ARIIBIN
ARI7329E	ARIIBIN
ARI7330E	ARIIBIN
ARI7350E	ARIIRUN
ARI7351E	ARIIRUN
ARI7354E	ARIIRUN
ARI7355E	ARIIRUN
ARI7357E	ARIIRUN
ARI7358E	ARIIRUN
ARI7359E	ARIIRUN
ARI7360E	ARIIRUN
ARI7361I	ARIIRUN
ARI7362E	ARIIRUN
ARI7363E	ARIIRUN
ARI7364E	ARIIRUN
ARI7398I	ARIIPRF
ARI7399I	ARIIPRF
ARI7500E	ARIIFMC
ARI7501E	ARIIFMC
ARI7502I	ARIIFCC
ARI7503I	ARIIFMC
	ARIIFMC
ARI7504E	ARIIFCC
	ARIIFMC
ARI7505E	ARIIFCC
	ARIIFMC
ARI7506I	ARIIFCC
ARI7507E	ARIIFMC
ARI7508E	ARIIFCC
	ARIIFCI
	ARIIFCS
	ARIIFMC
ARI7509I	ARIIFMC
ARI7510I	ARIIFMC
ARI7511I	ARIIFCC
	ARIIFMC
ARI7512I	ARIIFMC
ARI7513E	ARIIFCC
	ARIIFCI
	ARIIFCS
ARI7514E	ARIIFCS
ARI7515E	ARIIFCC
	ARIIFCS
ARI7516I	ARIIFCC
	ARIIFCI
	ARIIFCS
ARI7517E	ARIIFCC
ARI7518E	ARIIFCC
ARI7519E	ARIIFCC
ARI7520E	ARIIFCC
ARI7521I	ARIIFCI
ARI7522I	ARIIFCI
ARI7523E	ARIIFCC
	ARIIFCI
	ARIIFCS
ARI7524I	ARIIFMC
ARI7525I	ARIIFMC
ARI7526E	ARIIFCC
	ARIIFMC
ARI7527I	ARIIFMC
ARI7528I	ARIIFMC
ARI7529I	ARIIFMC
ARI7530E	ARIIFCI
	ARIIFCS
ARI7531I	ARIIFMC
ARI7532I	ARIIFCS
ARI7533I	ARIIFCS
ARI7534E	ARIIFCI
ARI7535E	ARIIFCI
ARI7536E	ARIIFCI
ARI7537E	ARIIFCI
	ARIIFCS
ARI7539I	ARIIFMC
ARI7543I	ARIIPQY
ARI7544I	ARIIPSQ
ARI7545E	ARIIFMC
ARI7546E	ARIIFMC
	ARIIFCC
ARI7547E	ARIIFCC
ARI7550D	ARIIERS
	ARIIHLP
	ARIILST
	ARIIREC
	ARIIRNM
	ARIISTR
ARI7551I	ARIIERS
	ARIIREC
	ARIIRNM
	ARIISTR
ARI7552E	ARIIERS
	ARIIRNM
	ARIISTR
ARI7553E	ARIIREC
	ARIIRNM
	ARIISTR
ARI7554E	ARIIREC
ARI7555E	ARIIREC
ARI7556E	ARIISTR
ARI7557I	ARIIREC
	ARIIRNM
	ARIISTR
ARI7558E	ARIIERS
	ARIILST
	ARIIREC
	ARIIRNM
	ARIISCC
	ARIISTR
ARI7559I	ARIIERS
	ARIIHLP
	ARIILST
	ARIIREC
	ARIIRNM
	ARIISTR
ARI7560E	ARIIREC
	ARIIRNM
ARI7561E	ARIIRNM
ARI7563E	ARIIRNM
ARI7564E	ARIIERS
	ARIIRNM
	ARIISTR
ARI7568E	ARIIRNM
	ARIISTR
ARI7569E	ARIIERS
	ARIIREC
	ARIISTR
ARI7570I	ARIIERS
	ARIIRNM
ARI7572E	ARIIRNM
	ARIISTR
ARI7574E	ARIIHLD
ARI7575E	ARIIHLD
ARI7576D	ARIISTR
ARI7577D	ARIISTR
ARI7578I	ARIISTR
ARI7579I	ARIISTR
ARI7580I	ARIISTR
ARI7581I	ARIISTR
ARI7582I	ARIISTR
ARI7583E	ARIISTR
ARI7585I	ARIIERS
ARI7590I	ARIIERS
	ARIILST
ARI7591I	ARIIERS
	ARIILST
ARI7592I	ARIIERS
	ARIILST
ARI7593I	ARIISTR
ARI7600E	AXTCENT
ARI7601I	ARIICMD
ARI7602D	ARIICMD
	ARIIST2
ARI7603I	ARIICMD
ARI7604E	ARIICMD
	ARIIST2
ARI7607I	ARIIIGN
	ARIIRMI
ARI7611E	ARIISCC
ARI7612E	ARIISCC
ARI7613I	ARIISCC
ARI7614E	ARIISCC
ARI7615E	ARIISCC
ARI7616I	ARIIHLP
	ARIILST
	ARIIQRY
	ARIISCC
	ARIIST2
	ARIIST3
ARI7617E	ARIIHLP
	ARIILST
	ARIISCC
	ARIISET
ARI7618E	ARIISCC
ARI7619E	ARIIRUN
	ARIISCC
ARI7620I	ARIILST
ARI7621I	ARIILST
ARI7622E	ARIICMD
ARI7623E	ARIICMD
ARI7624E	ARIIHLP
ARI7625E	ARIIHLP
ARI7626E	ARIIHLP
	ARIILST
	ARIISCC
ARI7627E	ARIISCC
ARI7628I	ARIISCC
ARI7629E	ARIISCC
ARI7630E	ARIISCC
ARI7631E	ARIISCC
ARI7632E	ARIICMD
ARI7633E	ARIICMD
ARI7634E	ARIICMD
ARI7635I	ARIICMD
ARI7636D	ARIICMD
ARI7637E	ARIICMD
ARI7638I	ARIISCC
ARI7639D	ARIICMD
ARI7640E	ARIINSQ
ARI7641D	ARIISCC
ARI7642I	ARIISCC
ARI7643E	ARIIHLP

Message ID	Issuing Module(s)
ARI7644E	ARIIHLP
ARI7645E	ARIICMD
ARI7646E	ARIIHLP
ARI7650E	ARIIEXT
ARI7651I	ARIICMD
ARI7652I	ARIICMD
ARI7700I	ARIIIGN
	ARIIHLP
	ARIIPQY
	ARIIRUN
	ARIISCC
ARI7701E	ARIILST
	ARIISET
	ARIIST2
ARI7702I	ARIIQRY
ARI7709I	ARIIQRY
ARI7710I	ARIIQRY
ARI7711I	ARIIQRY
	ARIISET
ARI7712I	ARIIQRY
ARI7713I	ARIIQRY
ARI7714I	ARIIQRY
ARI7715I	ARIIQRY
ARI7725E	ARIILST
	ARIIQRY
	ARIISET
	ARIIST2
	ARIIST3
ARI7726E	ARIISET
	ARIIST2
	ARIIST3
ARI7727I	ARIISET
	ARIIST2
ARI7729I	ARIIST3
ARI7731I	ARIIST2
ARI7732I	ARIIST2
ARI7733E	ARIISET
	ARIIST2
	ARIIST3
ARI7734E	ARIISET
	ARIIST2
ARI7735I	ARIISET
	ARIIST3
ARI7736E	ARIIST2
ARI7737I	ARIIST2
ARI7739E	ARIIST3
ARI7740E	ARIISET
	ARIIST3
ARI7742E	ARIIST2
ARI7743I	ARIIST3
ARI7744I	ARIIST3

Message ID	Issuing Module(s)
ARI7745I	ARIIST3
ARI7746I	ARIIST3
ARI7750E	ARIISET
ARI7751E	ARIISET
ARI7752I	ARIIQRY
ARI7753I	ARIIQRY
	ARIISET
ARI7754E	ARIISET
ARI7755I	ARIISET
	ARIIQRY
ARI7757I	ARIIST3
ARI7758I	ARIIST3
ARI7759I	ARIIQRY
ARI7760E	ARIISET
ARI7761I	ARIISET
ARI7865D	ARIIPRTD
ARI7866I	ARIIPRTD
ARI7867I	ARIIPRTD
ARI7868I	ARIIPRTD
ARI7869E	ARIIPRTD
ARI7870E	ARIIPRTD
ARI7871E	ARIIPRTD
ARI7872E	ARIIPRTD
ARI7873E	ARIIPRTD
ARI7874E	ARIIPRTD
ARI7875E	ARIIPRTD
ARI7876E	ARIIPRTD
ARI7877E	ARIIPRTD
ARI7878I	ARIIPRTD
ARI7879E	ARIIPRTD
ARI7882I	ARIIPRTD
ARI7883I	ARIIPRTD
ARI7884I	ARIIPRTD
ARI7885E	ARIIPRTC
ARI7886E	ARIIPRTC
ARI7887E	ARIIPRTC
ARI7888E	ARIIPRTC
ARI7889E	ARIIPRTC
ARI7890E	ARIIPRTC
ARI7891E	ARIIPRTC
ARI7892E	ARIIPRTC
ARI7893I	ARIIPRTD
ARI7894I	ARIIPRTC
ARI7895E	ARIIPRTC
ARI7896E	ARIIPRTC
ARI7897E	ARIIPRTC
ARI7898E	ARIIPRTC
ARI7901E	ARIIPRTC
	ARIIPRTD
ARI7902I	ARIIPRTC
	ARIIPRTD

Message ID	Issuing Module(s)
ARI7903E	ARIIPRTC
	ARIIPRTD
ARI7904I	ARIIPRTC
	ARIIPRTD
ARI7905I	ARIIPRTC
	ARIIPRTD
ARI7906I	ARIIPRTC
	ARIIPRTD
ARI7907I	ARIIPRTD
ARI7908I	ARIIPRTC
	ARIIPRTD
ARI7910E	ARIIPRTD
ARI7911E	ARIIPRTD
ARI7913E	ARIIPRTD
ARI7915E	ARIIPRTD
ARI7916E	ARIIPRTD
ARI7917E	ARIIPRTD
ARI7918E	ARIIPRTD
ARI7920E	ARIIPRTC
	ARIIPRTD
ARI7921E	ARIIPRTC
	ARIIPRTD
ARI7922I	ARIIPRTC
	ARIIPRTD
ARI7923D	ARIIPRTC
	ARIIPRTD
ARI7924I	ARIIPRTC
	ARIIPRTD
ARI7932I	ARIIPSQ
ARI7933I	ARIIPSQ
ARI7934I	ARIIPSQ
ARI7936E	ARIIPSQ
ARI7937E	ARIIPSQ
ARI7939E	ARIIDQY
ARI7940E	ARIIPQY
	ARIIRPT
ARI7942E	ARIIDQY
	ARIIPQY
ARI7943E	ARIIPQY
ARI7944E	ARIIPQY
ARI7945I	ARIIPSQ
ARI7946E	ARIISQL
ARI7947E	ARIISQL
ARI7949I	ARIIDQY
ARI7951I	ARIIDQY
ARI7952E	ARIIDQY
ARI7953E	ARIIDQY
ARI7954E	ARIIPQY
ARI7955I	ARIIPQY
ARI7956E	ARIIPQY
ARI7957E	ARIIDQY

Message ID	Issuing Module(s)
ARI7958I	ARIIPSQ
ARI7959I	ARIIPSQ
ARI7960I	ARIIPQY
ARI7961I	ARIIPQY
ARI7962E	ARIISQL
ARI7963E	ARIIPQY
ARI7964I	ARIIPQY
ARI7965I	ARIICICD
ARI801I	ARIDBS
ARI802I	ARIDCSP
ARI803E	ARIDBS
ARI804E	ARIDALC
	ARIDALI
	ARIDALT
	ARIDCFI
	ARIDCSP
	ARIDDFI
	ARIDDLO
	ARIDDUL
	ARIDEXI
	ARIDSFA
	ARIDSQL
	ARIDUNL
ARI805E	ARIDCSP
ARI806E	ARIDCSP
ARI807E	ARIDBS
ARI808I	ARIDBS
ARI809I	ARIDBS
ARI810E	ARIDCSP
ARI811I	ARIDBS
ARI812I	ARIDBS
ARI813I	ARIDBS
ARI814E	ARIDCFI
	ARIDDFI
	ARIDDUL
	ARIDMGE
ARI815I	ARIDBS
ARI816E	ARIDCSP
ARI817I	ARIDBS
ARI818E	ARIDEXI
ARI819I	ARIDBS
ARI820E	ARIDMGE
ARI821E	ARIDEXI
ARI822E	ARIDDLO
	ARIDDUL
ARI823E	ARIDSQL
ARI824E	ARIDALC
	ARIDALI
	ARIDALT
	ARIDDLO

Message ID	Issuing Module(s)
	ARIDDUL
	ARIDREL
	ARIDSEL
	ARIDSFA
	ARIDSQL
	ARIDUNL
ARI825E	ARIDALC
	ARIDALI
	ARIDALT
	ARIDBS
	ARIDDLO
	ARIDDUL
	ARIDREL
	ARIDSEL
	ARIDSFA
	ARIDSQL
	ARIDUNL
ARI826E	ARIDUSQ
ARI827I	ARIDBS
ARI828I	ARIDBS
ARI829E	ARIDMGE
ARI830E	ARIDDUL
ARI831I	ARIDDUL
ARI832I	ARIDDUL
ARI833I	ARIDDUL
ARI834E	ARIDDUL
ARI835I	ARIDDUL
ARI836I	ARIDDUL
ARI837I	ARIDDUL
ARI838E	ARIDCSP
	ARIDUSQ
ARI839E	ARIDBS
ARI840E	ARIDSFA
ARI841E	ARIDDLO
ARI845I	ARIDREL
ARI846I	ARIDREL
ARI847I	ARIDREL
ARI848E	ARIDREL
ARI849E	ARIDREL
ARI850I	ARIDSEL
ARI851E	ARIDSEL
ARI852I	ARIDDLO
	ARIDDUL
	ARIDREL
	ARIDUNL
ARI853I	ARIDREL
	ARIDUNL
ARI854E	ARIDREL
	ARIDUNL
ARI855I	ARIDDLO
	ARIDDUL

Message ID	Issuing Module(s)
ARI856E	ARIDSQL
ARI857E	ARIDDLO
	ARIDDUL
ARI858E	ARIDDLO
ARI859E	ARIDDLO
	ARIDDUL
ARI860E	ARIDDLO
ARI861E	ARIDDLO
	ARIDDUL
ARI862E	ARIDDLO
ARI863E	ARIDDLO
ARI864E	ARIDDLO
ARI865E	ARIDDLO
	ARIDDUL
ARI866E	ARIDDLO
ARI867E	ARIDUCP
	ARIDDUL
ARI868I	ARIDDFI
ARI869I	ARIDUCP
ARI870I	ARIDBS
ARI871E	ARIDREL
	ARIDUNL
ARI872E	ARIDREL
	ARIDUNL
ARI873E	ARIDUNL
ARI874I	ARIDUNL
ARI875I	ARIDDLO
	ARIDREL
ARI876E	ARIDREL

Message ID	Issuing Module(s)
ARI877E	ARIDREL
ARI878E	ARIDREL
ARI879I	ARIDREL
ARI880I	ARIDREL
ARI881E	ARIDREL
ARI882E	ARIDUCP
ARI883E	ARIDUCP
	ARIDUSQ
ARI884E	ARIDUCP
ARI885E	ARIDUCP
ARI886E	ARIDUCP
ARI887E	ARIDUCP
ARI888E	ARIDDLO
	ARIDEXI
	ARIDUCP
ARI889E	ARIDUCP
ARI890E	ARIDUCP
ARI891E	ARIDUCP
ARI892E	ARIDUCP
ARI893E	ARIDUCP
ARI894E	ARIDUCP
ARI895E	ARIDUCP
ARI896E	ARIDUCP
ARI897E	ARIDUCP
ARI898I	ARIDBS
ARI899I	ARIDBS
ARI900E	ARISDBR
	ARISDSK
	ARISEGA

Message ID	Issuing Module(s)
ARI901E	ARISDBR
	ARISPFM
ARI903E	ARISDBR
ARI904E	ARISDBR
ARI905E	ARISDBR
ARI906E	ARISFDB
	ARISIIO
ARI907E	ARISEGA
	ARISPFM
ARI908E	ARISPFM
ARI909E	ARISDSK
	ARISPFM
ARI910E	ARISDSK
	ARISEGA
	ARISPFM
ARI911E	ARISEGA
	ARISPFM
ARI912E	ARISEGA
	ARISFM1
	ARISPFM
ARI913E	ARISFM1
ARI915I	ARISEGA
ARI916I	ARISEGA
ARI917E	ARISDSK
ARI918I	ARISDSK
ARI919D	ARISPFM
ARI920D	ARIYL14
ARI921I	ARISEGA

Message ID	Issuing Module(s)
ARI922I	ARISDSK
ARI923E	ARISFM1(C
ARI925E	ARISFM1(C
	ARIYI07(C
ARI926E	ARIYI22(C
ARI927E	ARISFM1(C
	ARIYI22(C
	ARIYI23(C
ARI928E	ARISFM1(C
	ARIYI47(C
ARI950E	ARIMTRA
ARI951E	ARIMTRA
ARI952E	ARIMTRA
ARI953E	ARIMTRA
ARI954E	ARIMTRA
ARI955E	ARIMTRA
ARI956E	ARIMTRA
ARI957E	ARIMTRA
ARI958E	ARIMTRA
ARI959E	ARIMTRA
ARI960E	ARIMTRA
ARI961E	ARIMTRA
ARI962E	ARIMTRA
ARI963E	ARIMTRA
ARI964E	ARIMTRA
ARI965E	ARIMTRA
ARI966E	ARIMTRA
ARI967E	ARIMTRA

Message ID	Issuing Module(s)
ARI968A	ARIMTRA
ARI969I	ARIMTRA
ARI972E	ARIMTRA
ARI974E	ARIMTRA
ARI980A	ARI9MSGC
	SQLTRFMT (CMS
	EXECs)
ARI981D	ARI9MSGC
	SQLTRFMT (CMS
	EXECs)
ARI982I	ARI9MSGC
	SQLTRFMT (CMS
	EXECs)
ARI983D	ARI9MSGC
	SQLTRFMT (CMS
	EXECs)
ARI984E	ARI9MSGC
	SQLTRFMT (CMS
	EXECs)
ARI985E	ARI9MSGC
	SQLTRFMT (CMS
	EXECs)
ARI986E	ARI9MSGC
	SQLTRFMT (CMS
	EXECs)
ARI987E	ARI9MSGC
	SQLTRFMT (CMS
	EXECs)

MODULES IN SEQUENCE - MESSAGES ISSUED

Issuing Module	Message ID
ARICABE	ARI0035I ARI0036E ARI0037E ARI0038E ARI0040E ARI044D
ARICCLA	ARI0027E ARI0029I
ARICCRA	ARI0001E
ARICEMGC	(EXEC) ARI640E ARI641E ARI642E ARI643E
ARICENA	ARI0027E
ARICIP1	ARI0025I ARI0026E ARI0027E ARI0040E ARI0042I ARI0043I ARI0047E
ARICIP2	ARI0027E ARI0047E
ARICMUD	ARI0001E ARI0027E
ARICOMB	ARI0027E
ARICPRM	ARI0002E ARI0003E ARI0004I ARI0005E ARI0006E ARI0007E ARI0008E ARI0009E ARI0010E ARI0011E ARI0012I ARI0013E ARI0014E ARI0015I ARI0016I ARI0017E ARI0018I ARI0019E
ARICSHT	ARI0027E ARI0028I ARI0029I ARI0030I ARI0031E
ARICSPM	ARI0025I ARI0026E ARI0046I ARI0047E
ARICSTK	ARI0001E
ARICTRC	ARI082E ARI083I ARI084D ARI085E ARI086E ARI087D ARI088D ARI089E ARI090D ARI091D ARI093I ARI094E ARI095I ARI096I ARI097I ARI098E ARI099I
ARICTRI	ARI055A ARI056I ARI086E ARI094A ARI085A
ARICTRM	ARI032I ARI033E ARI034E ARI042I ARI043I
ARIDALC	ARI804E ARI824E ARI825E
ARIDALI	ARI804E ARI824E ARI825E
ARIDALT	ARI804E ARI824E ARI825E
ARIDBS	ARI801I ARI803E ARI807E ARI808I ARI809I ARI811I ARI812I ARI813I ARI815I ARI817I ARI819I ARI825E ARI827I ARI828I ARI839E ARI870I ARI898I ARI899I
ARIDCFI	ARI804E ARI814E
ARIDCSP	ARI802I ARI804E ARI805E ARI806E ARI810E ARI816E ARI838E
ARIDDFI	ARI804E ARI814E ARI868I
ARIDDLO	ARI804E ARI822E ARI824E ARI825E ARI841E ARI852I ARI855I ARI857E ARI858E ARI859E ARI860E ARI861E ARI862E ARI863E ARI864E ARI865E ARI866E ARI875I ARI888E
ARIDDUL	ARI804E ARI814E ARI822E ARI824E ARI825E ARI830E ARI831I ARI832I ARI833I ARI834E ARI835I ARI836I ARI837I ARI852I ARI855I ARI857E ARI859E ARI861E ARI865E ARI867E
ARIDEXI	ARI804E ARI818E ARI821E ARI888E
ARIDMGE	ARI814E ARI820E
ARIDREL	ARI824E ARI825E ARI829E ARI845I ARI846I ARI847I ARI848E ARI849E ARI852I ARI853I ARI854E ARI871E ARI872E ARI875I ARI876E ARI877E ARI878E ARI879I ARI880I ARI881E
ARIDSEL	ARI824E ARI825E ARI850I ARI851E
ARIDSFA	ARI804E ARI824E ARI825E ARI840E
ARIDSQL	ARI804E ARI823E ARI824E ARI825E ARI856E
ARIDSQLA	ARI500I ARI501I ARI502I ARI503E ARI504I ARI505I ARI511E
ARIDUCP	ARI867E ARI869I ARI882E ARI883E ARI884E ARI885E ARI886E ARI887E ARI888E ARI889E ARI890E ARI891E ARI892E ARI893E ARI894E ARI895E ARI896E ARI897E
ARIDUNL	ARI804E ARI824E ARI825E ARI852I ARI853I ARI854E ARI871E ARI872E ARI873E ARI874I
ARIDUSQ	ARI826E ARI838E ARI883E
ARIIBIN	ARI503E ARI504I ARI505I ARI7300E ARI7301E ARI7302E ARI7303E ARI7304E ARI7305E ARI7306E ARI7307A ARI7308A ARI7309E ARI7310E ARI7311E ARI7312I ARI7313I ARI7314I ARI7315E ARI7316E ARI7317I ARI7318E ARI7319I ARI7320I ARI7321E ARI7322I ARI7323I ARI7324E ARI7325I ARI7326E ARI7327E ARI7328E ARI7329E ARI7330E
ARIICAN	ARI503E ARI504I ARI505I ARI7042E ARI7043I ARI7045E ARI7051E
ARIICICD	ARI7000E ARI7001E ARI7041D ARI7044I ARI7046D ARI7047I ARI7048I ARI7300E ARI7301E ARI7965I
ARIICI2D	ARI503E ARI504I ARI505I ARI7081A ARI7082A ARI7083A ARI7084I ARI7085E ARI7086E ARI7087E
ARIICMD	ARI503E ARI504I ARI505I ARI7601I ARI7602D ARI7603I ARI7604E ARI7622E ARI7623E ARI7632E ARI7633E ARI7634E ARI7635I ARI7636D ARI7637E ARI7639D ARI7645E ARI7651I ARI7652I
ARIIDBS	ARI7040I ARI7080A
ARIIDQY	ARI7939E ARI7942E ARI7949I ARI7951I ARI7952E ARI7953E ARI7957E
ARIIERS	ARI503E ARI504I ARI505I ARI7550D ARI7551E ARI7552I ARI7558E ARI7559I ARI7564E ARI7569E ARI7570I ARI7585I ARI7590I ARI7591I ARI7592I
ARIIEXT	ARI7600E ARI7650E
ARIIFCC	ARI7502I ARI7504E ARI7505E ARI7506I ARI7508E ARI7511I ARI7513E ARI7515E ARI7516I ARI7517E ARI7518E ARI7519E ARI7520E ARI7523E ARI7526E ARI7546E ARI7547E
ARIIFCI	ARI7508E ARI7513E ARI7516I ARI7521I ARI7522I ARI7523E ARI7530E ARI7534E ARI7535E ARI7536E ARI7537E
ARIIFCS	ARI7508E ARI7513E ARI7516I ARI7521I ARI7522I ARI7523E ARI7530E ARI7533I ARI7534E ARI7535E ARI7536E ARI7537E

Issuing Module	Message ID
ARIIFMC	ARI7500E ARI7501E ARI7502I
	ARI7503I ARI7504E ARI7505E
	ARI7507E ARI7508E ARI7509I
	ARI7510I ARI7511I ARI7512I
	ARI7524I ARI7525I ARI7526E
	ARI7527I ARI7528I ARI7529I
	ARI7531I ARI7539I ARI7545E
	ARI7546E
ARIIFOL	ARI7127I
ARIIFOR	ARI058E
ARIIHLD	ARI7069I ARI7574E ARI7575E
ARIIHLP	ARI503E ARI504I ARI505I
	ARI7550D ARI7559I ARI7616I
	ARI7617E ARI7624E ARI7625E
	ARI7626E ARI7643E ARI7644E
	ARI7646E ARI7700I
ARIIIGN	ARI7607I ARI7700I
ARIILST	ARI503E ARI504I ARI505I
	ARI7550D ARI7558E ARI7559I
	ARI7590I ARI7591I ARI7592I
	ARI7616I ARI7617E ARI7620I
	ARI7621I ARI7626E ARI7701E
	ARI7725E
ARIIMSG	ARI057E ARI7022E
ARIINITC	ARI7001E ARI7085E
ARIINSQ	ARI7640E
ARIIPQY	ARI503E ARI504I ARI505I
	ARI7543I ARI7700I ARI7940E
	ARI7942E ARI7943E ARI7944E
	ARI7954E ARI7955I ARI7956E
	ARI7960I ARI7961I ARI7963E
	ARI7964I
ARIIPRF	ARI7398I ARI7399I
ARIIPRTC	ARI7885E ARI7886E ARI7887E
	ARI7888E ARI7889E ARI7890E
	ARI7891E ARI7892E ARI7894I
	ARI7895E ARI7896E ARI7897E
	ARI7898E ARI7901E ARI7902I
	ARI7903E ARI7904I ARI7905I
	ARI7906I ARI7908I ARI7920E
	ARI7921E ARI7922I ARI7923D
	ARI7924E
ARIIPRTD	ARI7865D ARI7866I ARI7867I
	ARI7868I ARI7869E ARI7870E
	ARI7871E ARI7872E ARI7873E
	ARI7874E ARI7875E ARI7876E
	ARI7877E ARI7878I ARI7879E
	ARI7882I ARI7883I ARI7884I
	ARI7893I ARI7901E ARI7902I
	ARI7903E ARI7904I ARI7905I
	ARI7906I ARI7907I ARI7908I
	ARI7910E ARI7911E ARI7913E
	ARI7915E ARI7916E ARI7917E
	ARI7918E ARI7920E ARI7921E
	ARI7922I ARI7923D ARI7924E
ARIIPSQ	ARI500I ARI501I ARI502I
	ARI503E ARI504I ARI505I
	ARI7544I ARI7932I ARI7933I
	ARI7934I ARI7936E ARI7937E
	ARI7945I ARI7958I ARI7959I
ARIIQRY	ARI7616I ARI7702I ARI7709I
	ARI7710I ARI7711I ARI7712I
	ARI7713I ARI7714I ARI7715I
	ARI7725E ARI7752I ARI7753I
	ARI7755I ARI7759I
ARIIREAC	ARI7102I
ARIIREC	ARI503E ARI504I ARI505I
	ARI7550D ARI7551E ARI7553E
	ARI7554E ARI7555E ARI7557I
	ARI7558E ARI7559I ARI7560E
	ARI7569E
ARIIRNM	ARI503E ARI504I ARI505I
	ARI7550D ARI7551E ARI7552I
	ARI7553E ARI7557I ARI7558E
	ARI7559I ARI7560E ARI7561E
	ARI7563E ARI7564E ARI7568E
	ARI7570I ARI7572E
ARIIRPT	ARI7940E
ARIIRUN	ARI503E ARI504I ARI505I
	ARI7350E ARI7351E ARI7354E
	ARI7355E ARI7357E ARI7358E
	ARI7359E ARI7360E ARI7361I
	ARI7362E ARI7363D ARI7364E
	ARI7619E ARI7700I
ARIIRWI	ARI057E ARI7051E ARI7060I
	ARI7061I ARI7062E ARI7063I
	ARI7064I ARI7065I ARI7066I
	ARI7068I ARI7070I ARI7071I
	ARI7607I
ARIISCC	ARI503E ARI504I ARI505I
	ARI7558E ARI7611E ARI7612E
	ARI7613I ARI7614E ARI7615E
	ARI7616I ARI7617E ARI7618E
	ARI7619E ARI7626E ARI7627E
	ARI7628I ARI7629E ARI7630E
	ARI7631E ARI7638I ARI7641D
	ARI7642I ARI7700I
ARIISET	ARI7701E ARI7711I ARI7725E
	ARI7726E ARI7727I ARI7733E
	ARI7734E ARI7735I ARI7740E
	ARI7750E ARI7751E ARI7753I
	ARI7754E ARI7755I ARI7760E
	ARI7761I
ARIISMG	ARI7020E ARI7021E
ARIISQL	ARI7049E ARI7946E ARI7947E
	ARI7962E
ARIISTR	ARI503E ARI504I ARI505I
	ARI7550D ARI7551E ARI7552I
	ARI7553E ARI7556E ARI7557I
	ARI7558E ARI7559I ARI7564E
	ARI7568E ARI7569E ARI7572E
	ARI7576D ARI7577D ARI7578I
	ARI7579I ARI7580I ARI7581I
	ARI7582I ARI7583E ARI7593I
ARIIST2	ARI503E ARI504I ARI505I
	ARI7602D ARI7604E ARI7616I
	ARI7701E ARI7725E ARI7726E
	ARI7727I ARI7731I ARI7732I
	ARI7733E ARI7734E ARI7736E
	ARI7737I
ARIIST3	ARI7616I ARI7725E ARI7727I
	ARI7729I ARI7733E ARI7735I
	ARI7739E ARI7740E ARI7742E
	ARI7743I ARI7744I ARI7745I
	ARI7746I ARI7757I ARI7758I
ARIISUB	ARI7120E ARI7121E ARI7123E
	ARI7124E ARI7125E ARI7126E
	ARI7128E
ARIISYSC	ARI7119E
ARIITIOC	ARI7101E
ARIITRC	ARI7180E ARI7181I ARI7182E
	ARI7184I ARI7185I ARI7186I
ARIIVFYC	ARI7041A ARI7047I ARI7048I
	ARI7050I
ARIIVFYD	ARI7041A ARI7047I ARI7048I
ARIMFMT	ARI058E
ARIMTRA	ARI950E ARI951E ARI952E
	ARI953E ARI954E ARI955E
	ARI956E ARI957E ARI958E
	ARI959E ARI960E ARI961E
	ARI962E ARI963E ARI964E
	ARI965E ARI966E ARI967E
	ARI968E ARI969I ARI972E
	ARI974E
ARIPDCF	ARI557E ARI567E ARI574E
ARIPLTP	ARI503E ARI504I ARI505I
ARIPMSF	ARI589E
ARIPMSH	ARI585E ARI589E
ARIPPIFA	ARI562I ARI570I ARI571I
	ARI579I ARI587I
ARIPPRA	ARI503E ARI504I ARI505I
	ARI547I ARI560I ARI561I

Issuing Module	Message ID
	ARI570I ARI571I ARI572I
	ARI574E ARI575E ARI579I
	ARI580E ARI582E ARI584E
	ARI585E ARI586I ARI587I
	ARI590E ARI591E ARI592E
	ARI593E ARI597E ARI598I
ARIPPRC	ARI503E ARI504I ARI505I
	ARI547I ARI560I ARI561I
	ARI570I ARI571I ARI572I
	ARI575E ARI577E ARI579I
	ARI580E ARI582E ARI584E
	ARI585E ARI586I ARI587I
	ARI598I ARI599E
ARIPPRF	ARI536I ARI537E ARI540E
	ARI542E ARI546E ARI547E
	ARI560E ARI561E ARI563E
	ARI564E ARI565E ARI577E
	ARI578E ARI580E ARI584E
	ARI585E ARI586E ARI598E
ARIPPRP	ARI503E ARI504I ARI505I
	ARI547I ARI560I ARI561I
	ARI570I ARI571I ARI572I
	ARI574E ARI575E ARI579I
	ARI580E ARI582E ARI584E
	ARI585E ARI586I ARI587I
	ARI590E ARI591E ARI597E
	ARI598I
ARIPSCN	ARI539I ARI551I ARI552I
	ARI580E ARI581E ARI583E
	ARI588E
ARIPSQA	ARI503E ARI504I ARI505I
	ARI537E ARI540E ARI541E
	ARI542E
ARIPSQC	ARI503E ARI504I ARI505I
	ARI537E ARI540E ARI541E
	ARI542E ARI576E ARI594E
	ARI595E ARI596E
ARIPSQF	ARI537E ARI541E ARI558E
	ARI559E ARI568E ARI569E
	ARI574E ARI581E ARI585E
	ARI590E ARI591E ARI597E
ARIPSQP	ARI503E ARI504I ARI505I
	ARI537E ARI540E ARI541E
	ARI542E
ARIPSCF	ARI539I ARI551I ARI552I
	ARI580I ARI581E ARI583E
	ARI588E
ARIPSSF	ARI545E ARI548E ARI549E
	ARI550E ARI553E ARI554E
	ARI555E ARI556E ARI559E

Issuing Module	Message ID
	ARI566E ARI569E ARI572E
	ARI575E
ARIPTXA	ARI585E
ARIPTXF	ARI585E
ARIPTXC	ARI503E ARI504I ARI505I
	ARI585E
ARIPTXP	ARI585E
ARIRBRM	ARI415D ARI416E ARI417I
ARIRDIS	ARI400E ARI403E ARI404E
	ARI406E ARI407E ARI411I
	ARI412I ARI413I ARI414I
	ARI421I ARI425I ARI426I
	ARI427A ARI430I ARI431E
	ARI432E ARI433E ARI434E
	ARI436E ARI438E
ARIRENA	ARI400E ARI401I ARI402E
	ARI403E ARI404E ARI405E
	ARI406E ARI407E ARI408E
	ARI410I ARI418I ARI419I
	ARI420E ARI423A ARI424I
	ARI437E
ARIRORMD	ARI422E ARI424I ARI436E
	ARI438E ARI450I
ARIRVRM	ARI429E
ARISAVES	ARI613E ARI627E ARI628E
(EXEC)	ARI629E ARI651E ARI652E
	ARI653E ARI661E ARI673I
ARISCNV	ARI525A
ARISCOL	ARI524A
ARISDBMA	ARI613E ARI627E ARI628E
(EXEC)	ARI629E ARI651E ARI653E
	ARI661I ARI664E ARI666I
ARISDBR	ARI001E ARI900E ARI901E
	ARI903E ARI904E ARI905E
ARISDFL	ARI522A ARI523A ARI524A
ARISDFM	ARI522A ARI523A ARI524A
ARISDFR	ARI521A ARI522A ARI523A
ARISDSK	ARI900E ARI909E ARI910E
	ARI917E ARI918I ARI922I
ARISEDI	ARI522A
ARISEGA	ARI900E ARI907E ARI910E
	ARI911E ARI912E ARI915E
	ARI916E ARI921I
ARISEGB	ARI523A
ARISEMSG	ARI657I ARI658E ARI662I
	(EXEC)
ARISERR	ARI057E
ARISESCP	ARI658E ARI659I ARI660I
	(EXEC)
ARISFDB	ARI001E ARI906E
ARISFIL	ARI522A ARI523A ARI524A

Issuing Module	Message ID
ARISFM1(C	ARI001E ARI912E ARI913E
	ARI923E ARI925E ARI927E
	ARI928E
ARISFM1(D	ARI001E ARI913E
ARISIIO	ARI001E ARI906E
ARISINI	ARI522A ARI523A
ARISIST	ARI520A ARI521A ARI522A
	ARI523A ARI524A
ARISIST2	ARI522A ARI523A ARI524A
ARISLKIT	ARI652E ARI653E ARI658E
(EXEC)	ARI670E ARI671E ARI672E
ARISMAI	ARI520A ARI521A ARI522A
	ARI523A ARI526A ARI527A
	ARI528A ARI529A ARI530A
ARISPDFC	ARI607E ARI639E ARI650E
(EXEC)	ARI652E
ARISPFM	ARI200E ARI901E ARI907E
	ARI908E ARI909E ARI910E
	ARI911E ARI912E ARI919D
ARISPROD	ARI631E ARI652E ARI653E
(EXEC)	ARI658E
ARISQLLD	ARI652E ARI653E ARI658E
(EXEC)	
ARISSERV	ARI631E ARI652E ARI653E
ARISSET	ARI522A
ARISSLKE	ARI652E ARI653E ARI658E
	ARI667E ARI668E ARI669I
ARISUPD	ARI521A ARI522A ARI523A
	ARI524A
ARISYSD (EP ARISYSDG)	ARI042I ARI043I
	ARI059E
ARISYSD(D (EP ARISYSD5)	ARI068E
ARISYSE(C (EP ARISYSD5)	ARI069E
	ARI070A ARI072D
ARISYSIN (EXEC)	ARI072D
ARIXERO	ARI053E ARI054E
ARIXETR	ARI081A
ARIYC11	ARI001E
ARIYD69	ARI001E
ARIYD81	ARI001E
ARIYE14	ARI081A
ARIYI07(C	ARI925E
ARIYI14	ARI203E
ARIYI19	ARI203E
ARIYI20	ARI201E
ARIYI22(C	ARI001E ARI926E ARI927E
ARIYI22(D	ARI001E
ARIYI23(C	ARI927E
ARIYI29	ARI201E
ARIYI36	ARI001E
ARIYI47(C	ARI928E ARI202I

Issuing Module	Message ID
ARIYI51	ARI202I
ARIYK12	ARI001E
ARIYL00	ARI285I
ARIYL01	ARI276E
ARIYL08	ARI001E
ARIYL13	ARI001E ARI280A ARI281I ARI282I ARI283I
ARIYL14	ARI200E ARI920D
ARIYL15	ARI284I
ARIYL21	ARI001E ARI286I ARI289I ARI294A ARI295A ARI296I ARI297A
ARIYL22	ARI290I ARI291I ARI294A ARI298E
ARIYL23	ARI001E ARI292I ARI293I ARI294A ARI296I ARI297A ARI298E ARI299A
ARIYL24	ARI287I ARI288I
ARIYM01	ARI045I ARI060I ARI095I
ARIYM02	ARI039E ARI040E ARI041E
ARIYM04	ARI057E
ARIYM05	ARI058E
ARIYM10	ARI062A ARI061E ARI063I ARI064E ARI065I ARI066E
ARIYM11	ARI067E
ARIYS08	ARI001E
ARIYT00	ARI204D
ARIYT01	ARI067E
ARIYT16	ARI230I
ARIYT18	ARI220E ARI221E ARI222E ARI223E ARI224E ARI225E ARI226I ARI227E ARI228E ARI229E
ARIYT29	ARI231E ARI232E

Issuing Module	Message ID
ARIYX04	ARI001E
ARIYZ00	ARI001E
ARIYZ19	ARI248E ARI249E
AXTCENT	ARI7600E
ARI9MSGC (EXEC)	ARI980A ARI981D ARI982I ARI983D ARI984E ARI985E ARI986E ARI987E
SQLADBEX (EXEC)	ARI043I ARI601E ARI607E ARI613E ARI614E ARI616E ARI621E ARI627E ARI628E ARI629E ARI631E ARI638D ARI639E ARI644E ARI645E ARI647D ARI648A
SQLADBSP (EXEC)	ARI043I ARI601E ARI613E ARI616E ARI627E ARI628E ARI629E ARI638D ARI639E ARI644E ARI645E ARI648A ARI649A
SQLDBGEN (EXEC)	ARI043I ARI601E ARI607E ARI613E ARI616E ARI621E ARI627E ARI628E ARI629E ARI631E ARI632D ARI633D ARI635D ARI636D ARI637D ARI638D ARI639E ARI644E ARI647D ARI651E
SQLDBID (EXEC)	ARI603I ARI652E
SQLDBINS (EXEC)	ARI606E ARI607E ARI613E ARI620I ARI626E ARI627E ARI628E ARI629E ARI631E ARI651E ARI652E ARI653E ARI661E ARI664E
SQLGENLD	ARI043I ARI600A ARI602D

Issuing Module	Message ID
(EXEC)	ARI604A ARI605E ARI606E ARI607E ARI608E ARI616E ARI620I ARI624A ARI625D ARI626E ARI651E
SQLINIT (EXEC)	ARI043I ARI600A ARI601E ARI606E ARI615D ARI618I ARI619E ARI626E ARI627E ARI628E ARI629E ARI630E ARI639E ARI650E ARI653E
SQLISTRT (EXEC)	ARI692E ARI693E ARI694I ARI697E ARI698E ARI699E
SQLLOG (EXEC)	ARI043I ARI601E ARI605D ARI607E ARI613E ARI616E ARI620I ARI621E ARI627E ARI628E ARI629E ARI631E ARI639E ARI644E ARI645E ARI647D ARI651E ARI682E ARI683E ARI684I ARI685I ARI686E ARI687E
SQLPREP (EXEC)	ARI612E ARI613E ARI628E ARI629E ARI663I ARI675E ARI676E ARI677I ARI678E ARI679E ARI680E ARI681E
SQLSTART (EXEC)	ARI043I ARI601E ARI607E ARI609E ARI611E ARI612E ARI613E ARI616E ARI617E ARI622E ARI627E ARI628E ARI629E ARI630E ARI631E ARI639E ARI653E ARI663I
SQLTRFMT (EXEC)	ARI980A ARI981D ARI982I ARI983D ARI984E ARI985E ARI986E ARI987E

MODULE-TO-MODULE CROSS REFERENCE

In the center of the page (contained within vertical bars) is a list of SQL/DS modules. To the left of these are the modules that call the center module, and to the right are the modules called by the center module.

Names with an asterisk (*) identify secondary entry points. For these, a separate list is included following the cross-reference list. This list gives each secondary entry point name and the module(s) to which it belongs.

Calling Modules			Module/ Entry Pt	Called Modules/Entry Points		
		ARIYM02	ARICABE	ARICDMP	ARICDSP1*	ARICPDM
Also from operating system via STXIT AB for VSE or via ABNEXIT SET for VM. Set by ARICIP1.				ARICTRM	ARISYSDG*	ARIXEBR1 (via RDCBRCL)
				ARIYE13	ARIYM04	ARIYT15
				ARIYT23		
		ARICDSP	ARICCLA	ARICCOM	ARICENA	ARICRST
				ARIYK19	ARIYM02	ARIYM04
ARICCLA	ARICDSP	ARICENA	ARICCOM			
ARICINT(C	ARICIP1(C	ARICIP2				
ARICMUD	ARICOMB	ARICSHT				
ARIRBRM	ARIRDIS(D	ARIRENA(D				
ARIRIRM	ARIRORM(D	ARIRSEN(D				
ARIRVIR(C	ARIRVRM(C	ARISFM1(C				
ARIYI22(C	ARIYI23(C	ARIYI47(C				
ARICIP1	ARICIP2	ARIYL00	ARICCRA	ARICSTK	ARICWSG	ARICWSI
		ARIYL13		ARIYM00	ARIYM02	
		ARICABE	ARICDMP	ARISYSDG*	ARISYSD2*	ARIYE13
		ARIYK19	ARICDPT			
		ARICIP1	ARICDSP	ARICCLA	ARICCOM	ARICENA
				ARICRST	ARIYM02	ARIYT23
ARICDWT	ARICWAT (via DS2WATP)		ARICDSP1*			
ARIXETR	ARIXERD	ARIYK18	ARICDWT	ARICDSP1*		
(via DS2DWTP)						
ARISAVES[0]	ARISDEMA[0]	ARISEMSG[0]	ARICEMGC[0]			
ARISESCP[0]	ARISLKIT[0]	ARISPDFC[0]				
ARISPROD[0]	ARISQLLD[0]	ARISSERV[0]				
ARISSLKE[0]	SQLADBEX[0]	SQLADBSP[0]				
SQLDBGEN[0]	SQLDBID[0]	SQLDBINS[0]				
SQLDBSU[0]	SQLGENLD[0]	SQLINIT[0]				
SQLISTRT[0]	SQLLOG[0]	SQLPREP[0]				
	SQLSTART[0]	SQLTRFMT[0]				
ARICCLA	ARICDSP	ARICIP1	ARICENA	ARICCOM	ARIYM02	ARIYM04
ARICIP2	ARIYL00	ARIYL13				
ARICINT(C	ARICIP1	ARICRST	ARICFSE			
ARIXCR1	ARIXEBR	ARIXEDR				
(Module ARICFSE may be invoked by most of the DBSS/DSC/RDS modules (via RSSOPTS) from the prolog and epilog processing.)						
ARICINT(C	ARIXCRE	ARIXEBR	ARICGSE	ARISYSDG*		
		ARIXEDR				
(Module ARICGSE may be invoked by most of the						

* Asterisked items are secondary entry points. See page 346. [0] indicates CMS EXEC

Calling Modules			Module/ Entry Pt	Called Modules/Entry Points		
DBSS/DSC/RDS modules (via RSSOPTS) from the prolog and epilog processing.)						
ARIRBRM	ARIRSEN(D	ARIRVRM(C	ARICIMB	ARISYSD1*	ARISYSD2*	
For VSE, called by operating system when EXEC ARISQLDS is done. For VM, branched to from the ARISQBBT bootstrap module which is invoked by the SQLSTART EXEC. ARICTRM returns to ARICIP1 upon SQL/DS termination to clean up storage and returns to VSE or VM.			ARICINT	ARICCOM	ARICFSE	ARICGSE
			ARICIP1	ARICCOM (VM)	ARICCRA	ARICDSP
				ARICENA	ARICFSE	ARICIP2 (via ARICDSP)
				ARICPRM	ARICRST	ARICWFR
				ARICWSG	ARICWSI	ARISYSDA*
				ARISYSDC* (VSE)	ARISYSDE* (VM)	ARISYSDG*
				ARISYSDK*	ARISYSD2*	ARISYSD3*
				ARISYSD6* (VSE)	ARISYSD9* (VM)	ARIYM02
				ARIYM04		
Called by ARICIP1 via ARICDSP			ARICIP2	ARICCOM	ARICCRA	ARICENA
				ARICSPM (via ARICDSP)		ARICTRM
				ARIYM01	ARIYM02	ARIYM04
				ARIYT00		
		SQLDBID⁰	ARICLOC (ARICLOC is a system product editor macro)			
ARIRBRM	ARIRSEN(D	ARIRVRM(C	ARICMMB			
	ARIXELK	ARIXERD	ARICMUD	ARICCOM	ARICWAT	ARICWSF
				ARICWSG	ARIYM02 (via DS2SYSER)	
				ARIYM04		
ARIXCR2	ARIXEDC	ARIXEDS	ARICOMB	ARICCOM	ARICWAT	ARIYM02 (via ARIYM04)
ARIXELK (via DS2OMBP)		ARIXERD				
ARICABE	ARIXETR (via DS2PDMP)		ARICPDM	ARISYSDA*		
		ARIYE14				
		ARICIP1	ARICPRM	ARICTKN	ARISYSDG*	ARISYSD8*
				ARIYM04		
ARICCLA	ARICDSP	ARICIP1	ARICRST	ARICFSE	ARICWSF	ARIYT06
		ARIYM11	ARICSHT	ARICCOM	ARICTRM	ARIYK19
				ARIYM02	ARIYM04	
Called by ARICIP2 via ARICDSP			ARICSPM	ARICTRM	ARISYSD6*	ARIYM02
				ARIYM04		
		ARICCRA	ARICSTK	ARICWSG		
ARICPRM	ARIRDIS(D	ARIRENA	ARICTKN			
		ARIYM11	ARICTRC	ARISYSD5*	ARIYE04	ARIYE06
				ARIYE08		
		ARICTRM	ARICTRC1*			
		ARIYT00	ARICTRI	ARISYSD5*	ARIYE06	
ARICABE	ARICIP2	ARICSHT	ARICTRM	ARICTRC1*	ARISYSDH*	ARIYI47
ARICSPM	ARISFM1(C	ARIYI22(C		ARIYM04		
ARIYL08	ARIYL09	ARIYL13				
ARIYL14	ARIYL21	ARIYL22				
	ARIYM02	ARIYT00				
	ARICMUD	ARICOMB	ARICWAT	ARICDSP1*	ARIYM02	ARIYM04
				ARIYT15 (via DS2SYSBP)		
				ARIYT23 (via DS2STBKP)		

* Asterisked items are secondary entry points. See page 346. ⁰ indicates CMS EXEC

Calling Modules			Module/ Entry Pt	Called Modules/Entry Points		
	ARICIP1	ARIYT06	ARICWFR	ARISYSD2*		
ARICMUD	ARICRST	ARICWSG	ARICWSF			
ARICWSI	ARISFM1(D	ARISII0				
ARIXA06	ARIXA07	ARIXA10				
ARIXCR2	ARIXC14	ARIXC19				
ARIXC21	ARIXC37	ARIXEDP				
ARIXEDS	ARIXEFB	ARIXELX				
ARIXEPP	ARIXERD	ARIXERO				
ARIXESP	ARIXESX	ARIXI03				
ARIXI07	ARIXI15	ARIXI19				
ARIXI25	ARIXOAF	ARIXOCU				
ARIXOD1	ARIXOD2	ARIXOGP				
ARIXOML	ARIXOOP	ARIXOSR				
ARIXOVC	ARIXOVD	ARIXPA1				
ARIYC11	ARIYD69	ARIYD81				
ARIYI36	ARIYL08	ARIYL13				
ARIYS24	ARIYX04	ARIYZ00				
		ARIYZ15				
ARICCRA	ARICIP1	ARICMUD	ARICWSG	ARICWSF	ARISYSDI*	
ARICSTK	ARISDBR	ARISFDB				
ARISFM1	ARISII0	ARIXA01				
ARIXA06	ARIXA07	ARIXA08				
ARIXA10	ARIXCR2	ARIXCR3				
ARIXC13	ARIXC19	ARIXC21				
ARIXC37	ARIXEAB	ARIXEDP				
ARIXEDS	ARIXELX	ARIXEPP				
ARIXERD	ARIXERO	ARIXESP				
ARIXESX	ARIXI01	ARIXI03				
ARIXI04	ARIXI06	ARIXI07				
ARIXI08	ARIXI09	ARIXI13				
ARIXI15	ARIXI18	ARIXI19				
ARIXI23	ARIXI24	ARIXI25				
ARIXI27	ARIXI28	ARIXOAF				
ARIXOCK	ARIXOD1	ARIXOD2				
ARIXOGP	ARIXOML	ARIXOOP				
ARIXOSR	ARIXOSU	ARIXOVC				
ARIXOVD	ARIXPA1	ARIYC11				
ARIYD69	ARIYD81	ARIYI22				
ARIYI36	ARIYK12	ARIYL08				
ARIYL13	ARIYL21	ARIYL23				
ARIYS08	ARIYS24	ARIYX04				
		ARIYZ00				
	ARICCRA	ARICIP1	ARICWSI	ARICWSF	ARISYSD1*	
	ARIDDLO	ARIDDUL	ARIDALC	ARIDMGE	ARISYSDA*	ARISYSD1*
				ARISYSD2*		
		ARIDREL	ARIDALI	ARIDMGE	ARISYSDA*	ARISYSD1*
				ARISYSD2*		
ARIDDLO	ARIDDUL	ARIDREL	ARIDALT	ARIDMGE	ARISYSDA*	ARISYSD1*
		ARIDUNL		ARISYSD2*		

* Asterisked items are secondary entry points. See page 346. ⁰ indicates CMS EXEC

Calling Modules			Module/ Entry Pt	Called Modules/Entry Points		
		SQLDBSU[0]	ARIDBS	ARIDCFI	ARIDCSP	ARIDDLO
				ARIDDUL	ARIDMGE	ARIDREL
				ARIDSEL	ARIDSQL	ARIDUNL
				ARIDUSQ	ARISYSDA*	ARISYSD1*
				ARISYSD2*		
	ARIDBS	ARIDCSP	ARIDCFI	ARIDMGE		
	ARIDREL	ARIDUNL	ARIDCIB			
ARIDBS	ARIDDLO	ARIDDUL	ARIDCSP	ARIDCFI	ARIDMGE	ARIDUCP
				ARISYSDA*		
	ARIDREL	ARIDUNL	ARIDCTB			
ARIDDLO	ARIDDUL	ARIDREL	ARIDDFI	ARIDMGE	ARISYSDA*	ARISYSD5*
		ARIDUNL				
		ARIDBS	ARIDDLO	ARIDALC	ARIDALT	ARIDCSP
				ARIDDFI	ARIDEXI	ARIDMGE
				ARIDSFA	ARIDSQL	ARIDSSB
				ARIMDFP2*	ARISYSDA*	ARISYSD1*
				ARISYSD2*		
		ARIDBS	ARIDDUL	ARIDALC	ARIDALT	ARIDCSP
				ARIDDFI	ARIDMGE	ARIDSFA
				ARIDSQL	ARIMDFP1	ARISYSDG*
				ARISYSD1*	ARISYSD2*	ARISYSDA*
		ARIDDLO	ARIDEXI	ARIDMGE	ARISYSDA*	ARISYSD6*
ARIDALC	ARIDALI	ARIDALT	ARIDMGE	ARIMFMT	ARISYSD3*	ARISYSD5*
ARIDBS	ARIDCFI	ARIDCSP				
ARIDDFI	ARIDDLO	ARIDDUL				
ARIDEXI	ARIDREL	ARIDSEL				
ARIDSFA	ARIDSQLA	ARIDUCP				
	ARIDUNL	ARIDUSQ				
		ARIDBS	ARIDREL	ARIDALI	ARIDALT	ARIDCIB
				ARIDCTB	ARIDDFI	ARIDMGE
				ARIDSFA	ARIDSQL	ARIDSSB
				ARISYSD1*	ARISYSD2*	
		ARIDBS	ARIDSEL	ARIDMGE	ARIDSFA	ARIDSQL
				ARIMDFP1*	ARISYSD1*	ARISYSD2*
ARIDDLO	ARIDDUL	ARIDREL	ARIDSFA	ARIDMGE	ARIDSQL	ARISYSDA*
	ARIDSEL	ARIDUNL		ARISYSD1*	ARISYSD2*	
ARIDBS	ARIDDLO	ARIDREL	ARIDSQL			
ARIDSEL	ARIDSFA	ARIDUNL				
		ARIDUSQ				
			ARIDSQLA	ARIDMGE	ARIMSMF	ARIPRDI
				ARISYSDA*	ARISYSD1*	ARISYSD2*
ARIDDLO	ARIDREL	ARIDUNL	ARIDSSB			
		ARIDCSP	ARIDUCP	ARIDMGE		
		ARIDBS	ARIDUNL	ARIDALT	ARIDCIB	ARIDCTB
				ARIDDFI	ARIDMGE	ARIDSFA
				ARIDSQL	ARIDSSB	ARISYSDA*
				ARISYSD1*	ARISYSD2*	
		ARIDBS	ARIDUSQ	ARIDMGE	ARIDSQL	

* Asterisked items are secondary entry points. See page 346. [0] indicates CMS EXEC

Calling Modules			Module/ Entry Pt	Called Modules/Entry Points		
		(see ARIPEIFA)	ARIFOR	(see ARIPEIFA)		
		ARIICMD	ARIIBIN	ARIIFM*	ARIIGM*	ARIIIGN
				ARIYM04*	ARIINME	ARIINSQ
				ARIIRWI	ARIISMG	ARIISQL
				ARIITKN	ARIIVLD	
ARIICMD	ARIIDBS	ARIIRWI	ARIICAN	ARIISMG	ARIISQL	ARIITS*
				ARIYM04*		
			ARIICICD	ARIIFULD	ARIIRETD	ARIISCMD
				ARIITRMD	ARIIVFYD	
			ARIICI2D	ARIIDBS	ARIIPRF	ARIISMG
				ARIISQL	ARIISQL3*	ARIISQL7*
				ARIISTK	ARIYM04*	
		ARIIDBS	ARIICMD	ARIIBIN	ARIICAN	ARIIERS
				ARIIHLD	ARIIHLP	ARIIIGN
				ARIILST	ARIIMAP	ARIIM04*
				ARIIOCI	ARIIREC	ARIIRNM
				ARIIRUN	ARIIRWI	ARIISCC
				ARIISCC1*	ARIISET	ARIISMG
				ARIISQL	ARIISTR	ARIITKN
				ARIITRC	ARIIVLD	
	ARIIFMT	ARIIPQY	ARIICNV			
			ARIIDBS	ARIICAN	ARIICMD	ARIIHLD
				ARIIPQL	ARIIRWI	ARIISMG
				ARIISQL	ARIISTR1	ARIITKN
				ARIITS*	ARIIWAIT*	ARIYM04*
		ARIIPRTD	ARIIDEQ*			
		ARIITRM	ARIIDGS*			
		ARIITRM	ARIIDM*			
			ARIIDQY	ARIIFET	ARIIFMT	ARIIHDR
				ARIITKN	ARIYM04*	
		ARIITRM	ARIIDST*			
		ARIIPRTD	ARIIENQ*			
		ARIICMD	ARIIERS	ARIYM04*	ARIISMG	ARIISQL
				ARIISQL1*	ARIITKN	
		ARIICMD	ARIIEXT	ARIYM04*		
		ARIIFMC	ARIIFCC	ARIIOVD	ARIITKN	ARIYM04*
			ARIIFCI	ARIITKN	ARIYM04*	
		ARIIFMC	ARIIFCI1*			
		ARIIFMC	ARIIFCI2*			
			ARIIFCS	ARIITKN	ARIYM04*	
		ARIIFMC	ARIIFCS1*			
		ARIIFMC	ARIIFCS2*			
		ARIIFMC	ARIIFCS3*			
ARIIDQY	ARIIPRTC	ARIIPRTD	ARIIFET	ARIIRPT2*	ARIIRPT3*	ARIIRPT5*
				ARIISQL		
ARIIBIN	ARIIHLP	ARIILST	ARIIFM*			
ARIIOCI	ARIIPQY	ARIIRPT				
	ARIISCC	ARIITRC				
		ARIIPQY	ARIIFMC	ARIIFCC	ARIIFCI1*	ARIIFCI2*

* Asterisked items are secondary entry points. See page 346. ⁰ indicates CMS EXEC

Calling Modules			Module/ Entry Pt	Called Modules/Entry Points		
				ARIIFCS1*	ARIIFCS2*	ARIIFCS3*
				ARIIOVD	ARIIRWI	ARIITKN
				ARIYM04*		
ARIIDQY	ARIIPRTC	ARIIPRTD	ARIIFMT	ARIICNV	ARIMDFP1*	
		ARIIRWI	ARIIFOL	ARIITKN	ARIYM04*	
ARIIBIN	ARIIMSG	ARIIRWI	ARIIFOR	ARIMBIN	ARIMDEC	
ARIICICD	ARIIVFYD	ARIIWRTD	ARIIFULD	ARIIRETD	ARIISCMD	
ARIIBIN	ARIIHLP	ARIILST	ARIIGM*	ARIIMSG		
		ARIISTK				
ARIIOCI	ARIIPQY	ARIIRPT				
	ARIISCC	ARIITRC				
ARIIDQY	ARIIPRTC	ARIIPRTD	ARIIHDR			
		ARIICMD	ARIIHLD	ARIYM04*	ARIIPQL1*	
		ARIICMD	ARIIHLP	ARIIFM*	ARIIGM*	ARIYM04*
				ARIIPSQ	ARIISMG	ARIISQL
				ARIISQL4*	ARIISQL5*	ARIISQL6*
				ARIITKN		
ARIIBIN	ARIICMD	ARIIPQY	ARIIIGN	ARIIMSG	ARIITKN	
		ARIICMD	ARIILST	ARIIFM*	ARIIGM*	ARIYM04*
				ARIIQRY	ARIIRWI	ARIISMG
				ARIISQL	ARIISQL1*	ARIISQL2*
				ARIITKN		
		ARIICMD	ARIIMAP	ARIIRWI		
			ARIIMSG [1]	[1] See "ARIYM04*" on page 340		
			ARIINITC	ARIIDBS	ARIIFM*	ARIIGM*
				ARIIPRF	ARIISTK	ARIISYS
				ARIITIO	ARISYSD2*	
		ARIIBIN	ARIINME			
	ARIIBIN	ARIIPQY	ARIINSQ	ARIYM04*		
		ARIICMD	ARIIOCI	ARIIFM*	ARIIGM*	ARIPRDI
ARIIFCC	ARIIFMC	ARIIPRTC	ARIIOVD			
		ARIIPRTD				
	ARIIQRY	ARIISET				
	ARIIVFYD	ARIIWRTD	ARIIPFKD	ARIIMGA*		
		ARIIDBS	ARIIPQL	ARIIPSQ		
ARIIHLD	ARIIREC	ARIIRNM	ARIIPQL1*			
		ARIISCC				
	ARIIPSQ	AXTRPRO	ARIIPQY	ARIICNV	ARIIDQY	ARIIFM*
				ARIIFMC	ARIIGM*	ARIIPQY1*
				ARIIPQY2*	ARIIPQY3*	ARIIPQY4*
				ARIIPQY5*	ARIIPQY6*	ARIIPRT
				ARIIRWI	ARIISMG	ARIISQL
				ARIITKN	ARIIVLD	ARIMDFP1*
				ARIYM04		
		ARIIPQY	ARIIPQY1*			
		ARIIPQY	ARIIPQY2*			
		ARIIPQY	ARIIPQY3*			
		ARIIPQY	ARIIPQY4*			
		ARIIPQY	ARIIPQY5*			

* Asterisked items are secondary entry points. See page 346. ° indicates CMS EXEC

Calling Modules			Module/ Entry Pt	Called Modules/Entry Points		
		ARIIPQY	ARIIPQY6*			
		ARIICI2D	ARIIPRF	ARIIRUN	ARIITKN	ARIITS*
				ARIYM04*		
			ARIIPRTC	ARIIFET	ARIIFMT	ARIIHDR
				ARIIOVD	ARIISMG	ARIISQL
				ARIITKN	ARIYM04*	
		ARIIPQY	ARIIPRTD	ARIIDEQ*	ARIIENQ*	ARIIFET
				ARIIFMT	ARIIHDR	ARIIOVD
				ARIIROUT*	ARIISMG	ARIISPGE*
				ARIISQL	ARIISTXT*	ARIITKN
				ARIWAIT*	ARIYM04*	
ARIIHLP	ARIIPQL	ARIISCC	ARIIPSQ	ARIIPQY	ARIISMG	ARIISQL
				ARIITKN	ARIYM04*	AXTCEXTS*
		ARIILST	ARIIQRY	ARIYM04*	ARIIOVD	ARIIRWI
				ARIITKN		
	ARIITIOC	ARIIVFYC	ARIIREAC	ARIIVFYC		
		ARIICMD	ARIIREC	ARIYM04*	ARIIPQL1*	ARIIRWI
				ARIISMG	ARIISQL	ARIISQL1*
				ARIITKN		
ARIICICD	ARIIFULD	ARIIVFYD	ARIIRETD			
		ARIIWRTD				
		ARIICMD	ARIIRNM	ARIYM04*	ARIIPQL1*	ARIIRWI
				ARIISMG	ARIISQL	ARIISQL1*
				ARIITKN		
			ARIIRPT	ARIIFM*	ARIIGM*	ARIYM04*
		ARIISQLA	ARIIRPT1*			
		ARIIFET	ARIIRPT2*			
		ARIIFET	ARIIRPT3*			
		ARIISQLA	ARIIRPT4*			
		ARIIFET	ARIIRPT5*			
		ARIISQL	ARIIRTRN*			
	ARIICMD	ARIIPRF	ARIIRUN	ARIYM04*	ARIISMG	ARIISQL
				ARIISUB	ARIITKN	ARIITS*
ARIIBIN	ARIICMD	ARIIDBS	ARIIRWI	ARIICAN	ARIIFOL	ARIIFOR
ARIIFMC	ARIILST	ARIIMAP		ARIITIO*	ARIITS*	
ARIIREC	ARIIRNM	ARIISCC				
ARIISET	ARIISMG	ARIIST3				
	ARIISUB	ARIITRC				
		ARIICMD	ARIISCC	ARIIFM*	ARIIGM*	ARIIRWI
				ARIITKN	ARIYM04*	
		ARIICMD	ARIISCC1*	ARIYFM *	ARIIGM	ARIIPQL1*
				ARIIPSQ	ARIIRWI	ARIISMG
				ARIISQL	ARIISQL1*	ARIITKN
				ARIYM04*		
ARIICICD	ARIIFULD	ARIIVFYD	ARIISCMD			
		ARIIWRTD				
		ARIICMD	ARIISET	ARIYM04*	ARIIOVD	ARIIST21*
				ARIIST22*	ARIIST23*	ARIIST24*
				ARIIST31*	ARIIST32*	ARIIST33*

* Asterisked items are secondary entry points. See page 346. ◊ indicates CMS EXEC

Calling Modules			Module/ Entry Pt	Called Modules/Entry Points		
				ARIIST34*		
ARIIBIN	ARIICAN	ARIICI2D	ARIISMG	ARIYM04*	ARIIRWI	ARIMSMF
	ARIISQL	ARIISQL8*				
ARIICMD	ARIIDBS	ARIIERS				
ARIIHLP	ARIILST	ARIIPQY				
ARIIPRTC	ARIIPRTD	ARIIPSQ				
ARIIREC	ARIIRNM	ARIIRUN				
		ARIISCC				
ARIISTR	ARIIST2	AXTCMSG				
ARIIBIN	ARIICAN	ARIICMD	ARIISQL*	ARIIRPT1*	ARIIRPT4*	ARIIRTRN
ARIIDBS	ARIIERS	ARIIFET		ARIISMG	ARIITKN	ARIMBIN
ARIIHLP	ARIILST	ARIIPQY		ARIPRDI	ARIYM04*	
ARIIPRTC	ARIIPRTD	ARIIPSQ				
ARIIREC	ARIIRNM	ARIIRUN				
ARIISCC	ARIIST2	ARIISTR				
ARIIERS	ARIILST	ARIIREC	ARIISQL1*	ARIYM04*		
ARIIRNM	ARIISCC	ARIISTR				
		ARIILST	ARIISQL2*	ARIYM04*	ARIPRDI	
		ARIICI2D	ARIISQL3*	ARIPRDI		
		ARIIHLP	ARIISQL4*	ARIPRDI		
		ARIIHLP	ARIISQL5*	ARIPRDI		
		ARIIHLP	ARIISQL6*	ARIPRDI		
		ARIICI2D	ARIISQL7*	ARIPRDI		
		ARIINITC	ARIISQL8*	ARIISMG	ARIYM04*	ARIPRDI
		ARIIHLP	ARIISQL9*	ARIPRDI		
	ARIICI2D	ARIINITC	ARIISTK	ARIIGM*		
		ARIICMD	ARIISTR	ARIISMG	ARIISQL	ARIISQL1*
				ARIITKN	ARIYM04*	
	ARIIDBS	ARIIRWI	ARIISTR1*			
		ARIISET	ARIIST21*	ARIISMG	ARIISQL	ARIITKN
		ARIISET		ARIYM04*		
		ARIISET	ARIIST22*	ARIIOVD	ARIITKN	ARIYM04*
		ARIISET	ARIIST23*	ARIITKN	ARIYM04*	
		ARIISET	ARIIST24*	ARIITKN	ARIYM04*	
		ARIISET	ARIIST31*	ARIITKN	ARIYM04*	
		ARIISET	ARIIST32*	ARIITKN	ARIYM04*	
		ARIISET	ARIIST33*	ARIITKN	ARIYM04*	
		ARIISET	ARIIST34*	ARIITKN	ARIYM04*	
	ARIIRUN	ARIISCC	ARIISUB	ARIIRWI	ARIITKN	ARIYM04*
		ARIITRC	ARIITDP*			
			ARIITIO*	ARIIREA		
ARIIBIN	ARIICMD	ARIIDQY	ARIITKN			
ARIIERS	ARIIFCC	ARIIFCI				
ARIIFCS	ARIIFMC	ARIIFOL				
ARIIHLP	ARIIIGN	ARIILST				
ARIIPQY	ARIIPRF	ARIIPRTC				
ARIIPRTD	ARIIPSQ	ARIIQRY				
ARIIREC	ARIIRNM	ARIIRUN				

* Asterisked items are secondary entry points. See page 346. ⁰ indicates CMS EXEC

Calling Modules			Module/ Entry Pt	Called Modules/Entry Points		
ARIISCC	ARIISET	ARIISQLA				
ARIISTR	ARIIST2	ARIIST3				
	ARIISUB	ARIITRC				
		ARIICMD	ARIITRC	ARIIFM*	ARIIGM*	ARIIRWI
				ARIITDP*	ARIITKN	ARIYM04*
		ARIICICD	ARIITRMD	ARIIWAT	ARIINRTD	
ARIICAN	ARIIDBS	ARIIPRF	ARIITS*			
	ARIIRUN	ARIIRWI				
ARIIBIN	ARIICMD	ARIIPQY	ARIIVLD			
	ARIIREAC	ARIRVRM(C	ARIIVFYC	ARIIREAC		
	ARIICICD	ARIIWRTD	ARIIVFYD	ARIIFULD	ARIIPFKD	ARIIRETD
				ARIISCMD		
ARIIDBS	ARIIPRTD	ARIIRWI	ARIIWAIT*			
		ARIITRM				
		ARIITRMD	ARIIWAT			
		ARIITRMD	ARIIWRTD	ARIIFULD	ARIIPFKD	ARIIRETD
				ARIISCMD	ARIIVFYD	
		ARIOOLRM	ARIIXITB			
ARIIFOR	ARIISQL	ARIMFMT	ARIMBIN			
	ARIXA01	ARIXERD	ARIMCID			
		ARIXII0				
	ARISFIL	ARISIST2				
ARIIFOR	ARIMFMT	ARISDFR	ARIMDEC			
ARISEGB	ARISFIL	ARISINI				
	ARISIST	ARISSET				
ARIDDLO	ARIDSEL	ARIIFMT	ARIMDFP			
		ARIIPQY	ARIMDFP1			
ARIDDUL	ARIXODF	ARIXOTF	ARIMFLP			
ARIDMGE	ARIMTRA	ARIPMSH	ARIMFMT	ARIMBIN	ARIMDEC	ARIMHEX
	ARISERR	ARIXPTT				
		ARIMFMT	ARIMHEX			
		ARIXPTT	ARIMPRT	ARISYSD5*		
		ARIXPTT	ARIMPT1	ARIMPT2	ARIMPT3	ARIMPT4
				ARIMPT5	ARISYSD1*	ARISYSD2*
	ARIMPT1	ARIMPT2	ARIMPT2	ARIMPT2		
		ARIMPT1	ARIMPT3			
		ARIMPT1	ARIMPT4	ARIMPT5		
	ARIMPT1	ARIMPT4	ARIMPT5			
ARIDSQLA	ARIISMG	ARIPMSH	ARIMSMF	ARIIFOR (ISQL version)	ARIMFMT (SQL/DS version)	
		SQLTRFMT[0]	ARIMTRA	ARIMFMT	ARISYSDE*	ARISYSD1*
				ARISYSD5*	ARISYSD7*	ARIXPTT
ARIPPRA	ARIPPRC	ARIPPRP	ARIPCID			
ARIPSCF	ARIPSCN	ARIPSQA				
ARIPSQC	ARIPSQF	ARIPSQP				
		ARIPSSF				
ARIPPRC	ARIPSQC	ARIPTXA	ARIPCVF			

* Asterisked items are secondary entry points. See page 346. [0] indicates CMS EXEC

Calling Modules			Module/ Entry Pt	Called Modules/Entry Points		
	ARIPTXC	ARIPTXP				
ARIPPRA	ARIPFRP	ARIPTXA	ARIPCVH			
ARIPPRC	ARIPTXC	ARIPTXP				
		ARIPTXA	ARIPCVP			
Called by FORTRAN application program			ARIPEIFA			
			ARIFOR (entry point for ARIPPRF)			ARIPMSF
			SQLADD (entry point for ARIPPRF)			ARIPMSF
			SQLLEN (entry point for ARIPPRF)			ARIPMSF
ARIPPRA	ARIPPRC	ARIPPRP	ARIPERR			
ARIPSQA	ARIPSQC	ARIPSQP				
		ARIPTXC				
		ARIPPRP	ARIPLTP			
ARIPDCF	ARIPPIF	ARIPPRF	ARIPMSF	ARINFMT	ARIPSMF	ARISYSD*
ARIPSCF	ARIPSQF	ARIPSSF		ARISYSD5*		
		ARIPTXF				
ARIPPRA	ARIPPRC	ARIPFRP	ARIPMSH	ARINFMT	ARIMSMF	ARISYSD3*
ARIPSCN	ARIPTXA	ARIPTXC		ARISYSD5*		
		ARIPTXP				
ARIPPRA	ARIPPRC	ARIPPRP	ARIPMVL			
	ARIPSQA	ARIPSQP				
		ARIPPRF	ARIPPIFA	ARIPMSF		
			ARIPPRA	ARIPCID	ARIPCVH	ARIPERR
				ARIPMSH	ARIPMVL	ARIPRDI
				ARIPSCN	ARIPSQA	ARIPTXA
				ARISYSD*	ARISYSD1*	ARISYSD2*
				ARISYSD3*	ARISYSD5*	ARISYSD8*
			ARIPPRC	ARIPCID	ARIPCVF	ARIPERR
				ARIPMSH	ARIPMVL	ARIPRDI
				ARIPSCN	ARIPSQC	ARIPTXC
				ARISYSD*	ARISYSD1*	ARISYSD2*
JCL (VSE) or EXECs (VM)			ARIPPRF	ARIPDCF	ARIPMSF	ARIPPIF
				ARIPSCF	ARIPSQF	ARIPSSF
				ARIPTXF	ARISYSD1*	ARISYSD2*
				ARISYSD5*		
				ARISYSD3*	ARISYSD5*	ARISYSD8*
			ARIPPRP	ARIPCID	ARIPCVH	ARIPERR
				ARIPLTP	ARIPMSH	ARIPMVL
				ARIPRDI	ARIPSCN	ARIPSQP
				ARIPTXP	ARISYSD*	ARISYSD1*
				ARISYSD2*	ARISYSD3*	ARISYSD5*
				ARISYSD8*		
ARIDSQLA	ARIIOCI	ARIISQLA	ARIPRDI			
ARIPPRA	ARIPPRC	ARIPFRP				
	ARIRSQLA	AXTCSQL				
		ARIPPRF	ARIPSCF	ARIPCID	ARIPMSF	
ARIPPRA	ARIPPRC	ARIPPRP	ARIPSCN	ARIPCID	ARIPMSH	
		ARIPPRA	ARIPSQA	ARIPCID	ARIPERR	ARIPMVL
				ARISYSD5*	ARISYSD8*	

* Asterisked items are secondary entry points. See page 346. ° indicates CMS EXEC

Calling Modules			Module/ Entry Pt	Called Modules/Entry Points		
		ARIPPRC	ARIPSQC	ARIPCID	ARIPCVF	ARIPERR
				ARISYSD5*	ARISYSD8*	
		ARIPPRF	ARIPSQF	ARIPCID	ARIPMSF	ARISYSD5*
				ARISYSD8*		
		ARIPPRP	ARIPSQP	ARIPCID	ARIPERR	ARIPMVL
				ARISYSD5*	ARISYSD8*	
		ARIPPRF	ARIPSSF	ARIPCID	ARIPMSF	ARIPPIF
		ARIPPRA	ARIPTXA	ARIPCVF	ARIPCVH	ARIPCVP
				ARIPMSH	ARISYSD2*	ARISYSD5*
		ARIPPRC	ARIPTXC	ARIPCVF	ARIPERR	ARIPMSH
				ARISYSD2*	ARISYSD3*	ARISYSD5*
		ARIPPRF	ARIPTXF	ARIPMSF	ARISYSD5*	
		ARIPPRP	ARIPTXP	ARIPCVF	ARIPCVH	ARIPMSH
				ARISYSD2*	ARISYSD5*	
		ARIRIRM	ARIRBRM	ABNDPRT	ARICCOM	ARICIMB
				ARICMMB	ARISYSDA*	ARISYSD2*
				ARISYSD7*	ARIYM04	
			ARIRB13	ARISYSD3*		
			ARIRB51	ARISYSD4*		
			ARIRCL1(C	ARIRCL2(C		
		ARIRCL1(C	ARIRCL2(C			
		ARIRENTD	ARIRDIS(D	ARICCOM	ARICTKN	ARIRECI
				ARIYE08		
ARIRDIS(D	ARIRENAD	ARIRO13	ARIRECI	ARISYSD1*	ARISYSD2*	
		ARIRENTD	ARIRENA	ARICCOM	ARICTKN	ARIRECI
				ARIRSQL	ARISYSDC*	ARIYE08
			ARIRENTD	ARIRDIS	ARIRENA	DFHEAI0
			ARIRIRM	ARICCOM	ARIRBRM	ARISYSDC*
				ARISYSD1*	ARISYSD2*	ARISYSD6*
				ARISYSD7*		
			ARIRMIID	ARIRORM		
		ARIRMIID	ARIRORM	ARICCOM	ARIRSEN	ARIRSQL
				ARISYSD2*		
			ARIRO13	ARIRECI	ARISYSD3*	
		ARIRORMD	ARIRSEN(D	ARICCOM	ARICIMB	ARICMMB
	ARIRENAD	ARIRORMD	ARIRSQL	ARIPRDI		
			ARIRTL1(C	ARIRTL2(C		
		ARIRTL1(C	ARIRTL2(C			
		ARIRVST(C	ARIRVIR(C	ARICCOM	ARIRVRM	ARISYSD1*
				ARISYSD2*	ARISYSD7*	
		ARIRVIR(C	ARIRVRM(C	ARICCOM	ARICIMB	ARICMMB
				ARISYSDA*	ARISYSD2*	ARISYSD7*
				ARIYM04		
			ARIRVST(C	ARISRMK(C	ARIRVIR(C	
			ARISAVES[0]	ARICEMGC[0]	ARISEMSG[0]	ARISQLLD[0]
ARISFIL	ARISIST	ARISIST2	ARISCNV	ARISERR	ARISYSD7*	
ARISDFM	ARISIST	ARISIST2	ARISCOL	ARISERR	ARISYSD7*	
		ARISUPD				
		SQLSTART[0]	ARISDBBT			

* Asterisked items are secondary entry points. See page 346. [0] indicates CMS EXEC

Calling Modules			Module/ Entry Pt	Called Modules/Entry Points		
		SQLGENLD[0]	ARISDBKC			
	SQL/DS data base machine owner		ARISDBMA[0]	ARICEMGC[0]	ARISEMSG[0]	ARISQLLD[0]
				SQLSTART[0]		
		ARISFDB	ARISDBR	ARICWSG	ARISYSD5*	ARISYSD7*
				ARIYE08	ARIYM02	
		ARISMAI	ARISDFL	ARISERR	ARISREL	ARISYSD7*
		ARISMAI	ARISDFM	ARISCOL	ARISERR	ARISREL
				ARISYSD7*		
		ARISMAI	ARISDFR	ARIMDEC	ARISEDI	ARISERR
				ARISYSD7*	ARISYSD8*	
	ARIYI22(C	ARIYT00	ARISDSK	ARISFM1	ARISYSD5*	ARISYSD7*
				ARIYE04	ARIYE08	ARIYI312*
				ARIYI313*	ARIYI315*	
		ARISYSDD	ARISD00D	ARISD10D	ARISD11D	ARISD12D
				ARISD13D	ARISD14D	ARISD15D
				ARISD16D	ARISD20D	ARISD21D
				ARISD22D	ARISD23D	ARISD24D
				ARISD25D	ARISD26D	ARISD30D
				ARISD31D	ARISD32D	ARISD33D
		ARISYSDD	ARISD01D	SLIB		
		ARISD00D	ARISD10D			
		ARISD00D	ARISD11D			
		ARISD00D	ARISD12D			
		ARISD00D	ARISD13D			
		ARISD00D	ARISD14D			
		ARISD00D	ARISD15D			
		ARISD00D	ARISD16D			
		ARISD00D	ARISD20D			
		ARISD00D	ARISD21D			
		ARISD00D	ARISD22D			
		ARISD00D	ARISD23D			
		ARISD00D	ARISD24D			
		ARISD00D	ARISD25D			
		ARISD00D	ARISD26D			
	ARISD00D	ARISYSDD	ARISD30D			
		ARISD00D	ARISD31D			
		ARISD00D	ARISD32D			
		ARISD00D	ARISD33D			
ARISDFR	ARISMAI	ARISUPD	ARISEDI	ARISERR	ARISYSD7*	
		ARIYT00	ARISEGA	ARISFM2	ARISYSD5*	ARISYSD7*
				ARIYE04	ARIYE08	ARIYI261*
				ARIYI313*	ARIYL08	ARIYM02
			ARISEGB	ARIMDEC	ARISERR	ARISSYSB
				ARISYSD7*		
ARISAVES[0]	ARISDBMA[0]	ARISSLKE[0]	ARISEMSG[0]	ARICEMGC[0]		
ISQL[0]	SQLDBINS[0]	SQLDBSU[0]				
ISQL[0]	SQLDBINS[0]	SQLDBSU[0]	ARISESCP[0]	ARICEMGC[0]		
ARISCNV	ARISCOL	ARISDFL	ARISERR	ARIMFMT	ARISYSD3*	ARISYSD5*
ARISDFM	ARISDFR	ARISEDI				

* Asterisked items are secondary entry points. See page 346. [0] indicates CMS EXEC

Calling Modules			Module/ Entry Pt	Called Modules/Entry Points		
ARISEGB	ARISFIL	ARISINI				
ARISIST	ARISIST2	ARISLAYD				
ARISMAI	ARISSET	ARISUPD				
		ARISFM1(C	ARISFDB	ARICWSG	ARISDBR	ARISII01*
		SQLSTART[0]	ARISFDEF			
				ARISSCB	ARISYSD7*	ARIYM02
		ARISMAI	ARISFIL	ARIMBIN	ARIMDEC	ARISCNV
				ARISERR	ARISREL	ARISYSD7*
ARISDSK	ARISPFM	ARIYL14	ARISFM1(C	ARICCOM	ARICTRM	ARICWSG
				ARICWSF	ARISFDB	ARIYE07
				ARIYE14	ARIYK18	ARIYM02
		ARIYL21		ARIYM04		
ARISDSK	ARISPFM	ARIYL14	ARISFM1(D	ARICWSF	ARICWSG	ARISYSD2*
				ARISYSD7*	ARIYE07	ARIYM04
				ARIYM02		
	ARISEGA	ARISPFM	ARISFM2			
		ARIYI22D	ARISII0	ARICWSF	ARICWSG	ARISYSD7*
				ARIYE07	ARIYI13	ARIYI311*
		ARISFDB	ARISII01*			
		ARISMAI	ARISINI	ARIMDEC	ARISERR	ARISYSD7*
		ARISMAI	ARISIST	ARIMDEC	ARISCNV	ARISCOL
				ARISERR	ARISIST2	ARISIST3
				ARISREL	ARISYSD1*	ARISYSD7*
				ARISYSD8*		
		ARISIST	ARISIST2	ARIMBIN	ARISCNV	ARISCOL
				ARISERR	ARISREL	ARISYSD7*
		ARISIST	ARISIST3			
		ARISMAI	ARISLAY	ARISERR	ARISYSD7*	
	User and	ARISSLKE[0]	ARISLKIT[0]	ARICEMGC[0]	ARISQLLD[0]	
			ARISMAI	ARISDFL	ARISDFM	ARISDFR
				ARISEDI	ARISERR	ARISFIL
				ARISINI	ARISIST	ARISLAY
				ARISSET	ARISUPD	ARISYSD1*
				ARISYSD2*	ARISYSD5*	ARISYSD7*
				ARISYSD8*		
			ARISOCMD	ARISYSD6*		
			ARISFDFC[0]	ARICEMGC[0]	ARISPROD[0]	
		ARIYT00	ARISPFM	ARISFM1	ARISFM2	ARISYSD5*
				ARISYSD7*	ARIYE04	ARIYE08
				ARIYI13	ARIYI22	ARIYI312*
				ARIYI313*	ARIYI315*	ARIYM02
ARISPDFC[0]	SQLADBEX[0]	SQLADBSP[0]	ARISPROD[0]	ARICEMGC[0]		
SQLDBGEN[0]	SQLDBINS[0]	SQLGENLD[0]				
	SQLLOG[0]	SQLSTART[0]				
ARISAVES[0]	ARISDBMA[0]	ARISLKIT[0]	ARISQLLD[0]	ARICEMGC[0]		
ARISSLKE[0]	SQLADBEX[0]	SQLADBSP[0]				
SQLDBGEN[0]	SQLDBINS[0]	SQLDBSU[0]				
SQLGENLD[0]	SQLLOG[0]	SQLPREP[0]				

* Asterisked items are secondary entry points. See page 346. [0] indicates CMS EXEC

Calling Modules			Module/ Entry Pt	Called Modules/Entry Points		
	SQLSTART°	SQLTRFMT°				
ARISDFL	ARISDFM	ARISFIL	ARISREL			
ARISIST	ARISIST2	ARISUPD				
	ARIRVST(C	SQLDBINS°	ARISRMK(C			
	ARISFDB	ARIYI04	ARISSCB	ARIYE07	ARIYE13	
			ARISSERV°	ARICEMGC°		
		ARISMAI	ARISSET	ARIMDEC	ARISERR	ARISYSD7*
		5748XXJ°	ARISSLKE°	ARICEMGC°	ARISEMSG°	ARISLKIT°
				ARISQLLD°		
		ARISEGB	ARISSYSB			
		ARISMAI	ARISUPD	ARISCOL	ARISEDI	ARISERR
				ARISREL	ARISYSD7*	ARISYSD8*
ARIPPRA	ARIPPRC	ARIPPRP	ARISYSD*			
ARICIPI	AR_CPDM	ARICWSR	ARISYSDA*			
ARIDALC	ARIDALI	ARIDALT				
ARIDBS	ARIDCFI	ARIDCSP				
ARIDDFI	ARIDDLO	ARIDEXI				
ARIDSFA	ARIDSQLA	ARIDUNL				
	ARIRBRM	ARIRVRM(C				
		ARIYM01	ARISYSDB*			
			ARISYSDC	ARISYSD3*		
ARICIPI(D	ARIRENAD	ARIRIRM	ARISYSDC*			
			ARISYSDD	ARISYSD1*	ARISYSD3*	ARISYSD5*
				ARISYSD8*		
	ARICIPI(C	ARIMTRA	ARISYSDE*			
ARIXEDP	ARIXEPP	ARIXOOP	ARISYSDF*			
ARICABE	ARICDMP	ARICGSE	ARISYSDG*	Called by many modules in Assembler prolog code to obtain module automatic storage.		
ARICIPI	ARICPRM	ARIYL13				
	ARICTRM	ARIYE13	ARISYSDH*			
ARICWSG	ARIXEAB	ARIXELX	ARISYSDI*			
		ARIXERA				
	ARISYSEC	ARIYM10	ARISYSDJ*			
ARICABE	ARICIPI	ARICTRM	ARISYSDK*			
ARICIMB	ARICWSG	ARICWSI	ARISYSD1*			
		ARIDDUL				
ARIDALC	ARIDALI	ARIDALT				
ARIDBS	ARIDDLO	ARIDREL				
		ARIRVIR(C				
ARIDSEL	ARIDSFA	ARIDSQLA				
ARIDUNL	ARIMPTI	ARIMTRA				
ARIPPRA	ARIPPRC	ARIPPRP				
ARIRECID	ARIRIRM	ARISIST				
ARISMAI	ARISYSDD	ARIXEAB				
	ARIXELX	ARIXERO				
		ARIXPTT				
ARICDMP	ARICIMB	ARICIPI	ARISYSD2*	Called by many modules in Assembler epilog code to free module automatic storage.		
ARICWFR	ARICWSR	ARIDALC				
ARIDALI	ARIDALT	ARIDBS				
ARIDDLO	ARIDDUL	ARIDREL				

* Asterisked items are secondary entry points. See page 346. ° indicates CMS EXEC

Calling Modules			Module/ Entry Pt	Called Modules/Entry Points		
ARIDSEL	ARIDSFA	ARIDSQLA				
ARIDUNL	ARIMPTI	ARIPPRA				
ARIPPRC	ARIPPRP	ARIPTXA				
ARIPTXC	ARIPTXP	ARIRBRM				
ARIRECID	ARIRIRM	ARIRORMD				
ARIRVRM(C	ARISFMID	ARISMAI				
ARIXEFB	ARIXPTT	ARIYL13				
ARICIPI	ARIDMGE	ARIPMSH	ARISYSD3*			
ARIPPRA	ARIPPRC	ARIPPRP				
ARIPTXC	ARIRB13	ARIRO13				
	ARISERR	ARISYSDD				
		ARISYSGC				
		ARIRB51	ARISYSD4*			
ARICTRC	ARICTRI	ARIDCFI	ARISYSD5*			
ARIDDFI	ARIDMGE	ARIMPRT				
ARIMTRA	ARIPMSH	ARIPPRA				
ARIPPRC	ARIPPRP	ARIPSQA				
ARIPSQC	ARIPSQP	ARIPTXA				
ARIPTXC	ARIPTXP	ARISDBR				
ARISDSK	ARISEGA	ARISERR				
ARISMAI	ARISPFM	ARISYSDD				
ARIXETR	ARIYE14	ARIYL21				
	ARIYL22	ARIYL23				
ARICIPI(D	ARICSPM(D	ARIDEXI	ARISYSD6*			
	ARIRIRM	ARISOCMD				
ARIMTRA	ARIRBRM	ARIRIRM	ARISYSD7*			
ARIRVRM(C	ARISCNV	ARISCOL				
ARISDBR	ARISDFL	ARISDFM				
ARISDFR	ARISDSK	ARISEDI				
ARISEGA	ARISEGB	ARISFDB				
ARISFIL	ARISFMI(D	ARISIIO				
ARISINI	ARISIST	ARISIST2				
ARISLAYD	ARISMAI	ARISPFM				
ARISSET	ARISUPD	ARIXPTT				
		ARIYL22				
ARICPRM	ARIPPRA	ARIPPRC	ARISYSD8*			
ARIPPRP	ARIPSQA	ARIPSQC				
ARIPSQP	ARISDFR	ARISIST				
ARISMAI	ARISUPD	ARISYSDD				
		ARISYSFC				
		ARICIPI(C	ARISYSD9*			
ARICTRC	ARICTRI	ARIDCFI	ARISYSEC	ARISYSD8*	ARISYSD3*	ARISYSDJ*
ARIDDFI	ARIDMGE	ARIMPRT				
ARIMTRA	ARIPMSH	ARIPPRA				
ARIPPRC	ARIPPRP	ARIPSQA				
ARIPSQC	ARIPSQP	ARIPTXA				
ARIPTXC	ARIPTXP	ARISDBR				
ARISDSK	ARISEGA	ARISERR				

* Asterisked items are secondary entry points. See page 346. 0 indicates CMS EXEC

Calling Modules			Module/ Entry Pt	Called Modules/Entry Points		
ARISMAI	ARISPFM	ARIXETR				
ARIYE14	ARIYL21	ARIYL22				
		ARIYL23				
ARICPRM	ARISDFR	ARISIST	ARISYSFC			
	ARISMAI	ARISUPD				
	ARICTRM	ARIYE13	ARISYSGC	ARISYS03*		
		User	ARISYSIN[0]	I5748XXJ[0]		
		ARIXERD	ARIXA01	ARICWSG	ARIMCID	ARIXECK
				ARIXEDB	ARIXEER	ARIXSLN
ARIXA06	ARIXA07	ARIXA09	ARIXA02	ARIXECK	ARIXEDB	ARIXEER
ARIXELX	ARIXERD	ARIXI03		ARIXSCF	ARIXSLN	
ARIXI07	ARIXI09	ARIXI15				
ARIXI25	ARIXOIN	ARIXOOP				
	ARIXOSC	ARIXOVD				
ARIXA04	ARIXA08	ARIXA09	ARIXA03	ARIXECK	ARIXEDB	
ARIXA04	ARIXA11	ARIXI30	ARIXA04	ARIXA03	ARIXA04	ARIXECK
		ARIXI32		ARIXEDB	ARIXSLN	
ARIXA07	ARIXEDS	ARIXERP	ARIXA06	ARICWSF	ARICWSG	ARIXA02
ARIXI07	ARIXI09	ARIXI13		ARIXECK	ARIXEDB	ARIXEER
ARIXOIN	ARIXOOP	ARIXOSC		ARIXSLN	ARIXSUT	
		ARIXI14	ARIXA07	ARICWSF	ARICWSG	ARIXA02
				ARIXA06	ARIXECK	ARIXEDB
				ARIXEER	ARIXSLN	
	ARIXA08	ARIXA09	ARIXA08	ARICWSG	ARIXA03	ARIXA06
				ARIXA10	ARIXA11	ARIXECK
				ARIXEDB	ARIXEER	ARIXI102*
				ARIXSLN		
		ARIXI14	ARIXA09	ARIXA02	ARIXA03	ARIXA08
				ARIXECK	ARIXEDB	ARIXEER
				ARIXI102*	ARIXSLN	
		ARIXA08	ARIXA10	ARICWSF	ARICWSG	ARIXECK
				ARIXEDB	ARIXEER	ARIXSLN
		ARIXA08	ARIXA11	ARIXA04	ARIXECK	ARIXEDB
				ARIXSLN		
		ARIXFTB	ARIXCOT			
		ARIXF61	ARIXCRA	ARIXCR1	ARIXEDB	
		ARIXF59	ARIXCRB	ARIXCR1	ARIXEDB	
		ARIXF63	ARIXCRC	ARIXCR1	ARIXEDB	
ARIXCRG	ARIXCR3	ARIXF21	ARIXCRD	ARIXECK	ARIXEDB	
		ARIXF30				
		ARIXF94	ARIXCRE	ARICGSE		
		ARIXF96	ARIXCRF			
	ARIXF20	ARIXF48	ARIXCRG	ARIXCRD	ARIXCR3	
		ARIXF32	ARIXCRH			
	ARIXCR2	ARIXF33	ARIXCRI			
	ARIXCR2	ARIXF34	ARIXCRJ			
		ARIXF44	ARIXCRL			
ARIXCR0	ARIXF75	ARIXF81	ARIXCR0			
ARIXCRA	ARIXCRB	ARIXCRC	ARIXCR1	ARICFSE	ARIXEBR	ARIXECK

* Asterisked items are secondary entry points. See page 346. [0] indicates CMS EXEC

Calling Modules			Module/ Entry Pt	Called Modules/Entry Points		
ARIXCR2	ARIXCR3	ARIXCR4		ARIXEER		
(Also called by most of the code fragments)						
		ARIXF06	ARIXCR2	ARICOMB	ARICWSF	ARICWSG
				ARIXCRI	ARIXCRJ	ARIXCR1
				ARIXEDB		
	ARIXCRG	ARIXFA0	ARIXCR3	ARICWSG	ARIXCRD	ARIXCR1
				ARIXECK	ARIXEDB	
	ARIXF21	ARIXF47	ARIXCR4	ARIXCR1		
		ARIXF50	ARIXCR5			
		ARIXF51	ARIXCR6			
		ARIXF52	ARIXCR7			
		ARIXF53	ARIXCR8			
ARIXC03	ARIXC21	ARIXC37	ARIXC01	ARIXEER		
	ARIXC38	ARIXEPP				
		ARIXC38	ARIXC02	ARIXC08	ARIXC18	ARIXC20
				ARIXC33	ARIXC39	ARIXEER
				ARIXSCH		
		ARIXC38	ARIXC03	ARIXC01	ARIXC08	ARIXC11
				ARIXC18	ARIXC20	ARIXC33
				ARIXC41	ARIXC42	ARIXC43
				ARIXC48	ARIXC49	ARIXC51
				ARIXEER	ARIXSCH	
ARIXC10	ARIXC15	ARIXC28	ARIXC05	ARIXC08	ARIXC18	ARIXC20
ARIXC30	ARIXC31	ARIXC32		ARIXC33	ARIXC36	ARIXC39
		ARIXC38		ARIXC45	ARIXC48	ARIXEER
				ARIXSCH		
		ARIXC38	ARIXC06	ARIXC08	ARIXC18	ARIXC20
				ARIXC33	ARIXC39	ARIXC40
				ARIXC41	ARIXC42	ARIXC43
				ARIXC48	ARIXEER	ARIXSCH
	ARIXC18	ARIXC25	ARIXC07	ARIXC39	ARIXEER	
ARIXC02	ARIXC03	ARIXC05	ARIXC08	ARIXC48		
ARIXC06	ARIXC16	ARIXC21				
ARIXC27	ARIXC28	ARIXC34				
	ARIXC36	ARIXC45				
		ARIXC38	ARIXC10	ARIXC05	ARIXC36	ARIXC39
				ARIXC40	ARIXC41	ARIXC47
				ARIXC48	ARIXEER	
ARIXC03	ARIXC23	ARIXC24	ARIXC11	ARIXC41		
		ARIXC30				
		ARIXC37	ARIXC13	ARICWSG	ARIXEAB	
			ARIXC14	ARICWSF	ARIXEER	
		ARIXC38	ARIXC15	ARIXC05	ARIXEER	
		ARIXC38	ARIXC16	ARIXC08	ARIXC20	ARIXC40
				ARIXC41	ARIXC42	
			ARIXC17	ARIXC41	ARIXC43	ARIXC53
ARIXC02	ARIXC03	ARIXC05	ARIXC18	ARIXC07	ARIXC39	ARIXEER
ARIXC06	ARIXC21	ARIXC24		ARIXSCF	ARIXSCH	
ARIXC28	ARIXC30	ARIXC31				

* Asterisked items are secondary entry points. See page 346. [0] indicates CMS EXEC

Calling Modules			Module/ Entry Pt	Called Modules/Entry Points		
	ARIXC36	ARIXC38				
		ARIXC38	ARIXC19	ARICWSF	ARICWSG	ARIXEAB
				ARIXEER		
ARIXC02	ARIXC03	ARIXC05	ARIXC20	ARIXC40	ARIXC41	ARIXC47
ARIXC06	ARIXC16	ARIXC21		ARIXC48	ARIXC49	ARIXEER
ARIXC27	ARIXC28	ARIXC34		ARIXSCH		
	ARIXC36	ARIXC45				
		ARIXC38	ARIXC21	ARICWSF	ARICWSG	ARIXC01
				ARIXC08	ARIXC18	ARIXC20
				ARIXC33	ARIXC40	ARIXC41
				ARIXC47	ARIXC48	ARIXEAB
				ARIXEER	ARIXSCH	STUFFIT
		ARIXC38	ARIXC22	ARIXC36	ARIXC48	ARIXEER
ARIXC24	ARIXC25	ARIXC30	ARIXC23	ARIXC11	ARIXC36	ARIXC43
		ARIXC38		ARIXC47	ARIXC48	ARIXC49
				ARIXEER		
		ARIXC38	ARIXC24	ARIXC11	ARIXC18	ARIXC23
				ARIXC40	ARIXC41	ARIXC42
				ARIXC47	ARIXC48	ARIXEER
				ARIXSCH		
		ARIXC38	ARIXC25	ARIXC07	ARIXC23	ARIXC39
				ARIXC40	ARIXC41	
		ARIXC38	ARIXC27	ARIXC08	ARIXC20	ARIXC33
				ARIXC40	ARIXC41	ARIXC48
				ARIXEER		
		ARIXC38	ARIXC28	ARIXC05	ARIXC08	ARIXC18
				ARIXC20	ARIXC33	ARIXC36
				ARIXC39	ARIXC40	ARIXC41
				ARIXC42	ARIXC43	ARIXC45
				ARIXC48	ARIXEER	ARIXSCH
		ARIXC38	ARIXC29	ARIXEER		
		ARIXC38	ARIXC30	ARIXC05	ARIXC11	ARIXC18
				ARIXC23	ARIXC39	ARIXC40
				ARIXC41	ARIXC42	ARIXC48
				ARIXC49	ARIXEER	ARIXSCH
		ARIXC38	ARIXC31	ARIXC05	ARIXC18	ARIXC39
				ARIXC40	ARIXC41	ARIXC48
				ARIXEER		
		ARIXC38	ARIXC32	ARIXC05	ARIXC36	ARIXC39
				ARIXC40	ARIXC41	ARIXC42
				ARIXC48	ARIXEER	
ARIXC02	ARIXC03	ARIXC05	ARIXC33	ARIXC48	ARIXEER	ARIXSCH
ARIXC06	ARIXC21	ARIXC27				
ARIXC28	ARIXC34	ARIXC36				
		ARIXC38	ARIXC34	ARIXC08	ARIXC20	ARIXC33
				ARIXC39	ARIXC40	ARIXC41
				ARIXC42	ARIXC43	ARIXEER
				ARIXSCH		

* Asterisked items are secondary entry points. See page 346. ° Indicates CMS EXEC

Calling Modules			Module/ Entry Pt	Called Modules/Entry Points		
		ARIXC38	ARIXC35	ARIXC36	ARIXC48	
ARIXC05	ARIXC10	ARIXC22	ARIXC36	ARIXCPR	ARIXC08	ARIXC18
ARIXC23	ARIXC28	ARIXC32		ARIXC20	ARIXC33	ARIXC39
		ARIXC35		ARIXC40	ARIXC41	ARIXC42
				ARIXC43	ARIXC45	ARIXC48
				ARIXC49	ARIXEER	ARIXSCH
	ARIXEDP	ARIXEPP	ARIXC37	ARICWSF	ARICWSG	ARIXC01
				ARIXC13	ARIXC38	ARIXEER
		ARIXC37	ARIXC38	ARIXC01	ARIXC02	ARIXC03
				ARIXC05	ARIXC06	ARIXC10
				ARIXC15	ARIXC16	ARIXC18
				ARIXC19	ARIXC21	ARIXC22
				ARIXC23	ARIXC24	ARIXC25
				ARIXC27	ARIXC28	ARIXC29
				ARIXC30	ARIXC31	ARIXC32
				ARIXC34	ARIXC35	ARIXC39
				ARIXC47	ARIXC48	ARIXC49
				ARIXC54	ARIXEER	
ARIXC02	ARIXC05	ARIXC06	ARIXC39	ARIXEER	ARIXSCH	
ARIXC07	ARIXC10	ARIXC18				
ARIXC25	ARIXC28	ARIXC30				
ARIXC31	ARIXC32	ARIXC34				
ARIXC36	ARIXC38	ARIXC53				
ARIXC06	ARIXC10	ARIXC16	ARIXC40	ARIXEER		
ARIXC20	ARIXC21	ARIXC24				
ARIXC25	ARIXC27	ARIXC28				
ARIXC30	ARIXC31	ARIXC32				
	ARIXC34	ARIXC36				
ARIXC03	ARIXC06	ARIXC10	ARIXC41	ARIXEER		
ARIXC11	ARIXC16	ARIXC17				
ARIXC20	ARIXC21	ARIXC24				
ARIXC25	ARIXC27	ARIXC28				
ARIXC30	ARIXC31	ARIXC32				
ARIXC34	ARIXC36	ARIXC45				
	ARIXC53	ARIXC54				
ARIXC03	ARIXC06	ARIXC16	ARIXC42	ARIXEER		
ARIXC24	ARIXC28	ARIXC30				
ARIXC32	ARIXC34	ARIXC36				
	ARIXC53	ARIXC54				
ARIXC03	ARIXC06	ARIXC17	ARIXC43	ARIXEER		
ARIXC23	ARIXC28	ARIXC34				
	ARIXC36	ARIXC53				
ARIXC05	ARIXC28	ARIXC36	ARIXC45	ARIXC08	ARIXC20	ARIXC41
				ARIXC48	ARIXC49	ARIXEER
ARIXC10	ARIXC20	ARIXC21	ARIXC47	ARIXC48	ARIXEER	
ARIXC23	ARIXC24	ARIXC38				
ARIXC03	ARIXC05	ARIXC06	ARIXC48	ARIXEER	ARIXSCH	
ARIXC08	ARIXC10	ARIXC20				
ARIXC21	ARIXC22	ARIXC23				

* Asterisked items are secondary entry points. See page 346. [0] indicates CMS EXEC

Calling Modules			Module/ Entry Pt	Called Modules/Entry Points		
ARIXC24	ARIXC27	ARIXC28				
ARIXC30	ARIXC31	ARIXC32				
ARIXC33	ARIXC35	ARIXC36				
ARIXC38	ARIXC45	ARIXC47				
		ARIXC51				
ARIXC03	ARIXC20	ARIXC23	ARIXC49			
ARIXC30	ARIXC36	ARIXC38				
		ARIXC45				
		ARIXC03	ARIXC51	ARIXC48	ARIXEER	ARIXSCH
		ARIXC17	ARIXC53	ARIXC39	ARIXC41	ARIXC42
				ARIXC43		
		ARIXC38	ARIXC54	ARIXC41	ARIXC42	
ARIXC13	ARIXC19	ARIXC21	ARIXEAB	ARICWSG	ARISYSDI*	ARIXEER
ARIXEDP	ARIXEDS	ARIXELX		ARIXSCH		
ARIXERD	ARIXERP	ARIXOBY				
		ARIXONW				
	ARIXEDR	ARIXERD	ARIXEAD	ARICFSE	ARICGSE	ARICOMB
				ARICWSF	ARICWSG	ARIXA06
				ARIXEAB	ARIXECK	ARIXEDB
				ARIXEER	ARIXEFB	ARIXELX2
				ARIXELX3	ARIXEPH	ARIXEPP
				ARIXERP	ARIXESX4	ARIXESX5
				ARIXSCF	ARIXSRT	
		ARIXERD	ARIXEAD1			
	ARIXEDR	ARIXERD	ARIXEAD2			
ARIXCR1	ARIXELK	ARIXERD	ARIXEBR	ARICFSE	ARICGSE	ARIXEDB
				ARIXEER		
		ARIXELK	ARIXEBR1*			
ARIXA01	ARIXA02	ARIXA03	ARIXECK	ARIXEDB		
ARIXA04	ARIXA06	ARIXA07				
ARIXA08	ARIXA09	ARIXA10				
ARIXA11	ARIXCRD	ARIXCR1				
ARIXCR3	ARIXECL	ARIXEDR				
ARIXELX	ARIXEPP	ARIXERD				
ARIXERP	ARIXESX	ARIXI01				
ARIXI03	ARIXI04	ARIXI06				
ARIXI07	ARIXI08	ARIXI09				
ARIXI12	ARIXI13	ARIXI15				
ARIXI16	ARIXI18	ARIXI19				
ARIXI21	ARIXI22	ARIXI23				
ARIXI24	ARIXI25	ARIXI26				
ARIXI27	ARIXI28	ARIXI29				
ARIXI30	ARIXI31	ARIXI32				
ARIXOCA	ARIXOCK	ARIXOD1				
ARIXOD2	ARIXOIU	ARIXOUS				
		ARIXERD	ARIXECL	ARIXECK	ARIXEDB	ARIXEER
ARIXELK	ARIXERD	ARIXERP	ARIXECW			
ARIXESX	ARIXI12	ARIXI16				

* Asterisked items are secondary entry points. See page 346. 0 indicates CMS EXEC

Calling Modules			Module/ Entry Pt	Called Modules/Entry Points		
		ARIXSRT				
ARIXA01	ARIXA02	ARIXA03	ARIXEDB			
ARIXA04	ARIXA06	ARIXA07				
ARIXA08	ARIXA09	ARIXA10				
ARIXA11	ARIXCRA	ARIXCRB				
ARIXCRC	ARIXCRD	ARIXCR2				
ARIXCR3	ARIXEBR	ARIXECK				
ARIXECL	ARIXELX	ARIXEOC				
ARIXERD	ARIXERP	ARIXESX				
ARIXI01	ARIXI03	ARIXI04				
ARIXI06	ARIXI07	ARIXI08				
ARIXI09	ARIXI12	ARIXI13				
ARIXI15	ARIXI16	ARIXI18				
ARIXI19	ARIXI20	ARIXI21				
ARIXI22	ARIXI23	ARIXI24				
ARIXI25	ARIXI26	ARIXI27				
ARIXI28	ARIXI29	ARIXI30				
ARIXI31	ARIXI32	ARIXOCA				
ARIXOCK	ARIXOD1	ARIXOD2				
ARIXOIU	ARIXOUS	ARIXSRT				
		ARIXERD	ARIXEDC	ARICOMB		
		ARIXERD	ARIXEDP	ARICWSF	ARICWSG	ARISYSDF*
				ARIXC37	ARIXEAB	ARIXEER
				ARIXEFB	ARIXOOP	ARIXPA1
				ARIXSCF	ARIXSCH	
		ARIXERD	ARIXEDP1*			
		ARIXERD	ARIXEDR	ARICFSE	ARICGSE	ARIXEAD
				ARIXEAD2	ARIXECK	ARIXEDB
				ARIXEER	ARIXEFB	ARIXSCH
				ARIXSCN	ARIXSLN	
		ARIXERD	ARIXEDS	ARICOMB	ARICWSF	ARICWSG
				ARIXA06	ARIXEAB	ARIXEER
				ARIXEFB	ARIXEPP	ARIXESX1*
				ARIXESX3*	ARIXI18	
ARIXA01	ARIXA02	ARIXA06	ARIXEER			
ARIXA07	ARIXA08	ARIXA09				
ARIXA10	ARIXCR1	ARIXC01				
ARIXC02	ARIXC03	ARIXC05				
ARIXC06	ARIXC07	ARIXC10				
ARIXC14	ARIXC15	ARIXC18				
ARIXC19	ARIXC20	ARIXC21				
ARIXC22	ARIXC23	ARIXC24				
ARIXC27	ARIXC28	ARIXC29				
ARIXC30	ARIXC31	ARIXC32				
ARIXC33	ARIXC34	ARIXC36				
ARIXC37	ARIXC38	ARIXC39				
ARIXC40	ARIXC41	ARIXC42				
ARIXC43	ARIXC45	ARIXC47				
ARIXC48	ARIXC51	ARIXEAB				

* Asterisked items are secondary entry points. See page 346. ⁰ indicates CMS EXEC

Calling Modules			Module/ Entry Pt	Called Modules/Entry Points		
ARIXEBR	ARIXECL	ARIXEDP				
ARIXEDR	ARIXEDS	ARIXEFB				
ARIXELX	ARIXEPP	ARIXERD				
ARIXERP	ARIXESX	ARIXEUH				
ARIXI01	ARIXI03	ARIXI04				
ARIXI06	ARIXI07	ARIXI08				
ARIXI09	ARIXI10	ARIXI13				
ARIXI14	ARIXI15	ARIXI18				
ARIXI19	ARIXI20	ARIXI21				
ARIXI22	ARIXI23	ARIXI24				
ARIXI25	ARIXI26	ARIXI27				
ARIXI28	ARIXI29	ARIXI30				
ARIXOAF	ARIXOBY	ARIXOB2				
ARIXOCA	ARIXOCK	ARIXOCT				
ARIXOCU	ARIXOCY	ARIXODF				
ARIXODM	ARIXOD1	ARIXOD2				
ARIXOEX	ARIXOFC	ARIXOFE				
ARIXOFF	ARIXOFP	ARIXOFQ				
ARIXOFR	ARIXOFT	ARIXOGA				
ARIXOGB	ARIXOGC	ARIXOGP				
ARIXOIN	ARIXOLC	ARIXOLF				
ARIXOMA	ARIXOML	ARIXOMS				
ARIXONV	ARIXOOP	ARIXOOS				
ARIXOQ1	ARIXOSC	ARIXOSL				
ARIXOSO	ARIXOSR	ARIXOST				
ARIXOSU	ARIXOS4	ARIXOTF				
ARIXOVC	ARIXOVD	ARIXOV1				
ARIXOW1	ARIXOW2	ARIXO1S				
ARIXPA1	ARIXPA2	ARIXPA3				
	ARIXSRT	ARIXSUT				
ARIXEDP	ARIXEDR	ARIXEDS	ARIXEFB	ARICWSF	ARISYSD2*	ARIXEER
ARIXEPP	ARIXERD	ARIXOVC				
			ARIXELK	ARICMUD	ARICOMB	ARIXEBR
				ARIXEBR1*	ARIXECW1	ARIXERD
				ARIXERD1	ARIXERO	ARIXESP
				ARIXEST	ARIXLNK	ARIXLNK1*
				ARIXPAT		
ARIXERD	ARIXERP	ARIXOVC	ARIXELX	ARICWSF	ARICWSG	ARISYSDI*
				ARIXA02	ARIXEAB	ARIXECK
				ARIXEDB	ARIXEER	ARIXETR
				ARIXSCF	ARIXSCH	
		ARIXERP	ARIXELX1*			
			ARIXELX2*			
			ARIXELX3*			
		ARIXERD	ARIXEOC	ARIXEDB		
		ARIXERD	ARIXEPH			
	ARIXEDS	ARIXERP	ARIXEPP	ARICWSF	ARICWSG	ARISYSDF*
				ARIXCR0	ARIXC01	ARIXC37
				ARIXECK	ARIXEER	ARIXEFB

* Asterisked items are secondary entry points. See page 346. [0] indicates CMS EXEC

Calling Modules			Module/ Entry Pt	Called Modules/Entry Points		
				ARIXESX2*	ARIXETR	ARIXOIN
				ARIXOOP	ARIXPAI	
		ARIXEST	ARIXERA	ARISYSDI*		
		ARIXELK	ARIXERD	ARICDWT	ARICMUD	ARICOMB
				ARICWSF	ARICWSG	ARIXA01
				ARIXA02	ARIXEAB	ARIXEBR
				ARIXECK	ARIXECL	ARIXECW
				ARIXEDB	ARIXEDC	ARIXEDP
				ARIXEDP1*	ARIXEDS	ARIXEER
				ARIXELX	ARIXEOC	ARIXERP
				ARIXI12	ARIXI14	ARIXI16
	ARIYT15 (via DS2XER1P)		ARIXERDL			
		ARIXELK	ARIXERD1*			
				ARIXSCH		
		ARIXELK	ARIXERO	ARICWSF	ARICWSG	ARISYSD1*
		ARIXERD	ARIXERP	ARIMCID	ARIXA06	ARIXEAB
				ARIXEAD	ARIXEAD1*	ARIXEAD2*
				ARIXECK	ARIXECW	ARIXEDB
				ARIXEDR	ARIXEER	ARIXELX
				ARIXELX1*	ARIXEPH	ARIXEPP
				ARIXESX3	ARIXETR	ARIXEUH
				ARIXSCN	ARIXSDT	ARIXSLN
				ARIXSTM		
		ARIXELK	ARIXESP	ARICWSF	ARICWSG	
		ARIXELK	ARIXEST	ARIXERA		
			ARIXESX	ARICWSF	ARICWSG	ARIXECK
				ARIXEDB	ARIXEER	ARIXI101
				ARIXECW	ARIXETR	ARIXSCH
	ARIXEDS	ARIXI09	ARIXESX1*			
	ARIXEPP	ARIXI09	ARIXESX2*			
ARIXEDS	ARIXERP	ARIXI09	ARIXESX3*			
			ARIXESX4*			
			ARIXESX5*			
(Called by many RDS modules via XTRACE macro)			ARIXETR	ARICDWT	ARICPDM	ARISYSD5*
				ARIYM04		
	ARIXERD	ARIXERP	ARIXEUH	ARIXEER		
		ARIXIST	ARIXISH			
		ARIXI19	ARIXIST	ARICWSF	ARICWSG	ARIXECK
				ARIXEDB	ARIXEER	ARIXETR
				ARIXISH		
		ARIXI14	ARIXI01	ARICWSG	ARIXECK	ARIXEDB
				ARIXEER		
		ARIXI14	ARIXI03	ARICWSF	ARICWSG	ARIXA02
				ARIXECK	ARIXEDB	ARIXEER
				ARIXSCH		
		ARIXI14	ARIXI04	ARICWSG	ARIXECK	ARIXEDB
				ARIXEER		
		ARIXI14	ARIXI06	ARICWSG	ARIXECK	ARIXEDB
				ARIXEER	ARIXI20	ARIXSRT

* Asterisked items are secondary entry points. See page 346. [0] indicates CMS EXEC

Calling Modules		Module/ Entry Pt	Called Modules/Entry Points		
	ARIXI14	ARIXI07	ARICWSF	ARICWSG	ARIXA02
			ARIXA06	ARIXECK	ARIXEDB
			ARIXEER	ARIXSCH	ARIXSDB
			ARIXSRT		
	ARIXI14	ARIXI08	ARICWSG	ARIXECK	ARIXEDB
			ARIXEER	ARIXETR	
	ARIXI14	ARIXI09	ARICWSG	ARIXA02	ARIXA06
			ARIXECK	ARIXEDB	ARIXEER
			ARIXESX1*	ARIXESX2*	ARIXESX3*
			ARIXSCH	ARIXSDB	ARIXSDT
			ARIXSLN	ARIXSRT	ARIXSTM
	ARIXI14	ARIXI10	ARINCID	ARIXEER	ARIXI22
			ARIXI23	ARIXI24	ARIXI25
			ARIXI26	ARIXI30	ARIXSCH
	ARIXESX	ARIXI101			
ARIXA08	ARIXA09	ARIXI102*			
ARIXERD	ARIXI14	ARIXI12	ARIXECK	ARIXECW	ARIXEDB
			ARIXI18		
	ARIXI14	ARIXI13	ARICWSG	ARIXA06	ARIXECK
			ARIXEDB	ARIXEER	ARIXSDB
			ARIXSRT		
	ARIXERD	ARIXI14	ARIXA07	ARIXA09	ARIXEER
			ARIXI01	ARIXI03	ARIXI04
			ARIXI06	ARIXI07	ARIXI08
			ARIXI09	ARIXI10	ARIXI12
			ARIXI13	ARIXI15	ARIXI16
			ARIXI19		
	ARIXI14	ARIXI15	ARICWSF	ARICWSG	ARIXA02
			ARIXECK	ARIXEDB	ARIXEER
ARIXERD	ARIXI14	ARIXI16	ARIXECK	ARIXECW	ARIXEDB
ARIXEDS	ARIXI12	ARIXI18	ARICWSG	ARIXECK	ARIXEDB
			ARIXEER	ARIXSDB	
	ARIXI33	ARIXI19	ARIXECK	ARIXEDB	ARIXEER
			ARIXETR	ARIXIST	ARIXI20
ARIXI06	ARIXI19	ARIXI20	ARIXEDB	ARIXECK	ARIXEER
			ARIXETR		
		ARIXI21	ARIXECK	ARIXEDB	ARIXEER
			ARIXSXD		
	ARIXI10	ARIXI22	ARIXECK	ARIXEDB	ARIXEER
			ARIXI27	ARIXI29	ARIXSRT
	ARIXI10	ARIXI23	ARICWSG	ARIXECK	ARIXEDB
			ARIXEER	ARIXI27	ARIXI29
			ARIXI32		
ARIXI10	ARIXI29	ARIXI24	ARICWSG	ARIXECK	ARIXEDB
			ARIXEER	ARIXI27	ARIXI29
			ARIXI32		
	ARIXI10	ARIXI25	ARICWSF	ARICWSG	ARIXA02
			ARIXECK	ARIXEDB	ARIXEER
			ARIXI27	ARIXI29	ARIXI32

* Asterisked items are secondary entry points. See page 346. ° indicates CMS EXEC

Calling Modules			Module/ Entry Pt	Called Modules/Entry Points		
				ARIXSDB		
		ARIXI10	ARIXI26	ARIXECK	ARIXEDB	ARIXEER
ARIXI22	ARIXI23	ARIXI24	ARIXI27	ARICWSG	ARIXECK	ARIXEDB
	ARIXI25	ARIXI30		ARIXEER		
		ARIXI32	ARIXI28	ARICWSG	ARIXECK	ARIXEDB
				ARIXEER		
ARIXI22	ARIXI23	ARIXI24	ARIXI29	ARIXECK	ARIXEDB	ARIXEER
		ARIXI25		ARIXI24	ARIXI30	ARIXI31
	ARIXI10	ARIXI29	ARIXI30	ARIXA04	ARIXECK	ARIXEDB
				ARIXEER	ARIXI27	ARIXI31
	ARIXI29	ARIXI30	ARIXI31	ARIXECK	ARIXEDB	
ARIXI23	ARIXI24	ARIXI25	ARIXI32	ARIXA04	ARIXECK	ARIXEDB
				ARIXI28	ARIXSRT	
		ARIXI14	ARIXI33	ARICWSF	ARICWSG	ARIXECK
				ARIXEDB	ARIXEER	ARIXOCK
				ARIXOFE	ARIXSRT	ARIXI19
		ARIXELK	ARIXLNK			
		ARIXELK	ARIXLNK1			
	ARIXOCU	ARIXOOP	ARIXOAF	ARICWSF	ARICWSG	ARIXEER
				ARIXSCH		
ARIXOB2	ARIXOCT	ARIXODM	ARIXOBY	ARIXEAB	ARIXEER	
ARIXOEX	ARIXOFR	ARIXOGA				
ARIXOGP	ARIXOLF	ARIXOMA				
ARIXOMD	ARIXOML	ARIXOMS				
ARIXOOP	ARIXOOS	ARIXOSR				
	ARIXOTF	ARIXO1S				
	ARIXOQ1	ARIXOW1	ARIXOB1	ARIXOFQ	ARIXOFT	ARIXOSL
				ARIXOW1		
	ARIXOOP	ARIXOQ2	ARIXOB2	ARIXEER	ARIXOBY	ARIXOSL
				ARIXOW2		
ARIXOFE	ARIXOIN	ARIXOOP	ARIXOCA	ARIXECK	ARIXEDB	ARIXEER
ARIXOSO	ARIXOVC	ARIXOVD		ARIXOIU	ARIXSCN	
ARIXOIN	ARIXOOP	ARIXOSO	ARIXOCK	ARICWSG	ARIXECK	ARIXEDB
	ARIXOVC	ARIXOVD		ARIXEER	ARIXSCN	
		ARIXOTS	ARIXOCS	ARIXOGC	ARIXOSS	
ARIXOGA	ARIXOIN	ARIXOMF	ARIXOCT	ARIXEER	ARIXOBY	
	ARIXOMS	ARIXOOP				
ARIXODF	ARIXOD1	ARIXOD2	ARIXOCU	ARICWSF	ARIXEER	ARIXOAF
ARIXOFP	ARIXOFQ	ARIXOGC		ARIXSCH		
ARIXOML	ARIXOSR	ARIXOVC				
		ARIXSO3				
		ARIXOGA	ARIXOCY	ARIXEER		
		ARIXOS4	ARIXOC4			
	ARIXODF	ARIXOGC	ARIXODF	ARIMFLP2*	ARIXEER	ARIXOCU
				ARIXODF	ARIXSCH	ARIXSCN
ARIXOIN	ARIXOMF	ARIXOMS	ARIXODM	ARIXEER	ARIXOBY	ARIXSCH
		ARIXOOP				
	ARIXOGP	ARIXOOP	ARIXOD1	ARICWSF	ARICWSG	ARIXECK
				ARIXEDB	ARIXEER	ARIXETR

* Asterisked items are secondary entry points. See page 346. [0] indicates CMS EXEC

Calling Modules			Module/ Entry Pt	Called Modules/Entry Points		
				ARIXOCU	ARIXOD2	
		ARIXOD1	ARIXOD2	ARICWSG	ARIXWSF	ARIXECK
				ARIXEDB	ARIXEER	ARIXETR
				ARIXOCU		
ARIXOEX	ARIXOVC	ARIXOW1	ARIXOEX	ARIXEER	ARIXOBY	ARIXOEX
		ARIXOW2		ARIXOFC	ARIXOTF	
ARIXOEX	ARIXOLF	ARIXOML	ARIXOFC	ARIXEER		
ARIXONV	ARIXOOS	ARIXOST				
	ARIXI19	ARIXOOP	ARIXOFE	ARIXEER	ARIXOCA	ARIXOUS
	ARIXOIN	ARIXOST	ARIXOFF	ARIXEER	ARIXOTF	ARIXSCH
		ARIXOW2	ARIXOFP	ARIXEER	ARIXOCU	
	ARIXOB1	ARIXOOP	ARIXOFQ	ARIXEER	ARIXOCU	
ARIXOIN	ARIXOMF	ARIXOMS	ARIXOFR	ARIXEER	ARIXOBY	
		ARIXOOP				
ARIXOB1	ARIXOOP	ARIXOVC	ARIXOFT	ARIXEER	ARIXOUS	
		ARIXOGP	ARIXOGA	ARIXEER	ARIXOBY	ARIXOCT
				ARIXOCY	ARIXOML	ARIXOMS
				ARIXOSR	ARIXOTF	ARIXSCH
ARIXOIN	ARIXOOP	ARIXOSO	ARIXOGB	ARIXEER		
	ARIXOCS	ARIXOTS	ARIXOGC	ARIXEER	ARIXOCU	ARIXODF
				ARIXOSS		
ARIXOIN	ARIXOOP	ARIXOSO	ARIXOGP	ARICWSF	ARICWSG	ARIXEER
				ARIXOBY	ARIXOGA	ARIXOTS
				ARIXSCH		
ARIXEPP	ARIXOML	ARIXOOP	ARIXOIN	ARIXA02	ARIXA06	ARIXEER
				ARIXOCA	ARIXOCK	ARIXOCT
				ARIXOOM	ARIXOFF	ARIXOFR
				ARIXOGB	ARIXOGP	ARIXOLF
				ARIXONV	ARIXOQ1	ARIXOQ2
				ARIXOSC	ARIXOSL	ARIXOVC
				ARIXSCH	ARIXSCN	
	ARIXOCA	ARIXOMS	ARIXOIU	ARIXECK	ARIXEDB	ARIXSUT
	ARIXOMA	ARIXOSR	ARIXOLC	ARIXEER		
ARIXOIN	ARIXOOP	ARIXOSO	ARIXOLF	ARIXEER	ARIXOBY	ARIXOFC
	ARIXOML	ARIXOSR	ARIXOMA	ARIXEER	ARIXOBY	ARIXOLC
	ARIXOMS	ARIXOIS	ARIXOMB	ARIXOTF		
	ARIXOOP	ARIXOSO	ARIXOMD	ARIXOBY	ARIXSCH	
		ARIXOOP	ARIXOMF	ARIXOCT	ARIXODM	ARIXOFR
		ARIXOGA	ARIXOML	ARICWSF	ARICWSG	ARIXEER
				ARIXOBY	ARIXOCU	ARIXOFC
				ARIXOIN	ARIXOMA	ARIXOMS
				ARIXSCH		
ARIXOGA	ARIXOML	ARIXOSR	ARIXOMS	ARIXEER	ARIXOBY	ARIXOCT
				ARIXODM	ARIXOFR	ARIXOIU
				ARIXOMB	ARIXORM	ARIXOSG
				ARIXOTF	ARIXSCH	
ARIXOIN	ARIXOOP	ARIXOSO	ARIXONV	ARIXEER	ARIXOFC	ARIXSCH
		ARIXOVD				
ARIXOOP	ARIXOSO	ARIXOVD	ARIXONN	ARIXEAB		

* Asterisked items are secondary entry points. See page 346. ⁰ indicates CMS EXEC

Calling Modules			Module/ Entry Pt	Called Modules/Entry Points		
	ARIXEDP	ARIXEPP	ARIXOOP	ARICWSF	ARICWSG	ARISYSDF*
				ARIXA02	ARIXA06	ARIXEER
				ARIXOAF	ARIXOBY	ARIXOB2
				ARIXOCA	ARIXOCK	ARIXOCT
			ARIXOOP	ARIXODM	ARIXOFE	ARIXOFQ
				ARIXOFR	ARIXOFT	ARIXOGB
				ARIXOGP	ARIXOIN	ARIXOLF
				ARIXOMD	ARIXOMF	ARIXONV
				ARIXONW	ARIXOQ1	ARIXOQ2
				ARIXOSC	ARIXOSL	ARIXOSO
				ARIXOST	ARIXOSU	ARIXOVC
				ARIXOVD	ARIXOW1	ARIXSCH
				ARIXSCN	ARIXSLN	
ARIXOOS	ARIXOSL	ARIXOST	ARIXOOS	ARIXEER	ARIXOBY	ARIXOFC
		ARIXOVC		ARIXOOS	ARIXOTF	
ARIXOIN	ARIXOOP	ARIXOSO	ARIXOQ1	ARIXEER	ARIXOB1	ARIXOSL
		ARIXOVD				
ARIXOIN	ARIXOOP	ARIXOSO	ARIXOQ2	ARIXOB2		
	ARIXOMS	ARIXOSG	ARIXORM			
ARIXOIN	ARIXOOP	ARIXOSO	ARIXOSC	ARIXA02	ARIXA06	ARIXEER
				ARIXSCN		
		ARIXOMS	ARIXOSG	ARIXORM	ARIXOS1	ARIXOS2
				ARIXOS3	ARIXOIS	
ARIXOB1	ARIXOB2	ARIXOIN	ARIXOSL	ARIXEER	ARIXOOS	ARIXSCH
ARIXOOP	ARIXOQ1	ARIXOVC				
		ARIXOVD				
	ARIXOOP	ARIXOSO	ARIXOSO	ARIXEER	ARIXOCA	ARIXOCK
				ARIXOGB	ARIXOGP	ARIXOLF
				ARIXOMD	ARIXONV	ARIXONW
				ARIXOQ1	ARIXOQ2	ARIXOSC
				ARIXOSO	ARIXOVC	ARIXSCH
		ARIXOGA	ARIXOSR	ARICWSF	ARICWSG	ARIXEER
				ARIXOBY	ARIXOCU	ARIXOLC
				ARIXOMA	ARIXOMS	ARIXSCH
ARIXOCS	ARIXOGC	ARIXOTS	ARIXOSS			
		ARIXOOP	ARIXOST	ARIXEER	ARIXOFC	ARIXOFF
				ARIXOOS		
		ARIXOOP	ARIXOSU	ARICWSG	ARIXEER	ARIXSCH
ARIXOSG	ARIXOS1	ARIXOS3	ARIXOS1	ARIXOS1		
		ARIXOS4				
	ARIXOSG	ARIXOS2	ARIXOS2	ARIXOS2	ARIXOS4	
	ARIXOSG	ARIXOS3	ARIXOS3	ARIXEER	ARIXOCU	ARIXOS1
				ARIXOS3		
		ARIXOS2	ARIXOS4	ARIXEER	ARIXOC4	ARICOS1
				ARIXOS4		
ARIXOEX	ARIXOFF	ARIXOGA	ARIXOTF	ARIMFLP2*	ARIXEER	ARIXOBY
ARIXOMB	ARIXOMS	ARIXOOS		ARIXSCH		
		ARIXOW2				
		ARIXOGP	ARIXOTS	ARIXOCS	ARIXOGC	ARIXOSS

* Asterisked items are secondary entry points. See page 346. [0] indicates CMS EXEC

Calling Modules			Module/ Entry Pt	Called Modules/Entry Points		
	ARIXOFE	ARIXOFT	ARIXOUS	ARIXECK	ARIXEOB	
ARIXOIN	ARIXOOP	ARIXOSO	ARIXOVC	ARICWSF	ARICWSG	ARIXEER
		ARIXOVD		ARIXEFB	ARIXELX	ARIXOCA
				ARIXOCK	ARIXOCU	ARIXOEX
				ARIXOFT	ARIXOOS	ARIXOSL
				ARIXOV1	ARIXOW1	ARIXSCH
		ARIXOOP	ARIXOVD	ARICWSF	ARICWSG	ARIXA02
				ARIXEER	ARIXOCA	ARIXOCK
				ARIXONV	ARIXONW	ARIXOQ1
				ARIXOSL	ARIXOVC	ARIXSCH
				ARIXSCN	ARIXSLN	
	ARIXOVC	ARIXOV1	ARIXOV1	ARIXEER	ARIXOV1	
ARIXOB1	ARIXOOP	ARIXOVC	ARIXOW1	ARIXEER	ARIXOB1	ARIXOEX
		ARIXOW1		ARIXOWI	ARIXSCH	
	ARIXOB2	ARIXOW2	ARIXOW2	ARIXEER	ARIXOEX	ARIXOFP
				ARIXOTF	ARIXOW2	ARIXSCH
		ARIXOSG	ARIXO1S	ARIXEER	ARIXOBY	ARIXOMB
				ARIXSCH		
		ARIXELK	ARIXPAT			
		ARIXPA1	ARIXPA0	ARIXPA2	ARIXPA3	
	ARIXEDP	ARIXEPP	ARIXPA1	ARICWSF	ARICWSG	ARIXEER
				ARIXPA0	ARIXPA4	ARIYE11
		ARIXPA0	ARIXPA2	ARIXEER		
		ARIXPA0	ARIXPA3	ARIXEER		
		ARIXPA1	ARIXPA4			
		ARIMTRA	ARIXPTT	ARIMFNT	ARIMPRT	ARIMPT1
				ARISYSD1*	ARISYSD2*	ARISYSD7*
ARIXA02	ARIXC18	ARIXEDP	ARIXSCF			
	ARIXELX	ARIXSRT				
ARIXC02	ARIXC03	ARIXC05	ARIXSCH			
ARIXC06	ARIXC18	ARIXC20				
ARIXC21	ARIXC24	ARIXC28				
ARIXC30	ARIXC33	ARIXC34				
ARIXC36	ARIXC39	ARIXC48				
ARIXC51	ARIXEAB	ARIXEDR				
ARIXEDP	ARIXELX	ARIXERD				
ARIXESX	ARIXI03	ARIXI07				
ARIXI09	ARIXI10	ARIXOAF				
ARIXOCU	ARIXODF	ARIXODM				
ARIXOFF	ARIXOGA	ARIXOGP				
ARIXOIN	ARIXOMD	ARIXOML				
ARIXOMS	ARIXONV	ARIXOOP				
ARIXOSL	ARIXOSO	ARIXOSR				
ARIXOSU	ARIXOTF	ARIXOVC				
ARIXOVD	ARIXOW1	ARIXOW2				
	ARIXO1S	ARIXSDB				
ARIXEDR	ARIXERD	ARIXOCA	ARIXSCN			
ARIXOCK	ARIXODF	ARIXOIN				

* Asterisked items are secondary entry points. See page 346. [0] indicates CMS EXEC

Calling Modules			Module/ Entry Pt	Called Modules/Entry Points		
ARIXOOP	ARIXOSC	ARIXOVD				
ARIXI07	ARIXI09	ARIXI13	ARIXSDB	ARIXSCH		
	ARIXI18	ARIXI25				
	ARIXERP	ARIXI09	ARIXSDT			
ARIXA01	ARIXA02	ARIXA04	ARIXSLN			
ARIXA06	ARIXA07	ARIXA08				
ARIXA09	ARIXA10	ARIXA11				
ARIXEDR	ARIXERD	ARIXI09				
ARIXOOP	ARIXOVD	ARIXSRT				
ARIXI06	ARIXI07	ARIXI09	ARIXSRT	ARIXECW	ARIXEDB	ARIXEER
ARIXI13	ARIXI22	ARIXI32		ARIXSCF	ARIXSLN	
	ARIXERP	ARIXI09	ARIXSTM			
	ARIXA06	ARIXOIU	ARIXSUT	ARIXEER		
		ARIXI21	ARIXSXD			
			ARIYC00	ARIYC06	ARIYC07	ARIYD26
				ARIYD38	ARIYD40	ARIYI422*
			ARIYC01	ARIYC06	ARIYC07	ARIYD26
				ARIYD38	ARIYD40	ARIYI422*
			ARIYC02	ARIYC03	ARIYC031*	ARIYC05
				ARIYC07	ARIYC09	ARIYC15
				ARIYD26	ARIYD40	ARIYD63
				ARIYI14	ARIYI19	ARIYI421*
				ARIYK40	ARIYM02	ARIYX04
ARIYC02	ARIYC05	ARIYC10	ARIYC03	ARIYC09	ARIYD14	ARIYI19
				ARIYM02		
ARIYC02	ARIYC05	ARIYC10	ARIYC031*			
		ARIYC02	ARIYC05	ARIYC03	ARIYC031*	ARIYC15
				ARIYC151*	ARIYD26	ARIYD40
				ARIYD44	ARIYD63	ARIYI14
				ARIYI42	ARIYK40	ARIYM02
ARIYC00	ARIYC01	ARIYC11	ARIYC06	ARIYI46	ARIYK09	ARIYM02
ARIYC00	ARIYC01	ARIYC02	ARIYC07	ARIYK08	ARIYM02	
	ARIYC10	ARIYC11				
			ARIYC08			
ARIYC02	ARIYC03	ARIYK40	ARIYC09			
			ARIYC10	ARIYC03	ARIYC031*	ARIYC07
				ARIYC15	ARIYD26	ARIYD40
				ARIYD63	ARIYK40	
			ARIYC11	ARICWSF	ARICWSG	ARIYC06
			ARIYC11	ARIYC07	ARIYC111*	ARIYC13
				ARIYC14	ARIYC15	ARIYC151*
				ARIYD26	ARIYD40	ARIYD44
				ARIYD48	ARIYI10	ARIYK39
				ARIYK40	ARIYM02	ARIYT141*
		ARIYC11	ARIYC111*			
		ARIYC11	ARIYC13	ARIYC15	ARIYC151*	ARIYD26
				ARIYD48	ARIYM02	
		ARIYC11	ARIYC14	ARIYC15	ARIYD26	ARIYD40

* Asterisked items are secondary entry points. See page 346. ⁰ indicates CMS EXEC

Calling Modules			Module/ Entry Pt	Called Modules/Entry Points		
				ARIYD44	ARIYX02	
ARIYC02	ARIYC05	ARIYC10	ARIYC15	ARIYD63	ARIYI46	ARIYM02
ARIYC11	ARIYC13	ARIYC14				
ARIYC05	ARIYC11	ARIYC13	ARIYC151*			
	ARIYC23	ARIYC24				
		ARIYL07	ARIYC18	ARIYD52	ARIYD74	ARIYD76
				ARIYD77		
		ARIYL07	ARIYC19	ARIYD52	ARIYD74	ARIYD76
				ARIYD77		
		ARIYL07	ARIYC20	ARIYD52	ARIYD74	ARIYD76
				ARIYD77	ARIYX02	ARIYX04
		ARIYL07	ARIYC21	ARIYD52	ARIYD74	ARIYD76
				ARIYD77		
		ARIYL07	ARIYC22	ARIYD52	ARIYD74	ARIYD76
				ARIYD77		
		ARIYL07	ARIYC23	ARIYC151*	ARIYC26	ARIYD52
				ARIYD74	ARIYD76	ARIYI10
				ARIYM02		
		ARIYL07	ARIYC24	ARIYC151*	ARIYC25	ARIYC26
				ARIYD48	ARIYD52	ARIYD74
				ARIYD76	ARIYD77	
	ARIYC24	ARIYL07	ARIYC25	ARIYD52	ARIYD74	ARIYD76
				ARIYD77	ARIYX02	ARIYX04
ARIYC23	ARIYC24	ARIYL07	ARIYC26	ARIYD48	ARIYD52	ARIYD74
				ARIYD76	ARIYD77	ARIYL05
			ARIYD00	ARIYK00		
			ARIYD01	ARIYD15	ARIYD32	ARIYD33
				ARIYD38	ARIYD40	ARIYD43
				ARIYD55	ARIYD551*	ARIYD60
				ARIYD64	ARIYI46	ARIYM02
			ARIYD02	ARIYD17	ARIYD18	ARIYD23
				ARIYD28	ARIYD32	ARIYD38
				ARIYD40	ARIYD43	ARIYD46
				ARIYD55	ARIYD551*	ARIYD59
				ARIYD64	ARIYI14	ARIYM02
				ARIYX01	ARIYX18	ARIYX22
				ARIYX28		
			ARIYD03	ARIYD18	ARIYD32	ARIYD38
				ARIYD40	ARIYD43	ARIYD55
				ARIYD551*	ARIYD59	ARIYD64
			ARIYD04	ARIYD11	ARIYD13	ARIYD23
				ARIYD32	ARIYD38	ARIYD41
				ARIYD43	ARIYD45	ARIYD46
				ARIYD49	ARIYD55	ARIYD551*
				ARIYD57	ARIYD58	ARIYD59
				ARIYD62	ARIYM02	
			ARIYD05	ARIYD15	ARIYD23	ARIYD29
				ARIYD32	ARIYD38	ARIYD40
				ARIYD43	ARIYD55	ARIYD551*

* Asterisked items are secondary entry points. See page 346. ° indicates CMS EXEC

Calling Modules			Module/ Entry Pt	Called Modules/Entry Points		
				ARIYD60	ARIYD63	ARIYI14
				ARIYI19	ARIYI46	ARIYM02
				ARIYS03	ARIYS04	ARIYX01
				ARIYX18	ARIYX21	ARIYX22
				ARIYX28		
			ARIYD06	ARIYD11	ARIYD13	ARIYD20
				ARIYD23	ARIYD232*	ARIYD27
				ARIYD32	ARIYD38	ARIYD41
				ARIYD411*	ARIYD43	ARIYD45
				ARIYD49	ARIYD53	ARIYD531*
				ARIYD55	ARIYD551*	ARIYD57
				ARIYD58	ARIYD62	ARIYI19
				ARIYI46	ARIYK00	ARIYK41
				ARIYM02	ARIYS01	ARIYS05
				ARIYX05	ARIYX20	ARIYX23
			ARIYD08	ARIYD13	ARIYD23	ARIYD28
				ARIYD32	ARIYD34	ARIYD38
				ARIYD40	ARIYD41	ARIYD411*
				ARIYD43	ARIYD45	ARIYD46
				ARIYD49	ARIYD50	ARIYD53
				ARIYD531*	ARIYD55	ARIYD551*
				ARIYD57	ARIYD58	ARIYD59
				ARIYD60	ARIYD62	ARIYI19
				ARIYI46	ARIYK00	ARIYK41
				ARIYM02	ARIYS01	ARIYS05
				ARIYX20	ARIYX21	ARIYX23
			ARIYD09	ARIYD13	ARIYD23	ARIYD32
				ARIYD38	ARIYD40	ARIYD41
				ARIYD43	ARIYD49	ARIYD53
				ARIYD531*	ARIYD55	ARIYD551*
				ARIYD59	ARIYD62	
			ARIYD10	ARIYD11	ARIYD23	ARIYD28
				ARIYD30	ARIYD32	ARIYD33
				ARIYD38	ARIYD43	ARIYD46
				ARIYD55	ARIYD551*	ARIYD59
				ARIYD64	ARIYM02	ARIYX01
				ARIYX18	ARIYX22	ARIYX28
ARIYD04	ARIYD06	ARIYD10	ARIYD11	ARIYD53	ARIYD531*	ARIYI46
		ARIYS00		ARIYK09	ARIYM02	ARIYT19
				ARIYT22		
ARIYI36	ARIYK01	ARIYK28	ARIYD111*			
ARIYD04	ARIYD06	ARIYD08	ARIYD13			
		ARIYD09				
ARIYC03	ARIYD33	ARIYD39	ARIYD14	ARIYK03		
	ARIYD01	ARIYD05	ARIYD15	ARIYD32	ARIYD33	ARIYD38
				ARIYD53	ARIYD531*	ARIYD64
				ARIYI46	ARIYM02	
	ARIYD77	ARIYS02	ARIYD16			
		ARIYD02	ARIYD17			

* Asterisked items are secondary entry points. See page 346. ⁰ indicates CMS EXEC

Calling Modules			Module/ Entry Pt	Called Modules/Entry Points		
	ARIYD02	ARIYD03	ARIYD18	ARIYD32	ARIYD38	ARIYD53
				ARIYD53I*	ARIYD64	ARIYM02
ARIYD39	ARIYD40	ARIYD77	ARIYD19	ARIYI14	ARIYM02	
	ARIYD06	ARIYD50	ARIYD20	ARIYD26	ARIYD58	ARIYI46
				ARIYK00	ARIYK41	ARIYM02
		ARIYD50	ARIYD201*			
ARIYD02	ARIYD04	ARIYD05	ARIYD23	ARIYI19	ARIYM02	
ARIYD06	ARIYD08	ARIYD09				
ARIYD10	ARIYD69	ARIYD74				
ARIYD75	ARIYZ05	ARIYZ16				
	ARIYD62	ARIYD64	ARIYD231*			
ARIYD06	ARIYD62	ARIYD64	ARIYD232*			
ARIYC00	ARIYC01	ARIYC02	ARIYD26	ARIYI19	ARIYM02	
ARIYC05	ARIYC10	ARIYC11				
ARIYC13	ARIYC14	ARIYD20				
ARIYK39	ARIYS00	ARIYS24				
ARIYD06	ARIYD46	ARIYS00	ARIYD27			
ARIYD02	ARIYD08	ARIYD10	ARIYD28			
	ARIYD69	ARIYD75				
		ARIYD05	ARIYD29			
		ARIYD10	ARIYD30			
ARIYD01	ARIYD02	ARIYD03	ARIYD32	ARIYD40	ARIYK41	ARIYM02
ARIYD04	ARIYD05	ARIYD06				
ARIYD08	ARIYD09	ARIYD10				
ARIYD15	ARIYD18	ARIYS00				
ARIYD01	ARIYD10	ARIYD15	ARIYD33	ARIYD14	ARIYD39	ARIYD63
				ARIYI46	ARIYK41	ARIYM02
ARIYD08	ARIYX04	ARIYX23	ARIYD34	ARIYD61		
ARIYC00	ARIYC01	ARIYD01	ARIYD38	ARIYI19	ARIYI42	ARIYK41
ARIYD02	ARIYD03	ARIYD04		ARIYM02		
ARIYD05	ARIYD06	ARIYD08				
ARIYD09	ARIYD10	ARIYD15				
	ARIYD18	ARIYS00				
		ARIYD33	ARIYD39	ARIYD14	ARIYD19	ARIYI14
				ARIYI19	ARIYI46	ARIYK00
				ARIYK41	ARIYM02	
ARIYC00	ARIYC01	ARIYC02	ARIYD40	ARIYD19	ARIYI19	ARIYM02
ARIYC05	ARIYC10	ARIYC11	ARIYD40			
ARIYC14	ARIYD01	ARIYD02				
ARIYD03	ARIYD05	ARIYD08				
ARIYD09	ARIYD32	ARIYD46				
ARIYD62	ARIYD64	ARIYZ03				
ARIYD04	ARIYD06	ARIYD08	ARIYD41	ARIYK41	ARIYK411*	ARIYM02
	ARIYD09	ARIYS24				
ARIYD06	ARIYD08	ARIYX03	ARIYD411*			
ARIYD01	ARIYD02	ARIYD03	ARIYD43	ARIYK08	ARIYM02	
ARIYD04	ARIYD05	ARIYD06				
ARIYD08	ARIYD09	ARIYD10				
ARIYC05	ARIYC11	ARIYC14	ARIYD44			

* Asterisked items are secondary entry points. See page 346. [0] indicates CMS EXEC

Calling Modules			Module/ Entry Pt	Called Modules/Entry Points		
	ARIYI00	ARIYI36				
ARIYD04	ARIYD06	ARIYD08	ARIYD45			
ARIYD02	ARIYD04	ARIYD08	ARIYD46	ARIYD27	ARIYD40	ARIYM02
	ARIYD10	ARIYZ05		ARIYX21		
	ARIYD62	ARIYD64	ARIYD47	ARIYK41	ARIYM02	
ARIYC11	ARIYC13	ARIYC24	ARIYD48	ARIYD531*	ARIYD63	ARIYI14
		ARIYC26		ARIYI19	ARIYI46	ARIYM02
ARIYD04	ARIYD06	ARIYD08	ARIYD49			
		ARIYD09				
ARIYD08	ARIYX04	ARIYZ11	ARIYD50	ARIYD20	ARIYD201*	ARIYI14
				ARIYI19	ARIYI46	ARIYK00
				ARIYK41	ARIYM02	
ARIYC18	ARIYC19	ARIYC20	ARIYD52			
ARIYC21	ARIYC22	ARIYC23				
ARIYC24	ARIYC25	ARIYC26				
ARIYD66	ARIYD68	ARIYS00				
ARIYD06	ARIYD08	ARIYD09	ARIYD53			
ARIYD11	ARIYD15	ARIYD18				
	ARIYD69	ARIYZ10				
ARIYD06	ARIYD08	ARIYD09	ARIYD531*			
ARIYD11	ARIYD15	ARIYD18				
ARIYD48	ARIYX04	ARIYZ10				
		ARIYZ11				
ARIYD01	ARIYD02	ARIYD03	ARIYD55	ARIYK09	ARIYM02	
ARIYD04	ARIYD05	ARIYD06				
ARIYD08	ARIYD09	ARIYD10				
		ARIYS00				
ARIYD01	ARIYD02	ARIYD03	ARIYD551*			
ARIYD04	ARIYD05	ARIYD06				
ARIYD08	ARIYD09	ARIYD10				
ARIYD04	ARIYD06	ARIYD08	ARIYD57			
		ARIYS00				
ARIYD04	ARIYD06	ARIYD08	ARIYD58	ARIYM03		
	ARIYD20	ARIYS24				
ARIYD02	ARIYD03	ARIYD04	ARIYD59			
ARIYD08	ARIYD09	ARIYD10				
ARIYD01	ARIYD05	ARIYD08	ARIYD60			
ARIYD34	ARIYX00	ARIYX19	ARIYD61			
ARIYD04	ARIYD06	ARIYD08	ARIYD62	ARIYD231*	ARIYD232*	ARIYD40
	ARIYD09	ARIYS00		ARIYD47	ARIYM02	
ARIYC02	ARIYC05	ARIYC10	ARIYD63	ARIYI14	ARIYM02	
ARIYC15	ARIYD05	ARIYD33				
ARIYD48	ARIYD69	ARIYD73				
		ARIYD77				
ARIYD01	ARIYD02	ARIYD03	ARIYD64	ARIYD231*	ARIYD232*	ARIYD40
ARIYD10	ARIYD15	ARIYD18		ARIYD47	ARIYM02	
		ARIYL07	ARIYD65	ARIYD71	ARIYD73	ARIYD74
				ARIYD76	ARIYD77	
		ARIYL07	ARIYD66	ARIYD52	ARIYD72	ARIYD74

* Asterisked items are secondary entry points. See page 346. ° indicates CMS EXEC

Calling Modules			Module/ Entry Pt	Called Modules/Entry Points		
				ARIYD75	ARIYD76	ARIYD77
		ARIYL07	ARIYD67	ARIYD72	ARIYD74	ARIYD76
				ARIYD77		
		ARIYL07	ARIYD68	ARIYD52	ARIYD71	ARIYD74
				ARIYD75	ARIYD76	ARIYD77
		ARIYL07	ARIYD69	ARICWSF	ARICWSG	ARIYD23
				ARIYD28	ARIYD53	ARIYD63
				ARIYD70	ARIYD74	ARIYD76
				ARIYD77	ARIYM02	ARIYX01
				ARIYX18		
ARIYD69	ARIYD73	ARIYD77	ARIYD70	ARIYK03		
	ARIYD65	ARIYD68	ARIYD71	ARIYD73	ARIYD77	
	ARIYD66	ARIYD67	ARIYD72	ARIYD77	ARIYM02	
	ARIYD65	ARIYD71	ARIYD73	ARIYD63	ARIYD70	ARIYD77
				ARIYM02		
ARIYC18	ARIYC19	ARIYC20	ARIYD74	ARIYD23	ARIYK41	ARIYM02
ARIYC21	ARIYC22	ARIYC23				
ARIYC24	ARIYC25	ARIYC26				
ARIYD65	ARIYD66	ARIYD67				
	ARIYD68	ARIYD69				
	ARIYD66	ARIYD68	ARIYD75	ARIYD23	ARIYD28	ARIYX01
				ARIYX18		
ARIYC18	ARIYC19	ARIYC20	ARIYD76			
ARIYC21	ARIYC22	ARIYC23				
ARIYC24	ARIYC25	ARIYC26				
ARIYD65	ARIYD66	ARIYD67				
	ARIYD68	ARIYD69				
ARIYC18	ARIYC19	ARIYC20	ARIYD77	ARIYD16	ARIYD19	ARIYD63
ARIYC21	ARIYC22	ARIYC24		ARIYD70	ARIYI19	ARIYK41
ARIYC25	ARIYC26	ARIYD65		ARIYM02		
ARIYD66	ARIYD67	ARIYD68				
ARIYD69	ARIYD71	ARIYD72				
		ARIYD73				
			ARIYD81	ARICWSF	ARICWSG	ARIYM02
	ARIYT09	ARIYT10	ARIYD87			
	ARIYT09	ARIYT10	ARIYD871*			
		ARIYT21	ARIYD872*			
		ARIYM00	ARIYE01	ARIS251*	ARIYC08	ARIYC081*
				ARIYD81	ARIYD811*	ARIYD812*
				ARIYD813*	ARIYI03	ARIYI031*
				ARIYS25	ARIYT08	ARIYZ27
				(ARIYS251 through ARIYZ27 via computed calls)		
				ARIYE14 (via TRACE macro)		
		ARIYM00	ARIYE011*			
(Called by many DBSS modules via SHOW and YSHOW macros)			ARIYE02	ARIYE05	ARIYE13	
ARICTRC	ARISDSK	ARISEGA	ARIYE04			
ARISPFM	ARIYL21	ARIYL23				
		ARIYM10				
ARIYE02	ARIYI09	ARIYI28	ARIYE05			

* Asterisked items are secondary entry points. See page 346. ⁰ indicates CMS EXEC

Calling Modules			Module/ Entry Pt	Called Modules/Entry Points		
ARIYK37	ARIYZ25	ARIYZ26				
ARICTRC	ARICTRI	ARIYI04	ARIYE06			
	ARIYL21	ARIYL23				
ARISFMID	ARISIIO	ARISSCB	ARIYE07			
ARIYI02	ARIYI09	ARIYI15				
ARIYI23	ARIYI28	ARIYK37				
ARIYL15	ARIYM05	ARIYT12				
ARIYT26	ARIYZ19	ARIYZ25				
		ARIYZ26				
ARICTRC	ARIRDIS(D	ARIRENAD	ARIYE08			
ARISDBR	ARISDSK	ARISEGA				
ARISPFM	ARIYM05	ARIYT18				
		ARIYI04	ARIYE09			
ARIYI28	ARIYZ25	ARIYZ26	ARIYE10			
		ARIXPA1	ARIYE11			
ARICABED	ARICDMP	ARISSCB	ARIYE13	ARICMUD	ARICOMB	ARISYSDH*
ARIYE02	ARIYI01D	ARIYI02		ARIYM50		
ARIYI09	ARIYI15	ARIYI28				
ARIYK37	ARIYL15	ARIYT12				
ARIYZ19	ARIYZ25	ARIYZ26				
		ARIYM04	ARIYE131*			
(Called by many DBSS modules via TRACE macro)			ARIYE14	ARICDWT	ARICPDM	ARISYSD5*
				ARIYM04 (via MSG macro)		
	ARIYM05	ARIYT12	ARIYE16			
		ARIYT01	ARIYE37	ARIYI15	ARIYK37	ARIYL15
				ARIYM52	ARIYT12	
			ARIYI00	ARIYD44	ARIYI363*	ARIYI364*
				ARIYI41	ARIYK02	ARIYK08
				ARIYK09	ARIYM02	
ARIYS20	ARIYS24	ARIYX04	ARIYI001*			
ARIYS20	ARIYS24	ARIYX04	ARIYI003*			
		ARIYI05	ARIYI004*			
			ARIYI01D	ARIYE13	ARIYM02	
		ARIYM11	ARIYI02	ARIYE07	ARIYE13	ARIYI19
				ARIYI42	ARIYI46	
		ARIYM11	ARIYI04	ARISSCB	ARIYE06	ARIYE09
				ARIYI09	ARIYI15	ARIYI28
ARIYT14	ARIYT19	ARIYT20	ARIYI05	ARIYI004*	ARIYM02	
		ARIYT27				
ARIYI14	ARIYI21	ARIYI36	ARIYI06	ARIYI07	ARIYM02	
ARIYI06	ARIYI19	ARIYI29	ARIYI07(C	ARIYE14	ARIYK18	ARIYM02
ARIYI31	ARIYI36	ARIYI48		ARIYM04		
		ARIYI49				
ARIYI06	ARIYI19	ARIYI29	ARIYI07(D	ARIYM02		
ARIYI31	ARIYI36	ARIYI48				
ARIYI49	ARIYL21	ARIYL22				
		ARIYL23				
		ARIYI04	ARIYI09	ARIYE05	ARIYE07	ARIYE13
ARIYC11	ARIYC23	ARIYS08	ARIYI10	ARIYI262*	ARIYI362*	ARIYK18

* Asterisked items are secondary entry points. See page 346.] ⁰ indicates CMS EXEC

Calling Modules			Module/ Entry Pt	Called Modules/Entry Points		
		ARIYI20	ARIYI10	ARIYK19		
			ARIYI12			
	ARISIIO	ARISPFM	ARIYI13			
ARIYC02	ARIYC05	ARIYD02	ARIYI14	ARIYI06	ARIYI17	ARIYM02
ARIYD05	ARIYD19	ARIYD39				
ARIYD48	ARIYD50	ARIYD63				
ARIYK40	ARIYS00	ARIYX02				
ARIYX04	ARIYX07	ARIYX11				
ARIYI20	ARIYI29	ARIYI36	ARIYI141*			
		ARIYI29	ARIYI142*			
	ARIYE37	ARIYI04	ARIYI15	ARIYE07	ARIYE13	
		ARIYI14	ARIYI17			
		ARIYI19	ARIYI171*			
ARIYC02	ARIYC03	ARIYD05	ARIYI19	ARIYI07	ARIYI171*	ARIYI29
ARIYD06	ARIYD08	ARIYD23		ARIYI49	ARIYM02	
ARIYD26	ARIYD38	ARIYD39				
ARIYD40	ARIYD48	ARIYD50				
ARIYD77	ARIYI02	ARIYI28				
ARIYK38	ARIYK40	ARIYS24				
ARIYX02	ARIYX03	ARIYX04				
ARIYX07	ARIYX11	ARIYX12				
ARIYX16	ARIYX20	ARIYX21				
ARIYX29	ARIYX30	ARIYZ04				
		ARIYZ16				
		ARIYI29	ARIYI20	ARIYI12	ARIYI141*	ARIYI46
				ARIYM02		
	ARIYT00	ARIYT17	ARIYI21	ARIYI06	ARIYI314*	ARIYI315*
				ARIYI51		
ARISPFM	ARIYM00	ARIYT00	ARIYI22(C	ARICCOM	ARICTRM	ARICWSG
				ARISDSK	ARISIIO	ARIYE14
				ARIYI23	ARIYI312	ARIYI313
				ARIYL21	ARIYM02	ARIYM04
ARISPFM	ARIYM00	ARIYT00	ARIYI22!D	ARICWSG	ARISIIO	ARIYI23
				ARIYI312*	ARIYI313*	ARIYL21
	ARIYI22	ARIYL00	ARIYI23(C	ARICCOM	ARIYE07	ARIYE14
				ARIYM02	ARIYM04	
	ARIYI22D	ARIYL00	ARIYI23(D	ARIYE07	ARIYM02	
		ARIYL13	ARIYI24			
			ARIYI26	ARIYI41	ARIYK09	ARIYM02
ARISEGA	ARIYT00	ARIYT17	ARIYI261*			
		ARIYI10	ARIYI262*			
		ARIYI04	ARIYI28	ARIYE05	ARIYE07	ARIYE10
				ARIYE13	ARIYI19	ARIYI46
		ARIYI19	ARIYI29	ARIYI07	ARIYI141*	ARIYI142*
				ARIYI20	ARIYK19	ARIYM02
			ARIYI31	ARIYI07	ARIYM02	
		ARISIIO	ARIYI311*			
ARISDSK	ARISPFM	ARIYI22D	ARIYI312*			

* Asterisked items are secondary entry points. See page 346. [0] indicates CMS EXEC

Calling Modules			Module/ Entry Pt	Called Modules/Entry Points		
		ARIYI36				
ARISDSK	ARISEGA	ARISPFM	ARIYI313*			
	ARIYI22D	ARIYL14				
	ARIYI21	ARIYI36	ARIYI314*			
ARISDSK	ARISPFM	ARIYI21	ARIYI315*			
		ARIYI36				
			ARIYI36	ARICWSF	ARICWSG	ARIYD111*
				ARIYD44	ARIYI06	ARIYI07
				ARIYI141*	ARIYI312*	ARIYI314*
				ARIYI315*	ARIYI41	ARIYI45
				ARIYI46	ARIYK09	ARIYK18
				ARIYM02		
		ARIYI10	ARIYI362*			
		ARIYI00	ARIYI363*			
		ARIYI00	ARIYI364*			
	ARIYL08	ARIYL13	ARIYI365*			
		ARIYL08	ARIYI366*			
ARIYI00	ARIYI26	ARIYI36	ARIYI41	ARIYK08	ARIYK41	ARIYM02
ARIYC05	ARIYD38	ARIYI02	ARIYI42			
ARIYK01	ARIYK22	ARIYK28				
	ARIYK40	ARIYZ04				
ARIYC02	ARIYS00	ARIYS20	ARIYI421*			
		ARIYS24				
	ARIYC00	ARIYC01	ARIYI422*			
		ARIYL00	ARIYI423*			
		ARIYI36	ARIYI45	ARIYM02		
ARIYC06	ARIYC15	ARIYD01	ARIYI46	ARIYM02		
ARIYD05	ARIYD06	ARIYD08				
ARIYD11	ARIYD15	ARIYD20				
ARIYD33	ARIYD39	ARIYD48				
ARIYD50	ARIYI02	ARIYI20				
ARIYI28	ARIYI36	ARIYK38				
ARIYS00	ARIYS02	ARIYS10				
ARIYS11	ARIYS20	ARIYS22				
ARIYS23	ARIYS24	ARIYX01				
ARIYX02	ARIYX03	ARIYX04				
ARIYX05	ARIYX07	ARIYX11				
ARIYX12	ARIYX18	ARIYX20				
ARIYX21	ARIYX29	ARIYX30				
		ARIYZ16				
		ARICTRM	ARIYI47(D			
		ARIYM00	ARIYI47(C	ARICCOM	ARIYM04	
			ARIYI48	ARIYI07	ARIYM02	
		ARIYI19	ARIYI49	ARIYI07	ARIYM02	
			ARIYI50D	ARIYK18		
	ARIYI21	ARIYL09	ARIYI51			
ARIYD00	ARIYD06	ARIYD08	ARIYK00	ARIYK19	ARIYM02	
ARIYD20	ARIYD39	ARIYD50				
ARIYK41	ARIYS24	ARIYX12				

* Asterisked items are secondary entry points. See page 346. ⁰ indicates CMS EXEC

Calling Modules			Module/ Entry Pt	Called Modules/Entry Points		
	ARIYK01	ARIYK28	ARIYK001*			
			ARIYK01	ARIYD111*	ARIYI42	ARIYK001*
				ARIYK09	ARIYK38	ARIYK41
				ARIYM02		
		ARIYI00	ARIYK02	ARIYM02		
	ARIYD14	ARIYD70	ARIYK03			
			ARIYK04	ARIYK18		
			ARIYK05	ARIYK19	ARIYM02	
			ARIYK06	ARIYK18		
			ARIYK07	ARIYK19	ARIYM02	
ARIYC07	ARIYD43	ARIYI00	ARIYK08	ARIYK18		
ARIYI41	ARIYK28	ARIYL08				
ARIYL23	ARIYS00	ARIYT09				
ARIYT10	ARIYT14	ARIYT19				
		ARIYT27				
ARIYC06	ARIYD11	ARIYD55	ARIYK09	ARIYK19		
ARIYI00	ARIYI26	ARIYI36				
ARIYK01	ARIYK28	ARIYL08				
ARIYL23	ARIYS00	ARIYT09				
ARIYT10	ARIYT14	ARIYT19				
		ARIYT27				
	ARIYL08	ARIYL23	ARIYK10	ARIYK18	ARIYK19	
	ARIYL08	ARIYL23	ARIYK11	ARIYK19		
		ARIYK41	ARIYK12	ARICWSG	ARIYK19	ARIYM02
ARISFM1(C	ARIYI07(C	ARIYI10	ARIYK18	ARICDWT	ARIYM02	
ARIYI36	ARIYI50D	ARIYK04				
ARIYK06	ARIYK08	ARIYK10				
ARIYK41	ARIYL03	ARIYL09				
ARIYM10	ARIYM50(D	ARIYM51(D				
	ARIYT14	ARIYT19				
ARICCLA	ARICSHT	ARIYI10	ARIYK19	ARICOPT		
ARIYI29	ARIYK00	ARIYK05				
ARIYK07	ARIYK09	ARIYK10				
ARIYK11	ARIYK12	ARIYK21				
ARIYK41	ARIYL01	ARIYL03				
ARIYL24	ARIYT14	ARIYT19				
		ARIYK25	ARIYK21	ARIYK19		
			ARIYK22	ADDLIST	ARIYI42	ARIYK41
				ARIYM02		
			ARIYK25	ARIYK21		
			ARIYK28	ARIYD111*	ARIYI42	ARIYK001*
				ARIYK08	ARIYK09	ARIYK38
				ARIYK41	ARIYM02	
	ARIYK01	ARIYK28	ARIYK38	ARIYI19	ARIYI46	ARIYM02
		ARIYC11	ARIYK39	ARIYD26	ARIYK40	ARIYM02
ARIYC02	ARIYC05	ARIYC1D	ARIYK40	ARIYC09	ARIYI14	ARIYI19
	ARIYC11	ARIYK39		ARIYI42	ARIYK41	ARIYM02
ARIYD06	ARIYD08	ARIYD20	ARIYK41	ARIYK00	ARIYK12	ARIYK18
ARIYD32	ARIYD33	ARIYD38		ARIYK19	ARIYT23	

* Asterisked items are secondary entry points. See page 346. ⁰ indicates CMS EXEC

Calling Modules			Module/ Entry Pt	Called Modules/Entry Points		
ARIYD39	ARIYD41	ARIYD47				
ARIYD50	ARIYD74	ARIYD77				
ARIYI41	ARIYK01	ARIYK22				
ARIYK28	ARIYK40	ARIYL13				
ARIYS00	ARIYS24	ARIYT09				
ARIYX01	ARIYX03	ARIYX05				
ARIYX11	ARIYX12	ARIYX16				
ARIYX18	ARIYX20	ARIYX21				
	ARIYX29	ARIYX30				
		ARIYD41	ARIYK411*			
		ARIYT00	ARIYK42			
		ARIYT00	ARIYL00	ARICCRA	ARICENA	ARIYI23
				ARIYI423*	ARIYL08	ARIYL13
				ARIYL14	ARIYM02	
			ARIYL01	ARIYK19	ARIYM02	ARIYT16
			ARIYL02	ARIYM02		
			ARIYL03	ARIYK18	ARIYK19	ARIYM02
			ARIYL04	ARIYM02		
		ARIYC26	ARIYL05	ARIYM02		
			ARIYL06	ARIYM02		
	ARIYL13	ARIYT19	ARIYL07	ARIYC18	ARIYC19	ARIYC20
				ARIYC21	ARIYC22	ARIYC23
				ARIYC24	ARIYC25	ARIYC26
				ARIYD65	ARIYD66	ARIYD67
				ARIYD68	ARIYD69	ARIYM02
				ARIYS02	ARIYS08	
ARISEGA	ARIYL00	ARIYL09	ARIYL08	ARICTRM	ARICWSF	ARICWSG
				ARIYI365*	ARIYI366*	ARIYK08
				ARIYK09	ARIYK10	ARIYK11
				ARIYL23	ARIYM02	ARIYM04
			ARIYL09	ARICTRM	ARIYI51	ARIYK18
				ARIYL08	ARIYM02	
		ARIYL13	ARIYL10	ARIYM02		
		ARIYL13	ARIYL11			
		ARIYL13	ARIYL12			
		ARIYL00	ARIYL13	ARICCRA	ARICENA	ARICTRM
				ARICWSF	ARICWSG	ARISYSDG*
				ARISYSD2*	ARIYI24	ARIYI365*
				ARIYK41	ARIYL07	ARIYL10
				ARIYL11	ARIYL12	ARIYL19
				ARIYM02	ARIYT29	
		ARIYL00	ARIYL14	ARICTRM	ARISFM1	ARISYSD7*
				ARIYI313*		
	ARIYE37	ARIYT01	ARIYL15	ARIYE07	ARIYE13	
		ARIYL13	ARIYL19	ARIYM02		
		ARIYI22D	ARIYL21	ARICTRM	ARICWSG	ARISYSD5*
				ARIYE04	ARIYE06	ARIYI07
				ARIYM02		
		ARIYT00	ARIYL22	ARISYSD5*	ARISYSD7*	ARIYI07

* Asterisked items are secondary entry points. See page 346. ⁰ indicates CMS EXEC

Calling Modules			Module/ Entry Pt	Called Modules/Entry Points		
		ARIYL08	ARIYL23	ARICWSG	ARISYSD5*	ARIYE04
				ARIYE06	ARIYI07	ARIYK08
				ARIYK09	ARIYK10	ARIYK11
				ARIYM02		
		ARIYM11	ARIYL24	ARIYK19		
ARICCRA	ARIGCAT	ARISEGB	ARIYM00	ARIYE01	ARIYE011*	ARIYI22
		ARIXEDB		ARIYT05	ARIYT09	ARIYT15
				ARIYT22		
				Plus the following computed calls:		
				ARIYC00	ARIYC01	ARIYC02
				ARIYC10	ARIYC11	ARIYD00
				ARIYD01	ARIYD02	ARIYD03
				ARIYD04	ARIYD05	ARIYD06
				ARIYD08	ARIYD09	ARIYD10
				ARIYI00	ARIYI002*	ARIYK01
				ARIYK28	ARIYM101*	ARIYS00
				ARIYT06	ARIYT10	ARIYT14
				ARIYT17	ARIYT26	ARIYT27
				ARIYT28	ARIYZ00	ARIYZ15
		ARICIP2	ARIYM01	ARISYSDB*	ARIYM10	
ARICCLA	ARICCRA	ARICDSP	ARIYM02	ARICABE	ARICTRM	ARIYM04
ARICENA	ARICINT	ARICIP1				
ARICIP2	ARICMUD (via DS2SYSER)					
ARICOMB (via DS2SYSER)		ARICSHT				
ARICWAT (via DS2SYSER)		ARICSPM				
ARISDBR	ARISEGA	ARISFDB				
ARISFM1	ARISPFM	ARIYC02				
ARIXEER (via DS2SYSER)		ARIYC02				
ARIYC03	ARIYC05	ARIYC06				
ARIYC07	ARIYC11	ARIYC13				
ARIYC15	ARIYC23	ARIYD01				
ARIYD02	ARIYD04	ARIYD05				
ARIYD06	ARIYD08	ARIYD10				
ARIYD11	ARIYD15	ARIYD18				
ARIYD19	ARIYD20	ARIYD23				
ARIYD26	ARIYD32	ARIYD33				
ARIYD38	ARIYD39	ARIYD40				
ARIYD41	ARIYD43	ARIYD46				
ARIYD47	ARIYD48	ARIYD50				
ARIYD55	ARIYD62	ARIYD63				
ARIYD64	ARIYD69	ARIYD72				
ARIYD73	ARIYD74	ARIYD77				
ARIYD81	ARIYI00	ARIYI01D				
ARIYI05	ARIYI06	ARIYI07				
ARIYI14	ARIYI19	ARIYI20				
ARIYI22(C	ARIYI23	ARIYI26				
ARIYI29	ARIYI31	ARIYI36				
ARIYI41	ARIYI45	ARIYI46				
ARIYI48	ARIYI49	ARIYK00				

* Asterisked items are secondary entry points. See page 346. ⁰ indicates CMS EXEC

Calling Modules			Module/ Entry Pt	Called Modules/Entry Points		
ARIYK01	ARIYK02	ARIYK05				
ARIYK07	ARIYK12	ARIYK18				
ARIYK22	ARIYK28	ARIYK38				
ARIYK39	ARIYK40	ARIYL00				
ARIYL01	ARIYL02	ARIYL03				
ARIYL04	ARIYL05	ARIYL06				
ARIYL07	ARIYL08	ARIYL09				
ARIYL10	ARIYL13	ARIYL19				
ARIYL21	ARIYL23	ARIYS00				
ARIYS02	ARIYS08	ARIYS09				
ARIYS10	ARIYS11	ARIYS20				
ARIYS21	ARIYS22	ARIYS23				
ARIYS24	ARIYT00	ARIYT09				
ARIYT10	ARIYT14	ARIYT15				
ARIYT17	ARIYT19	ARIYT21				
ARIYX01	ARIYX02	ARIYX03				
ARIYX04	ARIYX05	ARIYX07				
ARIYX11	ARIYX12	ARIYX14				
ARIYX15	ARIYX16	ARIYX18				
ARIYX20	ARIYX21	ARIYX23				
ARIYX24	ARIYX25	ARIYX26				
ARIYX27	ARIYX29	ARIYX30				
ARIYZ02	ARIYZ03	ARIYZ04				
ARIYZ05	ARIYZ10	ARIYZ11				
	ARIYZ15	ARIYZ16				
	ARIYD58	ARIYS01	ARIYM03			
ARIIBIN	ARIICMD	ARIIDQY	ARIYM04*	ARIIFOR	ARIIRWI	
ARIIERS	ARIIFCC	ARIIFCI				
ARIIFCS	ARIIFMC	ARIIHLD				
	ARIISQL	ARIISQL8*				
ARIIHLP	ARIIIGN	ARIILST				
ARIINSQ	ARIIPQY	ARIIPSQ				
ARIIQRY	ARIIREC	ARIIRNM				
ARIIRPT	ARIIRUN	ARIISET				
ARIISMG	ARIISQLA	ARIISTR				
ARIIST2	ARIIST3	ARIISUB				
	ARIXEER	AXTCMSG				
(Called by many modules via MSG or RMSG macros)			ARIYM04	ARIYE131*	ARIYM05	ARIYM50
				ARIYM51		
		ARIYM04	ARIYM05	ARIYE07	ARIYE08	ARIYE16
		ARIYM01	ARIYM10	ARISYSDJ*	ARIYE04	ARIYK18
				ARIYM11		
		ARIYM00	ARIYM101*	ARISYSDJ* (VM)	ARIYE04	ARIYK18
				ARIYM11		
		ARIYM10	ARIYM11	ARICSHT	ARICTRC	ARIYI02
				ARIYI04	ARIYL24	ARIYT01
				ARIYT18	ARIYZ19	ARIYZ20
	ARIYE13	ARIYM04	ARIYM50	ARIYK18		

* Asterisked items are secondary entry points. See page 346. [0] indicates CMS EXEC

Calling Modules			Module/ Entry Pt	Called Modules/Entry Points		
		ARIYM04	ARIYM51	ARIYK18		
ARIYE37	ARIYZ19	ARIYZ20	ARIYM52			
			ARIYS00	ARIYD11	ARIYD26	ARIYD27
				ARIYD32	ARIYD38	ARIYD52
				ARIYD55	ARIYD57	ARIYD62
				ARIYI14	ARIYI421*	ARIYI46
				ARIYK08	ARIYK09	ARIYK41
				ARIYM02	ARIYS24	ARIYS241*
				ARIYS242*	ARIYS243*	ARIYX20
		ARIYS08	ARIYS001*			
		ARIYX04	ARIYS002*			
ARIYD06	ARIYD08	ARIYS24	ARIYS01	ARIYM03		
		ARIYL07	ARIYS02	ARIYD16	ARIYI46	ARIYM02
		ARIYD05	ARIYS03			
		ARIYD05	ARIYS04			
	ARIYD06	ARIYD08	ARIYS05			
		ARIYL07	ARIYS08	ARICWSG	ARIYI10	ARIYM02
				ARIYS001*		
		ARIYS24	ARIYS09	ARIYM02	ARIYS21	
		ARIYS24	ARIYS10	ARIYI46	ARIYM02	
	ARIYS20	ARIYS24	ARIYS11	ARIYI46	ARIYM02	
		ARIYS24	ARIYS12			
		ARIYS24	ARIYS121*			
		ARIYS24	ARIYS17			
		ARIYS24	ARIYS18			
		ARIYS24	ARIYS19			
		ARIYS24	ARIYS20	ARIYI001*	ARIYI003*	ARIYI421*
				ARIYI46	ARIYM02	ARIYS11
				ARIYS21	ARIYS22	ARIYS23
	ARIYS09	ARIYS20	ARIYS21	ARIYM02		
		ARIYS20	ARIYS22	ARIYI46	ARIYM02	
		ARIYS20	ARIYS23	ARIYI46	ARIYM02	
		ARIYS00	ARIYS24	ARICWSF	ARICWSG	ARIYD26
				ARIYD41	ARIYD58	ARIYI001*
				ARIYI003*	ARIYI19	ARIYI421*
				ARIYI46	ARIYK00	ARIYK41
				ARIYM02	ARIYS01	ARIYS09
				ARIYS10	ARIYS11	ARIYS12
				ARIYS121*	ARIYS17	ARIYS18
				ARIYS19	ARIYS20	ARIYX20
			ARIYS24			
		ARIYS00	ARIYS241*			
		ARIYS00	ARIYS242*			
		ARIYS00	ARIYS243*			
		ARICIP2	ARIYT00	ARICTRI	ARICTRM	ARISDSK
				ARISEGA	ARISPFM	ARIYI21
				ARIYI22	ARIYI261*	ARIYK42
				ARIYL00	ARIYL22	ARIYM02
				ARIYT02	ARIYT05	ARIYZ23

* Asterisked items are secondary entry points. See page 346. [0] indicates CMS EXEC

Calling Modules			Module/ Entry Pt	Called Modules/Entry Points		
		ARIYM11	ARIYT01	ARIYE37	ARIYK37	ARIYL15
				ARIYT12		
		ARIYT00	ARIYT02			
	ARIYM00	ARIYT00	ARIYT05			
		ARIXERD	ARIYT06	ARICWFR		
		ARIYE01	ARIYT08	ARIYE14 (via TRACE macro)		
		ARIYM00	ARIYT09	ARIYD87	ARIYD871*	ARIYK08
				ARIYK09	ARIYK41	ARIYM02
				ARIYT19	ARIYT22	
			ARIYT10	ARIYD87	ARIYD871*	ARIYK08
				ARIYK09	ARIYM02	ARIYT19
				ARIYT22		
	ARIYE37	ARIYT01	ARIYT12	ARIYE07	ARIYE13	ARIYE16
		ARIYT15	ARIYT14	ARIYI05	ARIYK08	ARIYK09
				ARIYK18	ARIYK19	ARIYM02
				ARIYT19	ARIYT22	
		ARIYC11	ARIYT141*			
ARICABE	ARICSPM	ARIYM00	ARIYT15	ARIXERD1* (via DS2XER1P)		ARIYM02
				ARIYT14	ARIYT19	ARIYT22
				ARIYT23		
	ARIYL01	ARIYT18	ARIYT16	ARIYT23		
			ARIYT17	ARIYI21	ARIYI261*	ARIYM02
				ARIYT19	ARIYT23	
		ARIYM11	ARIYT18	ARIYE08	ARIYT16	
ARIYD11	ARIYT09	ARIYT10	ARIYT19	ARIYI05	ARIYK08	ARIYK09
ARIYT14	ARIYT15	ARIYT17		ARIYK18	ARIYK19	ARIYL07
				ARIYM02	ARIYT20	ARIYT21
				ARIYT22		
		ARIYT19	ARIYT20	ARIYI05		
		ARIYT19	ARIYT21	ARIYD872*	ARIYM02	
ARIYD11	ARIYM00	ARIYT09	ARIYT22			
ARIYT10	ARIYT14	ARIYT15				
		ARIYT19				
ARICABE	ARICDSP	ARIYK41	ARIYT23			
ARIYT15	ARIYT16	ARIYT17				
			ARIYT26	ARIYE07		
			ARIYT27	ARIYI05	ARIYK08	ARIYK09
		ARIYL13	ARIYT29			
	ARIYX03	ARIYX16	ARIYX00	ARIYD61		
ARIYD02	ARIYD05	ARIYD10	ARIYX01	ARIYI46	ARIYK41	ARIYM02
	ARIYD69	ARIYD75		ARIYX03	ARIYX17	ARIYX21
				ARIYX30		
ARIYC14	ARIYC20	ARIYC25	ARIYX02	ADVANCE	ARIYI14	ARIYI19
		ARIYX04		ARIYI46	ARIYM02	
		ARIYX01	ARIYX03	ARIYD411*	ARIYI19	ARIYI46
			ARIYX03	ARIYK41	ARIYM02	ARIYX00
				ARIYX19		
ARIYC02	ARIYC20	ARIYC25	ARIYX04	ARICWSF	ARICWSG	ARIYD34
				ARIYD50	ARIYD531*	ARIYI001*

* Asterisked items are secondary entry points. See page 346. ⁰ indicates CMS EXEC

Calling Modules			Module/ Entry Pt	Called Modules/Entry Points		
				ARIYI003*	ARIYI14	ARIYI19
				ARIYI46	ARIYM02	ARIYS002*
				ARIYX02	ARIYX06	ARIYX07
				ARIYX08	ARIYX09	ARIYX10
		ARIYD06	ARIYX05	ARIYI46	ARIYK41	ARIYM02
				ARIYX17	ARIYX20	
		ARIYX04	ARIYX06			
	ARIYX04	ARIYX09	ARIYX07	ARIYI14	ARIYI19	ARIYI46
				ARIYM02		
	ARIYX04	ARIYX09	ARIYX08			
	ARIYX04	ARIYX09	ARIYX09	ARIYX07	ARIYX08	ARIYX09
		ARIYX04	ARIYX10			
		ARIYX29	ARIYX11	ARIYI14	ARIYI19	ARIYI46
				ARIYK41	ARIYM02	
		ARIYX20	ARIYX12	ARIYI19	ARIYI46	ARIYK00
				ARIYK41	ARIYM02	ARIYX21
		ARIYX20	ARIYX121*			
	ARIYX18	ARIYX29	ARIYXI4	ARIYM02		
		ARIYX29	ARIYX15	ARIYM02		
		ARIYX18	ARIYX16	ARIYI19	ARIYK41	ARIYM02
				ARIYX00	ARIYX19	
ARIYX01	ARIYX05	ARIYX18	ARIYX17			
	ARIYX20	ARIYX21				
ARIYD02	ARIYD05	ARIYD10	ARIYX18	ARIYI46	ARIYK41	ARIYM02
	ARIYD69	ARIYD75		ARIYX14	ARIYX16	ARIYX17
				ARIYX21	ARIYX29	ARIYX30
ARIYX03	ARIYX16	ARIYX23	ARIYX19	ARIYD61		
ARIYD06	ARIYD08	ARIYS00	ARIYX20	ARIYI19	ARIYI46	ARIYK41
	ARIYS24	ARIYX05		ARIYM02	ARIYX12	ARIYX121*
				ARIYX17	ARIYX21	
ARIYD05	ARIYD08	ARIYD46	ARIYX21	ARIYI19	ARIYI46	ARIYK41
ARIYX01	ARIYX12	ARIYX18		ARIYM02	ARIYX17	ARIYX24
		ARIYX20		ARIYX25	ARIYX26	ARIYX27
ARIYD02	ARIYD05	ARIYD10	ARIYX22			
ARIYD06	ARIYD08	ARIYZ05	ARIYX23	ARIYD34	ARIYM02	ARIYX19
		ARIYX21	ARIYX24	ARIYM02		
		ARIYX21	ARIYX25	ARIYM02		
		ARIYX21	ARIYX26	ARIYM02		
		ARIYX21	ARIYX27	ARIYM02		
ARIYD02	ARIYD05	ARIYD10	ARIYX28			
		ARIYX18	ARIYX29	ARIYI19	ARIYI46	ARIYK41
				ARIYM02	ARIYX11	ARIYX14
			ARIYX29	ARIYX15		
	ARIYX01	ARIYX18	ARIYX30	ARIYI19	ARIYI46	ARIYK41
				ARIYM02		
			ARIYZ00	ARICWSF	ARICWSG	ARIYZ01
				ARIYZ02	ARIYZ05	ARIYZ061*
				ARIYZ10	ARIYZ11	
	ARIYZ00	ARIYZ15	ARIYZ01			

* Asterisked items are secondary entry points. See page 346. ⁹ indicates CMS EXEC

Calling Modules			Module/ Entry Pt	Called Modules/Entry Points		
		ARIYZ00	ARIYZ02	ARIYM02	ARIYZ03	
		ARIYZ02	ARIYZ03	ARIYD40	ARIYM02	ARIYZ04
ARIYZ03	ARIYZ05	ARIYZ11	ARIYZ04	ARIYI19	ARIYI42	ARIYM02
		ARIYZ15				
		ARIYZ00	ARIYZ05	ARIYD23	ARIYD46	ARIYM02
				ARIYX23	ARIYZ04	
			ARIYZ06			
		ARIYZ00	ARIYZ061*			
		ARIYZ11	ARIYZ062*			
		ARIYZ15	ARIYZ063*			
		ARIYZ00	ARIYZ10	ARIYD53	ARIYD531*	ARIYM02
		ARIYZ00	ARIYZ11	ARIYD50	ARIYD531*	ARIYM02
				ARIYZ04	ARIYZ062*	
			ARIYZ15	ARICWSF	ARIYM02	ARIYZ01
				ARIYZ04	ARIYZ063*	ARIYZ16
		ARIYZ15	ARIYZ16	ARIYD23	ARIYI19	ARIYI46
				ARIYM02		
		ARIYM11	ARIYZ19	ARIYE07	ARIYE13	ARIYM52
		ARIYM11	ARIYZ20	ARIYM52		
		ARIYT00	ARIYZ23			
			ARIYZ25	ARIYE05	ARIYE07	ARIYE10
				ARIYE13		
			ARIYZ26	ARIYE05	ARIYE07	ARIYE10
				ARIYE13		
		ARIICMD	AXTCENT	ARIYM04		
		ARIIPSQ	AXTCEXTS*			
			ISQL[0]	ARISEMSG[0]	ARISESCP[0]	SQLISTRT[0]
		ARISYSIN[0]	I5748XXJ[0]			
			SQLADBEX[0]	ARICEMGC[0]	ARISPROD[0]	ARISQLLD[0]
				SQLSTART[0]		
			SQLADBSP[0]	ARICEMGC[0]	ARISPROD[0]	ARISQLLD[0]
				SQLSTART[0]		
		(see ARIPEIFA)	SQLADD	(see ARIPEIFA)		
		SQLDBINS[0]	SQLDBGEN[0]	ARICEMGC[0]	ARISPROD[0]	ARISQLLD[0]
				SQLSTART[0]		
SQLDBSU[0]	SQLINIT[0]	SQLPREP[0]	SQLDBID[0]	ARICEMGC[0]	ARICLOC (system product editor macro)	
called by data base machine owner			SQLDBINS[0]	ARICEMGC[0]	ARISDBKC	ARISEMSG[0]
				ARISESCP[0]	ARISISKC	ARISPROD[0]
				ARISQLLD[0]	ARISRMKC	SQLDBGEN[0]
				SQLSTART[0]		
			SQLDBSU[0]	ARICEMGG[0]	ARIDBS	ARISEMSG[0]
				ARISESCP[0]	ARISQLLD[0]	SQLDBID[0]
				SQLSTART[0]	ARI0MSGC[0]	ARI6MSGC[0]
				ARI8MSGC[0]		
			SQLGENLD[0]	ARICEMGC[0]	ARISDBKC[0]	ARISISKC[0]
				ARISPROD[0]	ARISQLLD[0]	
		SQLINIT[0]	SQLINIT[0]	ARICEMGC[0]	SQLDBID[0]	SQLINIT[0]
		ISQL[0]	SQLISTRT[0]	ARICEMGC[0]	ARISISBT (bootstrap module)	
				SQLDBID[0]		

* Asterisked items are secondary entry points. See page 346. [0] indicates CMS EXEC

Calling Modules			Module/ Entry Pt	Called Modules/Entry Points		
		(see ARIPEIFA)	SQLLEN	(see ARIPEIFA)		
			SQLLOG°	ARICEMGC°	ARISPROD°	ARISQLLD°
				SQLSTART°		
			SQLPREP°	ARICEMGC°	ARISQLLD°	SQLDBID°
				SQLSTART°		
ARISDBMA°	SQLADBEX°	SQLADBSP°	SQLSTART°	ARICEMGC°	ARISDBBT	ARIØFDEF°
SQLDBGEN°	SQLDBINS°	SQLDBSU°		ARISPROD°	ARISQLLD°	
	SQLLOG°	SQLPREP°				
			SQLTRFMT°	ARICEMGC°	ARIMTRA	
		User	5748XXJ°	ARISSLKE°		

* Asterisked items are secondary entry points. See page 346. ° indicates CMS EXEC

SECONDARY ENTRY POINTS - MODULES

Secondary Entry Point Name	Module Name(s)
ARICDSP1	ARICDSP
ARICINT1	ARICINT
ARICTRC1	ARICTRC
ARIICIC	ARIICICD
ARIIDEQ	ARIICI2D
	ARIISYSC
ARIIDGS	ARIIWRTD
ARIIDST	ARIIWRTD
ARIIDM	ARIIWRTD
ARIIENQ	ARIICI2D
ARIIEXIT	ARIICMD
ARIIFM	ARIICI2D (VSE)
	ARIISYSC (VM)
ARIIFCI1	ARIIFCI
ARIIFCI2	ARIIFCI
ARIIFCS1	ARIIFCS
ARIIFCS2	ARIIFCS
ARIIFCS3	ARIIFCS
ARIIFOL1	ARIIFOL
ARIIFUL	ARIIFULD
ARIIGM	ARIICI2D (VSE)
	ARIISYSC (VM)
ARIIOP	ARIICMD
ARIIMGA	ARIICICD
ARIIPQL1	ARIIPQL
ARIIPQY1	ARIIPQY
ARIIPQY2	ARIIPQY
ARIIPQY3	ARIIPQY
ARIIPQY4	ARIIPQY
ARIIPQY5	ARIIPQY
ARIIPQY6	ARIIPQY
ARIIROUT	ARIICI2D
ARIIRPT1	ARIIRPT
ARIIRPT2	ARIIRPT
ARIIRPT3	ARIIRPT
ARIIRPT4	ARIIRPT
ARIIRPT5	ARIIRPT
ARIIRTRN	ARIICIC2D (VSE)
	ARIINITC (VM)
ARIISCM	ARIISCMD
ARIISPGE	ARIICI2D
ARIISQL	ARIISQLA
ARIISQL1	ARIISQLA
ARIISQL2	ARIISQLA
ARIISQL3	ARIISQLA
ARIISQL4	ARIISQLA
ARIISQL5	ARIISQLA
ARIISQL6	ARIISQLA

ARIISQL7	ARIISQLA
ARIISQL8	ARIISQLA
ARIISQL9	ARIISQLA
ARIISCC1	ARIISCC
ARIISTC	ARISYSC
ARIIST21	ARIIST2
ARIIST22	ARIIST2
ARIIST23	ARIIST2
ARIIST24	ARIIST2
ARIIST31	ARIIST3
ARIIST32	ARIIST3
ARIIST33	ARIIST3
ARIIST34	ARIIST3
ARIISTX	ARIICI2D
ARIITDP	ARICI2D (VSE)
	ARIISYS (VM)
ARIITS	ARIICI2D(VSE)
	ARIISYSC (VM)
ARIIVFY	ARIIVFYD
	ARIIVFYC
ARIIWAIT	ARIICI2D
ARIIWRT	ARIIWRTD
ARIMDFP1	ARIMDFP
ARIMDFP2	ARIMDFP
ARIMFLP1	ARIMFLP
ARIMFLP2	ARIMFLP
ARIMFMT	ARIIFOR
ARISII01	ARISIIO
ARISYSD1	ARISYSD(VSE),
	ARISYSD(VM)
ARISYSD2	ARIICICD (VSE),
	ARIICI2D (VSE),
	ARISYSD
ARISYSD3	ARISYSD(VSE),
	ARISYSD(VM)
ARISYSD4	ARISYSD
ARISYSD5	ARISYSD
	ARISYSE(VM)
ARISYSD6	ARISYSD
ARISYSD7	ARISYSD
ARISYSD8	ARISYSD,
	ARISYSF(VM)
ARISYSD9	ARISYSD
ARISYSDA	ARISYSD
ARISYSDB	ARISYSD
ARISYSDC	ARISYSD
ARISYSDE	ARISYSD
ARISYSDF	ARISYSD
ARISYSDG	ARIICICD(VSE),
	ARIICI2D(VSE),
	ARIISYSC(VM),
	ARISYSD
ARISYSDH	ARISYSDD(VSE),

	ARISYSGC(VM)
ARISYSDI	ARISYSD,
	ARISYSDC(VM)
ARISYSDJ	ARISYSD
ARISYSDK	ARISYSD
ARIXEBR1	ARIXEBR
ARIXEDP1	ARIXEDP
ARIXELX1	ARIXELX
ARIXESX1	ARIXESX
ARIXESX2	ARIXESX
ARIXESX3	ARIXESX
ARIXI101	ARIXI10
ARIXI102	ARIXI10
ARIYC031	ARIYC03
ARIYC111	ARIYC11
ARIYC151	ARIYC15
ARIYD111	ARIYD11
ARIYD201	ARIYD20
ARIYD231	ARIYD23
ARIYD232	ARIYD23
ARIYD411	ARIYD41
ARIYD531	ARIYD53
ARIYD551	ARIYD55
ARIYD871	ARIYD87
ARIYD872	ARIYD87
ARIYE011	ARIYE01
ARIYE131	ARIYE13
ARIYI001	ARIYI00
ARIYI003	ARIYI00
ARIYI004	ARIYI00
ARIYI141	ARIYI14
ARIYI142	ARIYI14
ARIYI171	ARIYI17
ARIYI261	ARIYI26
ARIYI262	ARIYI26
ARIYI311	ARIYI31
ARIYI312	ARIYI31
ARIYI313	ARIYI31
ARIYI314	ARIYI31
ARIYI315	ARIYI31
ARIYI362	ARIYI36
ARIYI363	ARIYI36
ARIYI364	ARIYI36
ARIYI365	ARIYI36
ARIYI366	ARIYI36
ARIYI421	ARIYI42
ARIYI422	ARIYI42
ARIYI423	ARIYI42
ARIYK001	ARIYK00
ARIYK411	ARIYK41
ARIYM04 [1]	ARIIMSG
ARIYS001	ARIYS00
ARIYS002	ARIYS00
ARIYS121	ARIYS12

[1]Not to be confused with module ARIYM04.

ARIYS241	ARIYS24
ARIYS242	ARIYS24
ARIYS243	ARIYS24
ARIYT141	ARIYT14
ARIYX121	ARIYX12
ARIYZ061	ARIYZ06
ARIYZ062	ARIYZ06
ARIYZ063	ARIYZ06
AXTCEXTS	ARIIEXT [2]
	AXTCENTD [3]

[2] If Extract is not installed
[3] If Extract is installed

DATA AREAS TO MODULES CROSS REFERENCE

This section lists the mapping macro names for the data areas included in Section 5, Data Areas and cross references those modules that directly use the mapping macro. It does not include cross references for data areas that are indirectly referenced by any of the modules.

To locate the proper mapping macro name to look up in the cross reference list, use the following list of names (as the data areas are known in Section 5):

Data Area	Macro Name
AUT	ARIBAUT
AUX SLT	ARIBAUX
Note: In the BASE and CBASE structures, the ARIB... macros are for RDS and the YUSER for DBSS.	
BASE	ARIBTB, YUSER
(and Auxiliary Structures)	
KDOMAINS	ARIBKDM, YUSER
DOMAINS	ARIBDOM, YUSER
SARGS	ARIBSAR, YUSER
CBASE	ARIBCB, YCUSER
(and Auxiliary structures)	
SCR Auxiliary	ARIBSCR, YUSER
MCR Auxiliary	ARIBMCR, YUSER
ICR Auxiliary	ARIBICR, YUSER
LCR Auxiliary	ARIBLCR, YUSER
PLCR Auxiliary	YUSER
CPA	ARIDCPA
CPLIST	ARICCPL
Control Header Information	YCONTROL
SCR	
MCR	
ICR	
LCR	
PLCR	
CTB (Cursor Table) (Exists only to Preprocessor. SQL/DS never sees it.)	None
DBSPACE Entity Page Format	
VHEADER	YVPAGE
VTUPLE	YVPAGE
VFIELD	YVPAGE
FFDATA	YVPAGE
DBSPACE Index Page Format	
IPAGFMT (Template for Index Page)	YIPAGE
DCE	DCE
DCLLIST	ARIBDCL
DSCAREA	ARICDSC
DS2CVT	DS2CVT
DS2MODE	DS2MODE

Data Area	Macro Name
EIB	ARICEIB
FDESC	ARIGENIO
Format Control Block	ARIIFMB
GCB and Extension	ARIIGCB
GCGLOBWA	ARIBGCG
IVIND	ARIBIVI
IVNAMES	ARIBIVN
OVIND	ARIBOVI
OVNAMES	ARIBOVN
Log Pages and Records:	YLOGTMPL
LOGPAGE TSAVDATA	
LOGHEAD TABTDATA	
VFIELD TENDDATA	
SCHKDATA TEND1DAT	
SCHKDAT1 TPRLOCKS	
Log Data for a DBSI call to/from the Log Linkage:	YLOG
LCOMHEAD CINSERTI CDELETEI	
LINSERT CINSDOMI CDESCRPT	
LCONNECT CINSERTR CDELETER	
LDELETE CINSERTS CDELETES	
LDISCONN CINSERTL CDELETEL	
LUPDATE CIPLINKS SRTLOGTP	
LLISTINS CUPDATE	
LRBs	YLRBS
Mailbox Data Areas:	
IIFPARM	ARICIIF
OHDHEAD	ARICOHD
OIFPARM	ARICOIF
MASTER	YTAB
NLST	ARIBNLS
NUCON (partial mapping of VM NUCON area)	ARICNUC
OBASE	ARIYOBA
OCOMBLK	ARIIOCB
Optimizer Data Areas	
OPTAREA	ARIBOPT
TBA	ARIBTBA
CLA	ARIBCLA

Data Area	Macro Name
IDX	ARIBIDX
QAR	ARIBQAR
PDA	ARIBPDA
COSTS	ARIBCOS
CHOICES	ARIBCOS
MINIPLANS	ARIBMP
PLANVEC	ARIBPV
OPTREE	ARIBOP
Descriptor Records:	
BQUERY	ARIBBQD
COLUMN	ARIBCLD
CURSOR	ARIBCUD
ENTITY	ARIBENT
FIELD-NAME (BUILT-IN FUNCTION)	ARIBFND
JOIN	ARIBJO
LITNODE	ARIBLIT
QUERY	ARIBQD
RELATION (table)	ARIBRD
SET FUNCTION	ARIBSFN
SQUERY	ARIBSQD
PGCTRS	YTAB
PPOPGNST	ARIBPPO
PREPDSCB	none
PROGS	ARIBPRO
RDAREA	ARICRDA
RDCVT	ARICRDC
RDIIN	ARIBRDI
RDSCG	ARIBCG
Report Control Block	ARIIVAR
Resource Manager Data Areas:	
RECP	ARIRECP
RMAR	ARIRMAR
RMCV	ARIRMCV
RMGL	ARIRMGL
RMLA	ARIRMLA
RMLO	ARIRMLO
RMLT	ARIRMLT
RMMR	ARIRMMR
RMRE	ARIRMRE
RMSP	ARIRMSP
RMWL	ARIRMWL
RMXC	ARIRMXCC
Scan Table	
SCANS	ARIYSCA
SCB	YSCAN
SQLCA and SQLDA	ARIBSQL
SQLCA	SQLCA
SQLDA	SQLDA
SRTBASE	YSRTBASE
STACK(STK)	ARIBSTK

Data Area	Macro Name
STOLDSTR	ARIBSTL
TPMAP	YTPMAP
Trace Point Descriptor Module Structures	TRACMAP
Trace Point Output Objects	ARICTRO
WSAELEM	ARICWSA
YDBCB	YTAB
YRSSCVT(DBSSCVT)	YRSSCVT
YRSSTRAN(TBASE)	YRSSTRAN
YTABLE1(DBSSAREA)	YTABLE1
YTABLE1S (short version of YTABLE1)	YTABLE1S
YTABLE1U	YTABLE1U
YTABLE2	YTAB
YTABLE4	YTAB
VM Cross-Machine Communication	
VMCBLOCK	ARICVMC
VMH	ARICVMH
VMQ	ARICVMQ

Data Area Macro Name	Referencing Modules
ARIBAUT	ARIPLTP ARIPPRA ARIPPRC ARIPPRP ARIXA02 ARIXA04 ARIXA06 ARIXA07 ARIXA08 ARIXA09 ARIXA10 ARIXA11 ARIXEAD ARIXEDS ARIXELX ARIXERD ARIXERP ARIXI03 ARIXI07 ARIXI09 ARIXI13 ARIXI15 ARIXI22 ARIXI23 ARIXI25 ARIXI26 ARIXI27 ARIXI28 ARIXI29 ARIXI30 ARIXI31 ARIXI32 ARIXOCT ARIXOCY ARIXOIN ARIXOOP ARIXOSC ARIXOSO ARIXOTS ARIXOVD
ARIBAUX	ARIXCR4 ARIXF21 ARIXF47 ARIPLTP ARIPPRA ARIPPRC ARIPPRP ARIPTXA ARIPTXC ARIPTXP ARIXA01 ARIXA09 ARIXC03 ARIXC13 ARIXC18 ARIXEAB ARIXEAD ARIXEDP ARIXEDS ARIXEFB ARIXELX ARIXEPH ARIXEPP ARIXERA ARIXERD ARIXERP ARIXEST ARIXESX ARIXI10 ARIXI12 ARIXI30 ARIXOBY ARIXOMS ARIXOVC ARIXOVD ARIXPA0 ARIXPA1 ARIXPA2 ARIXPA3
ARIBBQD	ARIMPT1 ARIMPT2C ARIMPT2D ARIMPT3 ARIMPT4 ARIMPT5 ARIPERR ARIPLTP ARIPMSH ARIPPRA ARIPPRC ARIPPRP ARIPSQA ARIPSQ ARIPTXA ARIPTXC ARIPTXP ARIXA01 ARIXA02 ARIXA03 ARIXA04 ARIXA06 ARIXA07 ARIXA08 ARIXA09 ARIXA10 ARIXA11 ARIXC01 ARIXC02 ARIXC03 ARIXC05 ARIXC06 ARIXC07 ARIXC08 ARIXC10 ARIXC11 ARIXC13 ARIXC14 ARIXC15 ARIXC16 ARIXC17 ARIXC18 ARIXC19 ARIXC20 ARIXC21 ARIXC22 ARIXC23 ARIXC24 ARIXC25 ARIXC27 ARIXC28 ARIXC29 ARIXC30 ARIXC31 ARIXC32 ARIXC33 ARIXC34 ARIXC35 ARIXC36 ARIXC37 ARIXC38 ARIXC39 ARIXC40 ARIXC41 ARIXC42 ARIXC43 ARIXC45 ARIXC48 ARIXC51 ARIXC54 ARIXEAB ARIXEAD ARIXEBR ARIXECK ARIXECL ARIXEDC ARIXEDP ARIXEDS ARIXEER ARIXEFB ARIXELX ARIXEPH ARIXEPP ARIXERA ARIXERD ARIXERO ARIXERP ARIXESP ARIXEST ARIXESX ARIXIST ARIXI01 ARIXI03 ARIXI04 ARIXI06 ARIXI07 ARIXI08 ARIXI09 ARIXI10 ARIXI12 ARIXI13 ARIXI14 ARIXI15 ARIXI18 ARIXI19 ARIXI20 ARIXI22 ARIXI23 ARIXI24 ARIXI25 ARIXI26 ARIXI27 ARIXI28 ARIXI29 ARIXI30 ARIXI31 ARIXI32 ARIXI33 ARIXOBY ARIXOB1 ARIXOB2 ARIXOCA ARIXOCK ARIXOCS ARIXOCT ARIXOCU ARIXOCY ARIXODF ARIXODM ARIXOD1 ARIXOD2 ARIXOEX ARIXOFC ARIXOFE ARIXOFF ARIXOFP ARIXOFQ ARIXOFR ARIXOFT ARIXOGA ARIXOGB ARIXOGC ARIXOGP ARIXOIN ARIXOIU ARIXOLC ARIXOLF ARIXOMA ARIXOMB ARIXOMD ARIXOMF ARIXOML ARIXOMS ARIXONV ARIXONW ARIXOOP ARIXOOS ARIXOQ1 ARIXOQ2 ARIXORM ARIXOSC ARIXOSG ARIXOSL ARIXOSO ARIXOSR ARIXOSS ARIXOST ARIXOSU ARIXOS3 ARIXOTF ARIXOTS ARIXOUS ARIXOVC ARIXOVD ARIXOV1 ARIXOW1 ARIXOW2 ARIXO1S ARIXPA0 ARIXPA1 ARIXPA2 ARIXPA3
ARIBCB	ARIXCRA ARIMPT1 ARIMPT2C ARIMPT2D ARIMPT3 ARIMPT4 ARIMPT5 ARIPPRA ARIPPRC ARIPPRP ARISDFL ARISDFM ARISDFR ARISEG ARISFIL ARISINI ARIXA01 ARIXA02 ARIXA03 ARIXA04 ARIXA06 ARIXA07 ARIXA08 ARIXA09 ARIXA10 ARIXA11 ARIXC38 ARIXEAB ARIXEAD ARIXECK ARIXECL ARIXEDR ARIXEDS ARIXEFB ARIXELX ARIXEPH ARIXEPP ARIXERA ARIXERD ARIXERO ARIXERP ARIXESP ARIXEST ARIXESX ARIXIST ARIXI01 ARIXI03 ARIXI04 ARIXI06 ARIXI07 ARIXI08 ARIXI09 ARIXI12 ARIXI13 ARIXI15 ARIXI18 ARIXI19 ARIXI20 ARIXI22 ARIXI23 ARIXI24 ARIXI25 ARIXI26 ARIXI27 ARIXI28 ARIXI29 ARIXI30 ARIXI31 ARIXI32 ARIXI33 ARIXOB1 ARIXOB2 ARIXOCA ARIXOCK ARIXOCT ARIXOCU ARIXOCY ARIXODM ARIXOD1 ARIXOD2 ARIXOFF ARIXOFR ARIXOGA ARIXOGP ARIXOIN ARIXOIU ARIXOLF ARIXOMA ARIXOMF ARIXOML ARIXOMS ARIXONV ARIXOOP ARIXOSO ARIXOSR ARIXOSS ARIXOST ARIXOUS ARIXOVC ARIXOVD ARIXO1S
ARIBCG	ARIXC01 ARIXC02 ARIXC03 ARIXC05 ARIXC06 ARIXC07 ARIXC10 ARIXC13 ARIXC14 ARIXC15 ARIXC16 ARIXC17 ARIXC18 ARIXC19 ARIXC20 ARIXC21 ARIXC22 ARIXC24 ARIXC25 ARIXC27 ARIXC28 ARIXC29 ARIXC30 ARIXC31 ARIXC32 ARIXC34 ARIXC35 ARIXC36 ARIXC37 ARIXC38 ARIXC39 ARIXC40 ARIXC41 ARIXC42 ARIXC43 ARIXC45 ARIXC48 ARIXC51 ARIXC53 ARIXC54
ARIBCLA	ARIXI09 ARIXOAF ARIXOBY ARIXOB1 ARIXOB2 ARIXOCA ARIXOCK ARIXOCS ARIXOCT ARIXOCU ARIXOCY ARIXODF ARIXODM ARIXOD1 ARIXOD2 ARIXOEX ARIXOFC ARIXOFE ARIXOFF ARIXOFP ARIXOFQ ARIXOFR ARIXOFT ARIXOGA ARIXOGB ARIXOGC ARIXOGP ARIXOIN ARIXOIU ARIXOLC ARIXOLF ARIXOMA ARIXOMB ARIXOMD ARIXOMF ARIXOML ARIXOMS ARIXONV ARIXONW ARIXOOP ARIXOOS ARIXOQ1 ARIXOQ2 ARIXORM ARIXOSC ARIXOSG ARIXOSL ARIXOSO ARIXOSR ARIXOSS ARIXOST ARIXOSU ARIXOTF ARIXOTS ARIXOUS ARIXOVC ARIXOVD ARIXOV1 ARIXOW1 ARIXOW2 ARIXO1S ARIXTR1
ARIBCLD	ARIMPT1 ARIMPT2C ARIMPT2D ARIMPT3 ARIMPT4 ARIMPT5 ARIPERR ARIPLTP ARIPMSH ARIPPRA ARIPPRC ARIPPRP ARIPSQA ARIPSQ ARIPTXA ARIPTXC ARIPTXP ARIXA01 ARIXA02 ARIXA03 ARIXA04 ARIXA06 ARIXA07 ARIXA08 ARIXA09 ARIXA10 ARIXA11 ARIXC01 ARIXC02 ARIXC03 ARIXC05 ARIXC06 ARIXC07 ARIXC08 ARIXC10 ARIXC11 ARIXC13 ARIXC14 ARIXC15 ARIXC16 ARIXC17 ARIXC18 ARIXC19 ARIXC20 ARIXC21 ARIXC22 ARIXC23 ARIXC24 ARIXC25 ARIXC27 ARIXC28 ARIXC29 ARIXC30 ARIXC31 ARIXC32 ARIXC33 ARIXC34 ARIXC35 ARIXC36 ARIXC37 ARIXC38 ARIXC39 ARIXC40 ARIXC41 ARIXC42 ARIXC43 ARIXC45 ARIXC48 ARIXC51 ARIXC54 ARIXEAB ARIXEAD ARIXEBR ARIXECK ARIXECL ARIXEDC ARIXEDP ARIXEDS ARIXEER ARIXEFB ARIXELX ARIXEPH ARIXEPP ARIXERA ARIXERD ARIXERO ARIXERP ARIXESP ARIXEST ARIXESX ARIXIST ARIXI01 ARIXI03 ARIXI04 ARIXI06 ARIXI07 ARIXI08 ARIXI09 ARIXI10 ARIXI12 ARIXI13 ARIXI14 ARIXI15 ARIXI18 ARIXI19 ARIXI20 ARIXI22 ARIXI23 ARIXI24 ARIXI25 ARIXI26 ARIXI27 ARIXI28 ARIXI29 ARIXI30 ARIXI31 ARIXI32 ARIXI33 ARIXOBY ARIXOB1 ARIXOB2 ARIXOCA ARIXOCK ARIXOCS ARIXOCT ARIXOCU ARIXOCY ARIXODF ARIXODM ARIXOD1 ARIXOD2 ARIXOEX ARIXOFC ARIXOFE ARIXOFF ARIXOFP ARIXOFQ ARIXOFR ARIXOFT ARIXOGA

Data Area Macro Name	Referencing Modules
	ARIXOGB ARIXOGC ARIXOGP ARIXOIN ARIXOIU ARIXOLC ARIXOLF ARIXOMA ARIXOMB ARIXOMD ARIXOMF ARIXOML ARIXOMS ARIXONV ARIXONW ARIXOOP ARIXOOS ARIXOQ1 ARIXOQ2 ARIXORM ARIXOSC ARIXOSG ARIXOSL ARIXOSO ARIXOSR ARIXOSS ARIXOST ARIXOSU ARIXOS3 ARIXOTF ARIXOTS ARIXOUS ARIXOVC ARIXOVD ARIXOV1 ARIXOW1 ARIXOW2 ARIXO1S ARIXPA0 ARIXPA1 ARIXPA2 ARIXPA3
ARIBCOS	ARIXOCS ARIXOGP ARIXOTS
ARIBCUD	ARIXCR4 ARIMPT1 ARIMPT2C ARIMPT2D ARIMPT3 ARIMPT4 ARIMPT5 ARIPERR ARIPLTP ARIPMSH ARIPPRA ARIPPRC ARIPPRP ARIPSQ ARIPSQP ARIPTXA ARIPTXC ARIPTXP ARIXA01 ARIXA02 ARIXA03 ARIXA04 ARIXA06 ARIXA07 ARIXA08 ARIXA09 ARIXA10 ARIXA11 ARIXC01 ARIXC02 ARIXC03 ARIXC05 ARIXC06 ARIXC07 ARIXC08 ARIXC10 ARIXC11 ARIXC13 ARIXC14 ARIXC15 ARIXC16 ARIXC17 ARIXC18 ARIXC19 ARIXC20 ARIXC21 ARIXC22 ARIXC23 ARIXC24 ARIXC25 ARIXC27 ARIXC28 ARIXC29 ARIXC30 ARIXC31 ARIXC32 ARIXC33 ARIXC34 ARIXC35 ARIXC36 ARIXC37 ARIXC38 ARIXC39 ARIXC40 ARIXC41 ARIXC42 ARIXC43 ARIXC45 ARIXC48 ARIXC51 ARIXC54 ARIXEAB ARIXEAD ARIXEBR ARIXECK ARIXECL ARIXEDC ARIXEDP ARIXEDS ARIXEER ARIXEFB ARIXELX ARIXEPH ARIXEPP ARIXERA ARIXERD ARIXERO ARIXERP ARIXESP ARIXEST ARIXESX ARIXIST ARIXI01 ARIXI03 ARIXI04 ARIXI06 ARIXI07 ARIXI08 ARIXI09 ARIXI10 ARIXI12 ARIXI13 ARIXI14 ARIXI15 ARIXI18 ARIXI19 ARIXI20 ARIXI22 ARIXI23 ARIXI24 ARIXI25 ARIXI26 ARIXI27 ARIXI28 ARIXI29 ARIXI30 ARIXI31 ARIXI32 ARIXI33 ARIXOBY ARIXOB1 ARIXOB2 ARIXOCA ARIXOCK ARIXOCS ARIXOCT ARIXOCU ARIXOCY ARIXODF ARIXODM ARIXOD1 ARIXOD2 ARIXOEX ARIXOFC ARIXOFE ARIXOFF ARIXOFP ARIXOFQ ARIXOFR ARIXOFT ARIXOGA ARIXOGB ARIXOGC ARIXOGP ARIXOIN ARIXOIU ARIXOLC ARIXOLF ARIXOMA ARIXOMB ARIXOMD ARIXOMF ARIXOML ARIXOMS ARIXONV ARIXONW ARIXOOP ARIXOOS ARIXOQ1 ARIXOQ2 ARIXORM ARIXOSC ARIXOSG ARIXOSL ARIXOSO ARIXOSR ARIXOSS ARIXOST ARIXOSU ARIXOS3 ARIXOTF ARIXOTS ARIXOUS ARIXOVC ARIXOVD ARIXOV1 ARIXOW1 ARIXOW2 ARIXO1S ARIXPA0 ARIXPA1 ARIXPA2 ARIXPA3
ARIBDCL	ARIPLTP ARIPPRA ARIPPRC ARIPPRP ARIPSQA ARIPSQC ARIPSQP ARIPTXA ARIPTXC ARIPTXP ARIXEAB ARIXEAD ARIXEDP ARIXEDS ARIXEFB ARIXELX ARIXEPP ARIXERD ARIXERP ARIXESX ARIXI09 ARIXOVC ARIXOVD
ARIBDOM	ARIMPT1 ARIMPT2C ARIMPT2D ARIMPT3 ARIMPT4 ARIMPT5 ARIPPRA ARIPPRC ARIPPRP ARISCNV ARISEGB ARISFIL ARISIST ARISIST2 ARISLAY ARISUPD ARIXA01 ARIXA02 ARIXA04 ARIXA06 ARIXA07 ARIXA08 ARIXA09 ARIXA10 ARIXA11 ARIXEAD ARIXEDR ARIXEDS ARIXELX ARIXERD ARIXERO ARIXERP ARIXESX ARIXIST ARIXI01 ARIXI03 ARIXI04 ARIXI06 ARIXI07 ARIXI08 ARIXI09 ARIXI13 ARIXI15 ARIXI18 ARIXI19 ARIXI20 ARIXI22 ARIXI23 ARIXI24 ARIXI25 ARIXI27 ARIXI29 ARIXI30 ARIXI33 ARIXOCA ARIXOCK ARIXODM ARIXOD1 ARIXOD2 ARIXOFF ARIXOIU ARIXOUS ARIXOVC ARIXOVD
ARIBENT	ARIMPT1 ARIMPT2C ARIMPT2D ARIMPT3 ARIMPT4 ARIMPT5 ARIPERR ARIPLTP ARIPMSH ARIPPRA ARIPPRC ARIPPRP ARIPSQA ARIPSQ ARIPTXA ARIPTXC ARIPTXP ARIXA01 ARIXA02 ARIXA03 ARIXA04 ARIXA06 ARIXA07 ARIXA08 ARIXA09 ARIXA10 ARIXA11 ARIXC01 ARIXC02 ARIXC03 ARIXC05 ARIXC06 ARIXC07 ARIXC08 ARIXC10 ARIXC11 ARIXC13 ARIXC14 ARIXC15 ARIXC16 ARIXC17 ARIXC18 ARIXC19 ARIXC20 ARIXC21 ARIXC22 ARIXC23 ARIXC24 ARIXC25 ARIXC27 ARIXC28 ARIXC29 ARIXC30 ARIXC31 ARIXC32 ARIXC33 ARIXC34 ARIXC35 ARIXC36 ARIXC37 ARIXC38 ARIXC39 ARIXC40 ARIXC41 ARIXC42 ARIXC43 ARIXC45 ARIXC48 ARIXC51 ARIXC54 ARIXEAB ARIXEAD ARIXEBR ARIXECK ARIXECL ARIXEDC ARIXEDP ARIXEDS ARIXEER ARIXEFB ARIXELX ARIXEPH ARIXEPP ARIXERA ARIXERD ARIXERO ARIXERP ARIXESP ARIXEST ARIXESX ARIXIST ARIXI01 ARIXI03 ARIXI04 ARIXI06 ARIXI07 ARIXI08 ARIXI09 ARIXI10 ARIXI12 ARIXI13 ARIXI14 ARIXI15 ARIXI18 ARIXI19 ARIXI20 ARIXI22 ARIXI23 ARIXI24 ARIXI25 ARIXI26 ARIXI27 ARIXI28 ARIXI29 ARIXI30 ARIXI31 ARIXI32 ARIXI33 ARIXOBY ARIXOB1 ARIXOB2 ARIXOCA ARIXOCK ARIXOCS ARIXOCT ARIXOCU ARIXOCY ARIXODF ARIXODM ARIXOD1 ARIXOD2 ARIXOEX ARIXOFC ARIXOFE ARIXOFF ARIXOFP ARIXOFQ ARIXOFR ARIXOFT ARIXOGA ARIXOGB ARIXOGC ARIXOGP ARIXOIN ARIXOIU ARIXOLC ARIXOLF ARIXOMA ARIXOMB ARIXOMD ARIXOMF ARIXOML ARIXOMS ARIXONV ARIXONW ARIXOOP ARIXOOS ARIXOQ1 ARIXOQ2 ARIXORM ARIXOSC ARIXOSG ARIXOSL ARIXOSO ARIXOSR ARIXOSS ARIXOST ARIXOSU ARIXOS3 ARIXOTF ARIXOTS ARIXOUS ARIXOVC ARIXOVD ARIXOV1 ARIXOW1 ARIXOW2 ARIXO1S ARIXPA0 ARIXPA1 ARIXPA2 ARIXPA3
ARIBFND	ARIMPT1 ARIMPT2C ARIMPT2D ARIMPT3 ARIMPT4 ARIMPT5 ARIPERR ARIPLTP ARIPMSH ARIPPRA ARIPPRC ARIPPRP ARIPSQA ARIPSQ ARIPTXA ARIPTXC ARIPTXP ARIXA01 ARIXA02 ARIXA03 ARIXA04 ARIXA06 ARIXA07 ARIXA08 ARIXA09 ARIXA10 ARIXA11 ARIXC01 ARIXC02 ARIXC03 ARIXC05 ARIXC06 ARIXC07 ARIXC08 ARIXC10 ARIXC11 ARIXC13 ARIXC14 ARIXC15 ARIXC16 ARIXC17 ARIXC18 ARIXC19 ARIXC20 ARIXC21 ARIXC22 ARIXC23 ARIXC24 ARIXC25 ARIXC27 ARIXC28 ARIXC29 ARIXC30 ARIXC31 ARIXC32 ARIXC33 ARIXC34 ARIXC35 ARIXC36 ARIXC37 ARIXC38 ARIXC39 ARIXC40 ARIXC41 ARIXC42 ARIXC43 ARIXC45 ARIXC48 ARIXC51 ARIXC54 ARIXEAB ARIXEAD ARIXEBR ARIXECK ARIXECL ARIXEDC ARIXEDP ARIXEDS ARIXEER ARIXEFB ARIXELX ARIXEPH ARIXEPP ARIXERA ARIXERD ARIXERO ARIXERP ARIXESP ARIXEST ARIXESX ARIXIST ARIXI01 ARIXI03 ARIXI04 ARIXI06 ARIXI07 ARIXI08 ARIXI09

Data Area Macro Name	Referencing Modules
	ARIXI10 ARIXI12 ARIXI13 ARIXI14 ARIXI15 ARIXI18 ARIXI19 ARIXI20 ARIXI22 ARIXI23 ARIXI24 ARIXI25 ARIXI26 ARIXI27 ARIXI28 ARIXI29 ARIXI30 ARIXI31 ARIXI32 ARIXI33 ARIXOBY ARIXOB1 ARIXOB2 ARIXOCA ARIXOCK ARIXOCS ARIXOCT ARIXOCU ARIXOCY ARIXODF ARIXODM ARIXOD1 ARIXOD2 ARIXOEX ARIXOFC ARIXOFE ARIXOFF ARIXOFP ARIXOFQ ARIXOFR ARIXOFT ARIXOGA ARIXOGB ARIXOGC ARIXOGP ARIXOIN ARIXOIU ARIXOLC ARIXOLF ARIXOMA ARIXOMB ARIXOMD ARIXOMF ARIXOML ARIXOMS ARIXONV ARIXONW ARIXOOP ARIXOOS ARIXOQ1 ARIXOQ2 ARIXORM ARIXOSC ARIXOSG ARIXOSL ARIXOSO ARIXOSR ARIXOSS ARIXOST ARIXOSU ARIXOS3 ARIXOTF ARIXOTS ARIXOUS ARIXOVC ARIXOVD ARIXOV1 ARIXOW1 ARIXOW2 ARIXO1S ARIXPA0 ARIXPA1 ARIXPA2
ARIBGCG	ARISCNV ARISCOL ARISDFL ARISDFM ARISDFR ARISEDI ARISERR ARISFIL ARISINI ARISIST ARISIST2 ARISLAY ARISMAI ARISSET
ARIBICR	ARIMPT1 ARIMPT2C ARIMPT2D ARIMPT3 ARIMPT4 ARIMPT5 ARIPPRA ARIPPRC ARIPPRP ARISDFM ARIXA01 ARIXA02 ARIXA03 ARIXA04 ARIXA06 ARIXA07 ARIXA08 ARIXA09 ARIXA10 ARIXA11 ARIXC38 ARIXEAB ARIXEAD ARIXECK ARIXECL ARIXEDR ARIXEDS ARIXEFB ARIXELX ARIXEPH ARIXEPP ARIXERA ARIXERD ARIXERO ARIXERP ARIXESP ARIXEST ARIXESX ARIXIST ARIXI01 ARIXI03 ARIXI04 ARIXI06 ARIXI07 ARIXI08 ARIXI09 ARIXI12 ARIXI13 ARIXI15 ARIXI18 ARIXI19 ARIXI20 ARIXI22 ARIXI23 ARIXI24 ARIXI25 ARIXI26 ARIXI27 ARIXI28 ARIXI29 ARIXI30 ARIXI31 ARIXI32 ARIXI33 ARIXOB1 ARIXOB2 ARIXOCA ARIXOCK ARIXOCT ARIXOCU ARIXOCY ARIXODM ARIXOD1 ARIXOD2 ARIXOFF ARIXOFR ARIXOGA ARIXOGP ARIXOIN ARIXOIU ARIXOLF ARIXOMA ARIXOMF ARIXOML ARIXOMS ARIXONV ARIXOOP ARIXOSO ARIXOSR ARIXOSS ARIXOST ARIXOUS ARIXOVC ARIXOVD ARIXO1S
ARIBIDX	ARIXI06 ARIXI09 ARIXI19 ARIXI20 ARIXI33 ARIXOAF ARIXOBY ARIXOB1 ARIXOB2 ARIXOCA ARIXOCK ARIXOCS ARIXOCT ARIXOCU ARIXOCY ARIXODF ARIXODM ARIXOD1 ARIXOD2 ARIXOEX ARIXOFC ARIXOFE ARIXOFF ARIXOFP ARIXOFQ ARIXOFR ARIXOFT ARIXOGA ARIXOGB ARIXOGC ARIXOGP ARIXOIN ARIXOIU ARIXOLC ARIXOLF ARIXOMA ARIXOMB ARIXOMD ARIXOMF ARIXOML ARIXOMS ARIXONV ARIXONW ARIXOOP ARIXOOS ARIXOQ1 ARIXOQ2 ARIXORM ARIXOSC ARIXOSG ARIXOSL ARIXOSO ARIXOSR ARIXOSS ARIXOST ARIXOSU ARIXOTF ARIXOTS ARIXOUS ARIXOVC ARIXOVD ARIXOV1 ARIXOW1 ARIXOW2 ARIXO1S ARIXTR1
ARIBIVI	ARIPLTP ARIPPRA ARIPPRC ARIPPRP ARIPTXA ARIPTXC ARIPTXP ARIXA01 ARIXA09 ARIXC03 ARIXC13 ARIXC18 ARIXEAB ARIXEAD ARIXEDP ARIXEDS ARIXEFB ARIXELX ARIXEPH ARIXEPP ARIXERA ARIXERD ARIXERP ARIXEST ARIXESX ARIXI10 ARIXI12 ARIXI30 ARIXOBY ARIXOMS ARIXOVC ARIXOVD ARIXPA0 ARIXPA1 ARIXPA2 ARIXPA3
ARIBIVN	ARIPLTP ARIPPRA ARIPPRC ARIPPRP ARIPTXA ARIPTXC ARIPTXP ARIXA01 ARIXA09 ARIXC03 ARIXC13 ARIXC18 ARIXEAB ARIXEAD ARIXEDP ARIXEDS ARIXEFB ARIXELX ARIXEPH ARIXEPP ARIXERA ARIXERD ARIXERP ARIXEST ARIXESX ARIXI10 ARIXI12 ARIXI30 ARIXOBY ARIXOMS ARIXOVC ARIXOVD ARIXPA0 ARIXPA1 ARIXPA2 ARIXPA3
ARIBJD	ARIMPT1 ARIMPT2C ARIMPT2D ARIMPT3 ARIMPT4 ARIMPT5 ARIPERR ARIPLTP ARIPMSH ARIPPRA ARIPPRC ARIPPRP ARIPSQA ARIPSQ ARIPTXA ARIPTXC ARIPTXP ARIXA01 ARIXA02 ARIXA03 ARIXA04 ARIXA06 ARIXA07 ARIXA08 ARIXA09 ARIXA10 ARIXA11 ARIXC01 ARIXC02 ARIXC03 ARIXC05 ARIXC06 ARIXC07 ARIXC08 ARIXC10 ARIXC11 ARIXC13 ARIXC14 ARIXC15 ARIXC16 ARIXC17 ARIXC18 ARIXC19 ARIXC20 ARIXC21 ARIXC22 ARIXC23 ARIXC24 ARIXC25 ARIXC27 ARIXC28 ARIXC29 ARIXC30 ARIXC31 ARIXC32 ARIXC33 ARIXC34 ARIXC35 ARIXC36 ARIXC37 ARIXC38 ARIXC39 ARIXC40 ARIXC41 ARIXC42 ARIXC43 ARIXC45 ARIXC48 ARIXC51 ARIXC54 ARIXEAB ARIXEAD ARIXEBR ARIXECK ARIXECL ARIXEDC ARIXEDP ARIXEDS ARIXEER ARIXEFB ARIXELX ARIXEPH ARIXEPP ARIXERA ARIXERD ARIXERO ARIXERP ARIXESP ARIXEST ARIXESX ARIXIST ARIXI01 ARIXI03 ARIXI04 ARIXI06 ARIXI07 ARIXI08 ARIXI09 ARIXI10 ARIXI12 ARIXI13 ARIXI14 ARIXI15 ARIXI18 ARIXI19 ARIXI20 ARIXI22 ARIXI23 ARIXI24 ARIXI25 ARIXI26 ARIXI27 ARIXI28 ARIXI29 ARIXI30 ARIXI31 ARIXI32 ARIXI33 ARIXOBY ARIXOB1 ARIXOB2 ARIXOCA ARIXOCK ARIXOCS ARIXOCT ARIXOCU ARIXOCY ARIXODF ARIXODM ARIXOD1 ARIXOD2 ARIXOEX ARIXOFC ARIXOFE ARIXOFF ARIXOFP ARIXOFQ ARIXOFR ARIXOFT ARIXOGA ARIXOGB ARIXOGC ARIXOGP ARIXOIN ARIXOIU ARIXOLC ARIXOLF ARIXOMA ARIXOMB ARIXOMD ARIXOMF ARIXOML ARIXOMS ARIXONV ARIXONW ARIXOOP ARIXOOS ARIXOQ1 ARIXOQ2 ARIXORM ARIXOSC ARIXOSG ARIXOSL ARIXOSO ARIXOSR ARIXOSS ARIXOST ARIXOSU ARIXOS3 ARIXOTF ARIXOTS ARIXOUS ARIXOVC ARIXOVD ARIXOV1 ARIXOW1 ARIXOW2 ARIXO1S ARIXPA0 ARIXPA1 ARIXPA2 ARIXPA3
ARIBKDM	ARIMPT1 ARIMPT2C ARIMPT2D ARIMPT3 ARIMPT4 ARIMPT5 ARIPPRA ARIPPRC ARIPPRP ARIXA01 ARIXA02 ARIXA03 ARIXA04 ARIXA07 ARIXA08 ARIXA09 ARIXA10 ARIXA11 ARIXEAD ARIXEDR ARIXEDS ARIXELX ARIXERD ARIXERP ARIXESX ARIXI01 ARIXI03 ARIXI04 ARIXI06 ARIXI07 ARIXI08 ARIXI09 ARIXI13 ARIXI15 ARIXI18 ARIXI19 ARIXI20 ARIXI22 ARIXI23 ARIXI24 ARIXI25 ARIXI26 ARIXI29 ARIXI30 ARIXI31 ARIXI33 ARIXOCA ARIXOD1 ARIXOD2 ARIXOIU ARIXOMS ARIXOUS ARIXOVC ARIXOVD

Data Area Macro Name	Referencing Modules
ARIBLCR	ARIMPT1 ARIMPT2C ARIMPT2D ARIMPT3 ARIMPT4 ARIMPT5 ARIPPRA ARIPPRC ARIPPRP ARISDFL ARIXA01 ARIXA02 ARIXA03 ARIXA0 ARIXA06 ARIXA07 ARIXA08 ARIXA09 ARIXA10 ARIXA11 ARIXC38 ARIXEAB ARIXEAD ARIXECK ARIXECL ARIXEDR ARIXEDS ARIXEFB ARIXELX ARIXEPH ARIXEPP ARIXERA ARIXERD ARIXERO ARIXERP ARIXESP ARIXEST ARIXESX ARIXIST ARIXI01 ARIXI03 ARIXI04 ARIXI06 ARIXI07 ARIXI08 ARIXI09 ARIXI12 ARIXI13 ARIXI15 ARIXI18 ARIXI19 ARIXI20 ARIXI22 ARIXI23 ARIXI24 ARIXI25 ARIXI26 ARIXI27 ARIXI28 ARIXI29 ARIXI30 ARIXI31 ARIXI32 ARIXI33 ARIXOB1 ARIXOB2 ARIXOCA ARIXOCK ARIXOCT ARIXOCU ARIXOCY ARIXODM ARIXOD1 ARIXOD2 ARIXOFF ARIXOFR ARIXOGA ARIXOGP ARIXOIN ARIXOIU ARIXOLF ARIXOMA ARIXOMF ARIXOML ARIXOMS ARIXONV ARIXOOP ARIXOSO ARIXOSR ARIXOSS ARIXOST ARIXOUS ARIXOVC ARIXOVD ARIXO1S
ARIBLIT	ARIMPT1 ARIMPT2C ARIMPT2D ARIMPT3 ARIMPT4 ARIMPT5 ARIPERR ARIPLTP ARIPMSH ARIPPRA ARIPPRC ARIPPRP ARIPSQA ARIPSQ ARIPTXA ARIPTXC ARIPTXP ARIXA01 ARIXA02 ARIXA03 ARIXA04 ARIXA06 ARIXA07 ARIXA08 ARIXA09 ARIXA10 ARIXA11 ARIXC01 ARIXC02 ARIXC03 ARIXC05 ARIXC06 ARIXC07 ARIXC08 ARIXC10 ARIXC11 ARIXC13 ARIXC14 ARIXC15 ARIXC16 ARIXC17 ARIXC18 ARIXC19 ARIXC20 ARIXC21 ARIXC22 ARIXC23 ARIXC24 ARIXC25 ARIXC27 ARIXC28 ARIXC29 ARIXC30 ARIXC31 ARIXC32 ARIXC33 ARIXC34 ARIXC35 ARIXC36 ARIXC37 ARIXC38 ARIXC39 ARIXC40 ARIXC41 ARIXC42 ARIXC43 ARIXC45 ARIXC48 ARIXC51 ARIXC54 ARIXEAB ARIXEAD ARIXEBR ARIXECK ARIXECL ARIXEDC ARIXEDP ARIXEDS ARIXEER ARIXEFB ARIXELX ARIXEPH ARIXEPP ARIXERA ARIXERD ARIXERO ARIXERP ARIXESP ARIXEST ARIXESX ARIXIST ARIXI01 ARIXI03 ARIXI04 ARIXI06 ARIXI07 ARIXI08 ARIXI09 ARIXI10 ARIXI12 ARIXI13 ARIXI14 ARIXI15 ARIXI18 ARIXI19 ARIXI20 ARIXI22 ARIXI23 ARIXI24 ARIXI25 ARIXI26 ARIXI27 ARIXI28 ARIXI29 ARIXI30 ARIXI31 ARIXI32 ARIXI33 ARIXOBY ARIXOB1 ARIXOB2 ARIXOCA ARIXOCK ARIXOCS ARIXOCT ARIXOCU ARIXOCY ARIXODF ARIXODM ARIXOD1 ARIXOD2 ARIXOEX ARIXOFC ARIXOFE ARIXOFF ARIXOFP ARIXOFQ ARIXOFR ARIXOFT ARIXOGA ARIXOGB ARIXOGC ARIXOGP ARIXOIN ARIXOIU ARIXOLC ARIXOLF ARIXOMA ARIXOMB ARIXOMD ARIXOMF ARIXOML ARIXOMS ARIXONV ARIXONW ARIXOOP ARIXOOS ARIXOQ1 ARIXOQ2 ARIXORM ARIXOSC ARIXOSG ARIXOSL ARIXOSO ARIXOSR ARIXOSS ARIXOST ARIXOSU ARIXOS3 ARIXOTF ARIXOTS ARIXOUS ARIXOVC ARIXOVD ARIXOV1 ARIXOW1 ARIXOW2 ARIXO1S ARIXPA0 ARIXPA1 ARIXPA2 ARIXPA3
ARIBMCR	ARIXCRA ARIMPT1 ARIMPT2C ARIMPT2D ARIMPT3 ARIMPT4 ARIMPT5 ARIPPRA ARIPPRC ARIPPRP ARISDFR ARIXA01 ARIXA02 ARIXA0 ARIXA04 ARIXA06 ARIXA07 ARIXA08 ARIXA09 ARIXA10 ARIXA11 ARIXC38 ARIXEAB ARIXEAD ARIXECK ARIXECL ARIXEDR ARIXEDS ARIXEFB ARIXELX ARIXEPH ARIXEPP ARIXERA ARIXERD ARIXERO ARIXERP ARIXESP ARIXEST ARIXESX ARIXIST ARIXI01 ARIXI03 ARIXI04 ARIXI06 ARIXI07 ARIXI08 ARIXI09 ARIXI12 ARIXI13 ARIXI15 ARIXI18 ARIXI19 ARIXI20 ARIXI22 ARIXI23 ARIXI24 ARIXI25 ARIXI26 ARIXI27 ARIXI28 ARIXI29 ARIXI30 ARIXI31 ARIXI32 ARIXI33 ARIXOB1 ARIXOB2 ARIXOCA ARIXOCK ARIXOCT ARIXOCU ARIXOCY ARIXODM ARIXOD1 ARIXOD2 ARIXOFF ARIXOFR ARIXOGA ARIXOGP ARIXOIN ARIXOIU ARIXOLF ARIXOMA ARIXOMF ARIXOML ARIXOMS ARIXONV ARIXOOP ARIXOSO ARIXOSR ARIXOSS ARIXOST ARIXOUS ARIXOVC ARIXOVD ARIXO1S
ARIBMP	ARIXOB2 ARIXOCS ARIXOCU ARIXOD1 ARIXOD2 ARIXOGA ARIXOGC ARIXOGP ARIXOMS ARIXOOP ARIXOSO ARIXOSR ARIXOTS
ARIBNLS	ARIPLTP ARIPPRA ARIPPRC ARIPPRP ARIPSQA ARIPSQC ARIPSQP ARIPTXA ARIPTXC ARIPTXP ARIXEAB ARIXEAD ARIXEDP ARIXEDS ARIXEFB ARIXELX ARIXEPP ARIXERD ARIXERP ARIXESX ARIXI09 ARIXOVC ARIXOVD
ARIBOP	ARIXF21 ARIXF47 ARIMPT1 ARIMPT2C ARIMPT2D ARIMPT3 ARIMPT4 ARIMPT5 ARIPERR ARIPLTP ARIPMSH ARIPPRA ARIPPRC ARIPPR ARIPSQA ARIPSQP ARIPTXA ARIPTXC ARIPTXP ARIXA01 ARIXA02 ARIXA03 ARIXA04 ARIXA06 ARIXA07 ARIXA08 ARIXA09 ARIXA10 ARIXA11 ARIXC01 ARIXC02 ARIXC03 ARIXC05 ARIXC06 ARIXC07 ARIXC08 ARIXC10 ARIXC11 ARIXC13 ARIXC14 ARIXC15 ARIXC16 ARIXC17 ARIXC18 ARIXC19 ARIXC20 ARIXC21 ARIXC22 ARIXC23 ARIXC24 ARIXC25 ARIXC27 ARIXC28 ARIXC29 ARIXC30 ARIXC31 ARIXC32 ARIXC33 ARIXC34 ARIXC35 ARIXC36 ARIXC37 ARIXC38 ARIXC39 ARIXC40 ARIXC41 ARIXC42 ARIXC43 ARIXC45 ARIXC48 ARIXC51 ARIXC54 ARIXEAB ARIXEAD ARIXEBR ARIXECK ARIXECL ARIXEDC ARIXEDP ARIXEDS ARIXEER ARIXEFB ARIXELX ARIXEPH ARIXEPP ARIXERA ARIXERD ARIXERO ARIXERP ARIXESP ARIXEST ARIXESX ARIXIST ARIXI01 ARIXI03 ARIXI04 ARIXI06 ARIXI07 ARIXI08 ARIXI09 ARIXI10 ARIXI12 ARIXI13 ARIXI14 ARIXI15 ARIXI18 ARIXI19 ARIXI20 ARIXI22 ARIXI23 ARIXI24 ARIXI25 ARIXI26 ARIXI27 ARIXI28 ARIXI29 ARIXI30 ARIXI31 ARIXI32 ARIXI33 ARIXOBY ARIXOB1 ARIXOB2 ARIXOCA ARIXOCK ARIXOCS ARIXOCT ARIXOCU ARIXOCY ARIXODF ARIXODM ARIXOD1 ARIXOD2 ARIXOEX ARIXOFC ARIXOFE ARIXOFF ARIXOFP ARIXOFQ ARIXOFR ARIXOFT ARIXOGA ARIXOGB ARIXOGC ARIXOGP ARIXOIN ARIXOIU ARIXOLC ARIXOLF ARIXOMA ARIXOMB ARIXOMD ARIXOMF ARIXOML ARIXOMS ARIXONV ARIXONW ARIXOOP ARIXOOS ARIXOQ1 ARIXOQ2 ARIXORM ARIXOSC ARIXOSG ARIXOSL ARIXOSO ARIXOSR ARIXOSS ARIXOST ARIXOSU ARIXOS3 ARIXOTF ARIXOTS ARIXOUS ARIXOVC ARIXOVD ARIXOV1 ARIXOW1 ARIXOW2 ARIXO1S ARIXPA0 ARIXPA1 ARIXPA2 ARIXPA3 ARIXPTT

Data Area Macro Name	Referencing Modules
ARIBOPT	ARIXIST ARIXI06 ARIXI09 ARIXI19 ARIXI20 ARIXI33 ARIXOAF ARIXOBY ARIXOB1 ARIXOB2 ARIXOCA ARIXOCK ARIXOCS ARIXOCT ARIXOCU ARIXOCY ARIXODF ARIXODM ARIXOD1 ARIXOD2 ARIXOEX ARIXOFC ARIXOFE ARIXOFF ARIXOFP ARIXOFQ ARIXOFR ARIXOFT ARIXOGA ARIXOGB ARIXOGC ARIXOGP ARIXOIN ARIXOIU ARIXOLC ARIXOLF ARIXOMA ARIXOMB ARIXOMD ARIXOMF ARIXOML ARIXOMS ARIXONV ARIXONW ARIXOOP ARIXOOS ARIXOQ1 ARIXOQ2 ARIXORM ARIXOSC ARIXOSG ARIXOSL ARIXOSO ARIXOSR ARIXOSS ARIXOST ARIXOSU ARIXOS1 ARIXOS2 ARIXOS3 ARIXOS4 ARIXOTF ARIXOTS ARIXOUS ARIXOVC ARIXOVD ARIXOV1 ARIXOW1 ARIXOW2 ARIXO1S ARIXTR1
ARIBOVI	ARIPLTP ARIPPRA ARIPPRC ARIPPRP ARIPTXA ARIPTXC ARIPTXP ARIXA01 ARIXA09 ARIXC03 ARIXC13 ARIXC18 ARIXEAB ARIXEAD ARIXEDP ARIXEDS ARIXEFB ARIXELX ARIXEPH ARIXEPP ARIXERA ARIXERD ARIXERP ARIXEST ARIXESX ARIXI10 ARIXI12 ARIXI30 ARIXOBY ARIXOMS ARIXOVC ARIXOVD ARIXPA0 ARIXPA1 ARIXPA2 ARIXPA3
ARIBOVN	ARIPLTP ARIPPRA ARIPPRC ARIPPRP ARIPTXA ARIPTXC ARIPTXP ARIXA01 ARIXA09 ARIXC03 ARIXC13 ARIXC18 ARIXEAB ARIXEAD ARIXEDP ARIXEDS ARIXEFB ARIXELX ARIXEPH ARIXEPP ARIXERA ARIXERD ARIXERP ARIXEST ARIXESX ARIXI10 ARIXI12 ARIXI30 ARIXOBY ARIXOMS ARIXOVC ARIXOVD ARIXPA0 ARIXPA1 ARIXPA2 ARIXPA3
ARIBPDA	ARIXI09 ARIXOAF ARIXOBY ARIXOB1 ARIXOB2 ARIXOCA ARIXOCK ARIXOCS ARIXOCT ARIXOCU ARIXOCY ARIXODF ARIXODM ARIXOD1 ARIXOD2 ARIXOEX ARIXOFC ARIXOFE ARIXOFF ARIXOFP ARIXOFQ ARIXOFR ARIXOFT ARIXOGA ARIXOGB ARIXOGC ARIXOGP ARIXOIN ARIXOIU ARIXOLC ARIXOLF ARIXOMA ARIXOMB ARIXOMD ARIXOMF ARIXOML ARIXOMS ARIXONV ARIXONW ARIXOOP ARIXOOS ARIXOQ1 ARIXOQ2 ARIXORM ARIXOSC ARIXOSG ARIXOSL ARIXOSO ARIXOSR ARIXOSS ARIXOST ARIXOSU ARIXOTF ARIXOTS ARIXOUS ARIXOVC ARIXOVD ARIXOV1 ARIXOW1 ARIXOW2 ARIXO1S ARIXTR1
ARIBPPO	ARIPLTP ARIPPRA ARIPPRC ARIPPRP ARIPSQA ARIPSQC ARIPSQP ARIPTXA ARIPTXC ARIPTXP ARIXEAB ARIXEAD ARIXEDP ARIXEDS ARIXEFB ARIXELX ARIXEPP ARIXERD ARIXERP ARIXESX ARIXI09 ARIXOVC ARIXOVD
ARIBPRO	ARIXCR4 ARIPLTP ARIPPRA ARIPPRC ARIPPRP ARIPTXA ARIPTXC ARIPTXP ARIXA01 ARIXA09 ARIXC03 ARIXC13 ARIXC18 ARIXEAB ARIXEAD ARIXEDP ARIXEDS ARIXEFB ARIXELX ARIXEPH ARIXEPP ARIXERA ARIXERD ARIXERO ARIXERP ARIXEST ARIXESX ARIXEUH ARIXI10 ARIXI12 ARIXI30 ARIXOBY ARIXOMS ARIXOVC ARIXOVD ARIXPA0 ARIXPA1 ARIXPA2 ARIXPA3
ARIBPV	ARIXOB2 ARIXOCS ARIXOCU ARIXOD1 ARIXOD2 ARIXOGA ARIXOGC ARIXOGP ARIXOMS ARIXOOP ARIXOSO ARIXOSR ARIXOTS
ARIBQAR	ARIXI09 ARIXI33 ARIXOAF ARIXOBY ARIXOB1 ARIXOB2 ARIXOCA ARIXOCK ARIXOCS ARIXOCT ARIXOCU ARIXOCY ARIXODF ARIXODM ARIXOD1 ARIXOD2 ARIXOEX ARIXOFC ARIXOFE ARIXOFF ARIXOFP ARIXOFQ ARIXOFR ARIXOFT ARIXOGA ARIXOGB ARIXOGC ARIXOGP ARIXOIN ARIXOIU ARIXOLC ARIXOLF ARIXOMA ARIXOMB ARIXOMD ARIXOMF ARIXOML ARIXOMS ARIXONV ARIXONW ARIXOOP ARIXOOS ARIXOQ1 ARIXOQ2 ARIXORM ARIXOSC ARIXOSG ARIXOSL ARIXOSO ARIXOSR ARIXOSS ARIXOST ARIXOSU ARIXOTF ARIXOTS ARIXOUS ARIXOVC ARIXOVD ARIXOV1 ARIXOW1 ARIXOW2 ARIXO1S ARIXTR1
ARIBQD	ARIMPT1 ARIMPT2C ARIMPT2D ARIMPT3 ARIMPT4 ARIMPT5 ARIPERR ARIPLTP ARIPMSH ARIPPRA ARIPPRC ARIPPRP ARIPSQA ARIPSQ ARIPTXA ARIPTXC ARIPTXP ARIXA01 ARIXA02 ARIXA03 ARIXA04 ARIXA06 ARIXA07 ARIXA08 ARIXA09 ARIXA10 ARIXA11 ARIXC01 ARIXC02 ARIXC03 ARIXC05 ARIXC06 ARIXC07 ARIXC08 ARIXC10 ARIXC11 ARIXC13 ARIXC14 ARIXC15 ARIXC16 ARIXC17 ARIXC18 ARIXC19 ARIXC20 ARIXC21 ARIXC22 ARIXC23 ARIXC24 ARIXC25 ARIXC27 ARIXC28 ARIXC29 ARIXC30 ARIXC31 ARIXC32 ARIXC33 ARIXC34 ARIXC35 ARIXC36 ARIXC37 ARIXC38 ARIXC39 ARIXC40 ARIXC41 ARIXC42 ARIXC43 ARIXC45 ARIXC48 ARIXC51 ARIXC54 ARIXEAB ARIXEAD ARIXEBR ARIXECK ARIXECL ARIXEDC ARIXEDP ARIXEDS ARIXEER ARIXEFB ARIXELX ARIXEPH ARIXEPP ARIXERA ARIXERD ARIXERO ARIXERP ARIXESP ARIXEST ARIXESX ARIXIST ARIXI01 ARIXI03 ARIXI04 ARIXI06 ARIXI07 ARIXI08 ARIXI09 ARIXI10 ARIXI12 ARIXI13 ARIXI14 ARIXI15 ARIXI18 ARIXI19 ARIXI20 ARIXI22 ARIXI23 ARIXI24 ARIXI25 ARIXI26 ARIXI27 ARIXI28 ARIXI29 ARIXI30 ARIXI31 ARIXI32 ARIXI33 ARIXOBY ARIXOB1 ARIXOB2 ARIXOCA ARIXOCK ARIXOCS ARIXOCT ARIXOCU ARIXOCY ARIXODF ARIXODM ARIXOD1 ARIXOD2 ARIXOEX ARIXOFC ARIXOFE ARIXOFF ARIXOFP ARIXOFQ ARIXOFR ARIXOFT ARIXOGA ARIXOGB ARIXOGC ARIXOGP ARIXOIN ARIXOIU ARIXOLC ARIXOLF ARIXOMA ARIXOMB ARIXOMD ARIXOMF ARIXOML ARIXOMS ARIXONV ARIXONW ARIXOOP ARIXOOS ARIXOQ1 ARIXOQ2 ARIXORM ARIXOSC ARIXOSG ARIXOSL ARIXOSO ARIXOSR ARIXOSS ARIXOST ARIXOSU ARIXOS3 ARIXOTF ARIXOTS ARIXOUS ARIXOVC ARIXOVD ARIXOV1 ARIXOW1 ARIXOW2 ARIXO1S ARIXPA0 ARIXPA1 ARIXPA2 ARIXPA3

Data Area Macro Name	Referencing Modules
ARIBRD	ARIMPT1 ARIMPT2C ARIMPT2D ARIMPT3 ARIMPT4 ARIMPT5 ARIPERR ARIPLTP ARIPMSH ARIPPRA ARIPPRC ARIPPRP ARIPSQA ARIPSQ ARIPTXA ARIPTXC ARIPTXP ARIXA01 ARIXA02 ARIXA03 ARIXA04 ARIXA06 ARIXA07 ARIXA08 ARIXA09 ARIXA10 ARIXA11 ARIXC01 ARIXC02 ARIXC03 ARIXC05 ARIXC06 ARIXC07 ARIXC08 ARIXC10 ARIXC11 ARIXC13 ARIXC14 ARIXC15 ARIXC16 ARIXC17 ARIXC18 ARIXC19 ARIXC20 ARIXC21 ARIXC22 ARIXC23 ARIXC24 ARIXC25 ARIXC27 ARIXC28 ARIXC29 ARIXC30 ARIXC31 ARIXC32 ARIXC33 ARIXC34 ARIXC35 ARIXC36 ARIXC37 ARIXC38 ARIXC39 ARIXC40 ARIXC41 ARIXC42 ARIXC43 ARIXC45 ARIXC48 ARIXC51 ARIXC54 ARIXEAB ARIXEAD ARIXEBR ARIXECK ARIXECL ARIXEDC ARIXEDP ARIXEDS ARIXEER ARIXEFB ARIXELX ARIXEPH ARIXEPP ARIXERA ARIXERD ARIXERO ARIXERP ARIXESP ARIXEST ARIXESX ARIXIST ARIXI01 ARIXI03 ARIXI04 ARIXI06 ARIXI07 ARIXI08 ARIXI09 ARIXI10 ARIXI12 ARIXI13 ARIXI14 ARIXI15 ARIXI18 ARIXI19 ARIXI20 ARIXI22 ARIXI23 ARIXI24 ARIXI25 ARIXI26 ARIXI27 ARIXI28 ARIXI29 ARIXI30 ARIXI31 ARIXI32 ARIXI33 ARIXOBY ARIXOB1 ARIXOB2 ARIXOCA ARIXOCK ARIXOCS ARIXOCT ARIXOCU ARIXOCY ARIXODF ARIXODM ARIXOD1 ARIXOD2 ARIXOEX ARIXOFC ARIXOFE ARIXOFF ARIXOFP ARIXOFQ ARIXOFR ARIXOFT ARIXOGA ARIXOGB ARIXOGC ARIXOGP ARIXOIN ARIXOIU ARIXOLC ARIXOLF ARIXOMA ARIXOMB ARIXOMD ARIXOMF ARIXOML ARIXOMS ARIXONV ARIXONW ARIXOOP ARIXOOS ARIXOQ1 ARIXOQ2 ARIXORM ARIXOSC ARIXOSG ARIXOSL ARIXOSO ARIXOSR ARIXOSS ARIXOST ARIXOSU ARIXOS3 ARIXOTF ARIXOTS ARIXOUS ARIXOVC ARIXOVD ARIXOV1 ARIXOW1 ARIXOW2 ARIXO1S ARIXPA0 ARIXPA1 ARIXPA2 ARIXPA3
ARIBRDI	ARIPRDID ARIRVSTC ARICIMB ARIIOCI ARIPLTP ARIPPRA ARIPPRC ARIPPRP ARIPSCN ARIPTXA ARIPTXC ARIPTXP ARIRBRM ARIRENAD ARIRIRM ARIRORMD ARIRSEND ARIRTL2C ARIRVIRC ARIRVRMC ARIXA01 ARIXA04 ARIXA09 ARIXC03 ARIXC13 ARIXC18 ARIXEAB ARIXEAD ARIXEDP ARIXEDR ARIXEDS ARIXEFB ARIXELX ARIXEOC ARIXEPH ARIXEPP ARIXERA ARIXERD ARIXERP ARIXEST ARIXESX ARIXIST ARIXI01 ARIXI06 ARIXI10 ARIXI12 ARIXI19 ARIXI30 ARIXOBY ARIXOMS ARIXOVC ARIXOVD ARIXPA0 ARIXPA1 ARIXPA2 ARIXPA3 ARIYE13
ARIBSAR	ARIMPT1 ARIMPT2C ARIMPT2D ARIMPT3 ARIMPT4 ARIMPT5 ARISUPD ARIXA02 ARIXA04 ARIXA08 ARIXA11 ARIXI01 ARIXI03 ARIXI1 ARIXI20 ARIXI25 ARIXI33 ARIXOIU ARIXO1S
ARIBSCR	ARIXCRA ARIMPT1 ARIMPT2C ARIMPT2D ARIMPT3 ARIMPT4 ARIMPT5 ARIPPRA ARIPPRC ARIPPRP ARISEGB ARISFIL ARISINI ARIXA0 ARIXA02 ARIXA03 ARIXA04 ARIXA06 ARIXA07 ARIXA08 ARIXA09 ARIXA10 ARIXA11 ARIXC38 ARIXEAB ARIXEAD ARIXECK ARIXECL ARIXEDR ARIXEDS ARIXEFB ARIXELX ARIXEPH ARIXEPP ARIXERA ARIXERD ARIXERO ARIXERP ARIXESP ARIXEST ARIXESX ARIXIST ARIXI01 ARIXI03 ARIXI04 ARIXI06 ARIXI07 ARIXI08 ARIXI09 ARIXI12 ARIXI13 ARIXI15 ARIXI18 ARIXI19 ARIXI20 ARIXI22 ARIXI23 ARIXI24 ARIXI25 ARIXI26 ARIXI27 ARIXI28 ARIXI29 ARIXI30 ARIXI31 ARIXI32 ARIXI33 ARIXOB1 ARIXOCA ARIXOCK ARIXOCT ARIXOCU ARIXOCY ARIXODM ARIXOD1 ARIXOD2 ARIXOFF ARIXOFR ARIXOGA ARIXOGP ARIXOIN ARIXOIU ARIXOLF ARIXOMA ARIXOMF ARIXOML ARIXOMS ARIXONV ARIXOOP ARIXOSO ARIXOSR ARIXOSS ARIXOST ARIXOUS ARIXOVC ARIXOVD ARIXO1S
ARIBSFN	ARIMPT1 ARIMPT2C ARIMPT2D ARIMPT3 ARIMPT4 ARIMPT5 ARIPERR ARIPLTP ARIPMSH ARIPPRA ARIPPRC ARIPPRP ARIPSQA ARIPSQ ARIPTXA ARIPTXC ARIPTXP ARIXA01 ARIXA02 ARIXA03 ARIXA04 ARIXA06 ARIXA07 ARIXA08 ARIXA09 ARIXA10 ARIXA11 ARIXC01 ARIXC02 ARIXC03 ARIXC05 ARIXC06 ARIXC07 ARIXC08 ARIXC10 ARIXC11 ARIXC13 ARIXC14 ARIXC15 ARIXC16 ARIXC17 ARIXC18 ARIXC19 ARIXC20 ARIXC21 ARIXC22 ARIXC23 ARIXC24 ARIXC25 ARIXC27 ARIXC28 ARIXC29 ARIXC30 ARIXC31 ARIXC32 ARIXC33 ARIXC34 ARIXC35 ARIXC36 ARIXC37 ARIXC38 ARIXC39 ARIXC40 ARIXC41 ARIXC42 ARIXC43 ARIXC45 ARIXC48 ARIXC51 ARIXC54 ARIXEAB ARIXEAD ARIXEBR ARIXECK ARIXECL ARIXEDC ARIXEDP ARIXEDS ARIXEER ARIXEFB ARIXELX ARIXEPH ARIXEPP ARIXERA ARIXERD ARIXERO ARIXERP ARIXESP ARIXEST ARIXESX ARIXIST ARIXI01 ARIXI03 ARIXI04 ARIXI06 ARIXI07 ARIXI08 ARIXI09 ARIXI10 ARIXI12 ARIXI13 ARIXI14 ARIXI15 ARIXI18 ARIXI19 ARIXI20 ARIXI22 ARIXI23 ARIXI24 ARIXI25 ARIXI26 ARIXI27 ARIXI28 ARIXI29 ARIXI30 ARIXI31 ARIXI32 ARIXI33 ARIXOBY ARIXOB1 ARIXOB2 ARIXOCA ARIXOCK ARIXOCS ARIXOCT ARIXOCU ARIXOCY ARIXODF ARIXODM ARIXOD1 ARIXOD2 ARIXOEX ARIXOFC ARIXOFE ARIXOFF ARIXOFP ARIXOFQ ARIXOFR ARIXOFT ARIXOGA ARIXOGB ARIXOGC ARIXOGP ARIXOIN ARIXOIU ARIXOLC ARIXOLF ARIXOMA ARIXOMB ARIXOMD ARIXOMF ARIXOML ARIXOMS ARIXONV ARIXONW ARIXOOP ARIXOOS ARIXOQ1 ARIXOQ2 ARIXORM ARIXOSC ARIXOSG ARIXOSL ARIXOSO ARIXOSR ARIXOSS ARIXOST ARIXOSU ARIXOS3 ARIXOTF ARIXOTS ARIXOUS ARIXOVC ARIXOVD ARIXOV1 ARIXOW1 ARIXOW2 ARIXO1S ARIXPA0 ARIXPA1 ARIXPA2 ARIXPA3
ARIBSQD	ARIMPT1 ARIMPT2C ARIMPT2D ARIMPT3 ARIMPT4 ARIMPT5 ARIPERR ARIPLTP ARIPMSH ARIPPRA ARIPPRC ARIPPRP ARIPSQA ARIPSQ ARIPTXA ARIPTXC ARIPTXP ARIXA01 ARIXA02 ARIXA03 ARIXA04 ARIXA06 ARIXA07 ARIXA08 ARIXA09 ARIXA10 ARIXA11 ARIXC01 ARIXC02 ARIXC03 ARIXC05 ARIXC06 ARIXC07 ARIXC08 ARIXC10 ARIXC11 ARIXC13 ARIXC14 ARIXC15 ARIXC16 ARIXC17 ARIXC18 ARIXC19 ARIXC20 ARIXC21 ARIXC22 ARIXC23 ARIXC24 ARIXC25 ARIXC27 ARIXC28 ARIXC29 ARIXC30 ARIXC31 ARIXC32 ARIXC33 ARIXC34 ARIXC35 ARIXC36 ARIXC37 ARIXC38 ARIXC39 ARIXC40 ARIXC41 ARIXC42 ARIXC43 ARIXC45 ARIXC48 ARIXC51 ARIXC54 ARIXEAB ARIXEAD ARIXEBR ARIXECK ARIXECL ARIXEDC ARIXEDP ARIXEDS ARIXEER ARIXEFB ARIXELX ARIXEPH ARIXEPP ARIXERA

Data Area Macro Name	Referencing Modules
	ARIXERD ARIXERO ARIXERP ARIXESP ARIXEST ARIXESX ARIXIST ARIXI01 ARIXI03 ARIXI04 ARIXI06 ARIXI07 ARIXI08 ARIXI09 ARIXI10 ARIXI12 ARIXI13 ARIXI14 ARIXI15 ARIXI18 ARIXI19 ARIXI20 ARIXI22 ARIXI23 ARIXI24 ARIXI25 ARIXI26 ARIXI27 ARIXI28 ARIXI29 ARIXI30 ARIXI31 ARIXI32 ARIXI33 ARIXOBY ARIXOB1 ARIXOB2 ARIXOCA ARIXOCK ARIXOCS ARIXOCT ARIXOCU ARIXOCY ARIXODF ARIXODM ARIXOD1 ARIXOD2 ARIXOEX ARIXOFC ARIXOFE ARIXOFF ARIXOFP ARIXOFQ ARIXOFR ARIXOFT ARIXOGA ARIXOGB ARIXOGC ARIXOGP ARIXOIN ARIXOIU ARIXOLC ARIXOLF ARIXOMA ARIXOMB ARIXOMD ARIXOMF ARIXOML ARIXOMS ARIXONV ARIXONW ARIXOOP ARIXOOS ARIXOQ1 ARIXOQ2 ARIXORM ARIXOSC ARIXOSG ARIXOSL ARIXOSO ARIXOSR ARIXOSS ARIXOST ARIXOSU ARIXOS3 ARIXOTF ARIXOTS ARIXOUS ARIXOVC ARIXOVD ARIXOV1 ARIXOW1 ARIXOW2 ARIXO1S ARIXPA0 ARIXPA1 ARIXPA2 ARIXPA3
ARIBSQL	ARIPRDID ARIRVSTC ARIXCRE ARIXCRH ARIXCR1 ARIXCR5 ARIXCR6 ARIXCR7 ARIXFA0 ARIXF01 ARIXF03 ARIXF04 ARIXF07 ARIXF1 ARIXF14 ARIXF15 ARIXF19 ARIXF20 ARIXF21 ARIXF24 ARIXF27 ARIXF28 ARIXF31 ARIXF42 ARIXF58 ARIXF64 ARIXF65 ARIXF70 ARIXF71 ARIXF75 ARIXF76 ARIXF77 ARIXF78 ARIXF80 ARIXF81 ARIXF82 ARIXF83 ARIXF84 ARIXF86 ARIXF87 ARIXF88 ARICABEC ARICABED ARICIP1C ARIMPT1 ARIMPT2C ARIMPT2D ARIMPT3 ARIMPT4 ARIMPT5 ARIPERR ARIPLTP ARIPMSH ARIPPRA ARIPPRC ARIPPRP ARIPSCN ARIPSQA ARIPSQC ARIPSQP ARIPTXA ARIPTXC ARIPTXP ARIXA01 ARIXA02 ARIXA03 ARIXA04 ARIXA06 ARIXA07 ARIXA08 ARIXA09 ARIXA10 ARIXA11 ARIXCR2 ARIXCR8 ARIXC01 ARIXC02 ARIXC03 ARIXC05 ARIXC06 ARIXC07 ARIXC08 ARIXC10 ARIXC11 ARIXC13 ARIXC14 ARIXC15 ARIXC16 ARIXC17 ARIXC18 ARIXC19 ARIXC20 ARIXC21 ARIXC22 ARIXC23 ARIXC24 ARIXC25 ARIXC27 ARIXC28 ARIXC29 ARIXC30 ARIXC31 ARIXC32 ARIXC33 ARIXC34 ARIXC35 ARIXC36 ARIXC37 ARIXC38 ARIXC39 ARIXC40 ARIXC41 ARIXC42 ARIXC43 ARIXC45 ARIXC48 ARIXC51 ARIXC54 ARIXEAB ARIXEAD ARIXEBR ARIXECK ARIXECL ARIXEDC ARIXEDP ARIXEDR ARIXEDS ARIXEER ARIXEFB ARIXELX ARIXEPH ARIXEPP ARIXERA ARIXERD ARIXERO ARIXERP ARIXESP ARIXEST ARIXESX ARIXIST ARIXI01 ARIXI03 ARIXI04 ARIXI06 ARIXI07 ARIXI08 ARIXI09 ARIXI10 ARIXI12 ARIXI13 ARIXI14 ARIXI15 ARIXI16 ARIXI18 ARIXI19 ARIXI20 ARIXI22 ARIXI23 ARIXI24 ARIXI25 ARIXI26 ARIXI27 ARIXI28 ARIXI29 ARIXI30 ARIXI31 ARIXI32 ARIXI33 ARIXOBY ARIXOB1 ARIXOB2 ARIXOCA ARIXOCK ARIXOCS ARIXOCT ARIXOCU ARIXOCY ARIXODF ARIXODM ARIXOD1 ARIXOD2 ARIXOEX ARIXOFC ARIXOFE ARIXOFF ARIXOFP ARIXOFQ ARIXOFR ARIXOFT ARIXOGA ARIXOGB ARIXOGC ARIXOGP ARIXOIN ARIXOIU ARIXOLC ARIXOLF ARIXOMA ARIXOMB ARIXOMD ARIXOMF ARIXOML ARIXOMS ARIXONV ARIXONW ARIXOOP ARIXOOS ARIXOQ1 ARIXOQ2 ARIXORM ARIXOSC ARIXOSG ARIXOSL ARIXOSO ARIXOSR ARIXOSS ARIXOST ARIXOSU ARIXOS2 ARIXOS3 ARIXOS4 ARIXOTF ARIXOTS ARIXOUS ARIXOVC ARIXOVD ARIXOV1 ARIXOW1 ARIXOW2 ARIXO1S ARIXPA0 ARIXPA1 ARIXPA2 ARIXPA3 ARIXSRT ARIXSUT
ARIBSTK	ARICFSE ARICGSE ARICIP1C ARICIPID ARICRST ARICSTK ARIISTK ARIRENTD ARIRIRM ARIRMI1D ARIRORMD ARIRVIRC ARIXERD
ARIBSTL	ARIPLTP ARIPPRA ARIPPRC ARIPPRP ARIPSQA ARIPSQC ARIPSQP ARIPTXA ARIPTXC ARIPTXP ARIXEAB ARIXEAD ARIXEDP ARIXEDR ARIXEDS ARIXEFB ARIXELX ARIXEPP ARIXERD ARIXERP ARIXESX ARIXI09 ARIXOVC ARIXOVD
ARIBTB	ARIXCRA ARIXCR1 ARIXF03 ARIXF15 ARIXF20 ARIXF21 ARIXF28 ARIXF30 ARIXF48 ARIXF71 ARIXF75 ARIXF77 ARIXF81 ARIXF82 ARIXF86 ARIXF87 ARIXF88 ARIMPT1 ARIMPT2C ARIMPT2D ARIMPT3 ARIMPT4 ARIMPT5 ARISCNV ARISEGB ARISFIL ARISIST ARISIST2 ARISLAY ARISUPD ARIXA01 ARIXA02 ARIXA03 ARIXA04 ARIXA06 ARIXA07 ARIXA08 ARIXA09 ARIXA10 ARIXA11 ARIXCRD ARIXCR2 ARIXCR3 ARIXEAD ARIXEBR ARIXECL ARIXEDR ARIXEDS ARIXELX ARIXEPH ARIXERD ARIXERO ARIXERP ARIXESX ARIXIST ARIXI01 ARIXI03 ARIXI04 ARIXI06 ARIXI07 ARIXI08 ARIXI09 ARIXI13 ARIXI15 ARIXI18 ARIXI19 ARIXI20 ARIXI22 ARIXI23 ARIXI24 ARIXI25 ARIXI26 ARIXI27 ARIXI28 ARIXI29 ARIXI30 ARIXI31 ARIXI32 ARIXI33 ARIXOCA ARIXOCK ARIXOCT ARIXOCU ARIXOCY ARIXOD1 ARIXOD2 ARIXOFF ARIXOFR ARIXOGA ARIXOGP ARIXOIN ARIXOIU ARIXOLF ARIXOMA ARIXOMF ARIXOMS ARIXONV ARIXOOP ARIXOSR ARIXOSS ARIXOST ARIXOUS ARIXOVC ARIXOVD
ARIBTBA	ARIXIST ARIXI06 ARIXI09 ARIXI19 ARIXI20 ARIXI33 ARIXOAF ARIXOBY ARIXOB1 ARIXOB2 ARIXOCA ARIXOCK ARIXOCS ARIXOCT ARIXOCU ARIXOCY ARIXODF ARIXODM ARIXOD1 ARIXOD2 ARIXOEX ARIXOFC ARIXOFE ARIXOFF ARIXOFP ARIXOFQ ARIXOFR ARIXOFT ARIXOGA ARIXOGB ARIXOGC ARIXOGP ARIXOIN ARIXOIU ARIXOLC ARIXOLF ARIXOMA ARIXOMB ARIXOMD ARIXOMF ARIXOML ARIXOMS ARIXONV ARIXONW ARIXOOP ARIXOOS ARIXOQ1 ARIXOQ2 ARIXORM ARIXOSC ARIXOSG ARIXOSL ARIXOSO ARIXOSR ARIXOSS ARIXOST ARIXOSU ARIXOTF ARIXOTS ARIXOUS ARIXOVC ARIXOVD ARIXOV1 ARIXOW1 ARIXOW2 ARIXO1S ARIXTR1
ARIBUSR	ARIPLTP ARIPPRA ARIPPRC ARIPPRP ARIPTXA ARIPTXC ARIPTXP ARIXA01 ARIXA09 ARIXC03 ARIXC13 ARIXC18 ARIXEAB ARIXEAD ARIXEDP ARIXEDS ARIXEFB ARIXELX ARIXEPH ARIXEPP ARIXERA ARIXERD ARIXERO ARIXERP ARIXEST ARIXESX ARIXEUH ARIXI10 ARIXI12 ARIXI30 ARIXOBY ARIXOMS ARIXOVC ARIXOVD ARIXPA0 ARIXPA1 ARIXPA2 ARIXPA3

Data Area Macro Name	Referencing Modules
ARICCPL	ARICCLAC ARICCLAD ARICCOMC ARICCOMD ARICCRAC ARICCRAD ARICDSPC ARICDSPD ARICENAD ARICINTC ARICIP1C ARICIP1D ARICIP2C ARICIP2D ARICMUDC ARICMUDD ARICOMB ARICSHT ARICTRM ARIRBRM ARIRDISD ARIRENAD ARIRINTC ARIRIRM ARIRORMD ARIRSEND ARIRTL2C ARIRVIRC ARIRVRMC ARISFM1C ARIYI22C ARIYI23C ARIYI47C
ARICDSC	ARICABEC ARICABED ARICCLAC ARICCLAD ARICCRAC ARICCRAD ARICDPT ARICDSPC ARICDSPD ARICENAC ARICENAD ARICINTC ARICIP1C ARICIP1D ARICIP2C ARICIP2D ARICMUDC ARICMUDD ARICOMB ARICPDM ARICPRM ARICRST ARICSHOC ARICSHT ARICSPMC ARICSPMD ARICSTK ARICTRM ARICWFR ARICWSF ARICWSG ARICWSI ARICWSR ARIRBRM ARIRVRMC ARISDBR ARISDSK ARISEGA ARISFDB ARISFM1C ARISFM1D ARISIIO ARIXCR2 ARIXEAD ARIXEBR ARIXECW ARIXEDB ARIXEDC ARIXEDS ARIXEER ARIXEOC ARIXERD ARIXEST ARIYC11 ARIYD55 ARIYD69 ARIYD81 ARIYE13 ARIYI14 ARIYI19 ARIYI20 ARIYI22C ARIYI22D ARIYI23C ARIYI29 ARIYI36 ARIYI47C ARIYK12 ARIYK19 ARIYK37 ARIYK41 ARIYL00 ARIYL01 ARIYL08 ARIYL13 ARIYL15 ARIYL21 ARIYL22 ARIYL23 ARIYM01 ARIYM02 ARIYM10 ARIYM11 ARIYM51C ARIYS08 ARIYT01 ARIYT05 ARIYT06 ARIYT12 ARIYT14 ARIYT16 ARIYT18 ARIYT19 ARIYT23 ARIYT26 ARIYT28 ARIYT29 ARIYX04 ARIYZ00 ARIYZ19 ARIYZ20
ARICEIB	ARICINTC ARIRINTC
ARICIIF	ARICIMB ARICMMB ARIRBRM ARIRENAD ARIRIRM ARIRORMD ARIRSEND ARIRTL1C ARIRVIRC ARIRVRMC
ARICNUC	ARICCOMC ARICIP1C ARICSPMC ARICTRM ARIRVIRC ARISDBKC ARISI5KC ARISYSDC ARISYSEC
ARICOHD	ARICMMB ARICMUDC ARICMUDD ARICOMB ARIRBRM ARIRORMD ARIRSEND ARIRVRMC
ARICOIF	ARICCRAC ARICCRAD ARICIP1C ARICIP1D ARICOMB ARIXCR2 ARIXEAD ARIXEDC ARIXEDS ARIXERD ARIYE13
ARICRDA	ARIXSDTD ARIXA06 ARIXA07 ARICABEC ARICABED ARICCRAC ARICCRAD ARICENAC ARICENAD ARICINTC ARICIP1C ARICIP1D ARICMUDC ARICMUDD ARICOMB ARICRST ARICTRC ARICWAT ARICWSF ARICWSG ARIMCID ARIMFLP ARIMPT1 ARIMPT2C ARIMPT2D ARIMPT3 ARIMPT4 ARIMPT5 ARIPCID ARIRBRM ARIRVRMC ARISMAI ARIXA01 ARIXA02 ARIXA03 ARIXA04 ARIXA08 ARIXA09 ARIXA10 ARIXA11 ARIXCRD ARIXCRG ARIXCRI ARIXCR2 ARIXCR3 ARIXCR8 ARIXC01 ARIXC02 ARIXC03 ARIXC05 ARIXC06 ARIXC07 ARIXC08 ARIXC10 ARIXC11 ARIXC13 ARIXC14 ARIXC15 ARIXC16 ARIXC17 ARIXC18 ARIXC19 ARIXC20 ARIXC21 ARIXC22 ARIXC23 ARIXC24 ARIXC25 ARIXC27 ARIXC28 ARIXC29 ARIXC30 ARIXC31 ARIXC32 ARIXC33 ARIXC34 ARIXC35 ARIXC36 ARIXC37 ARIXC38 ARIXC39 ARIXC40 ARIXC41 ARIXC42 ARIXC43 ARIXC45 ARIXC47 ARIXC48 ARIXC49 ARIXC51 ARIXC53 ARIXC54 ARIXEAB ARIXEAD ARIXEBR ARIXECK ARIXECL ARIXECW ARIXEDB ARIXEDC ARIXEDP ARIXEDR ARIXEDS ARIXEER ARIXEFB ARIXELK ARIXELX ARIXEOC ARIXEPH ARIXEPP ARIXERA ARIXERD ARIXERO ARIXERP ARIXESP ARIXEST ARIXESX ARIXETR ARIXEUH ARIXIST ARIXI01 ARIXI03 ARIXI04 ARIXI06 ARIXI07 ARIXI08 ARIXI09 ARIXI10 ARIXI12 ARIXI13 ARIXI14 ARIXI15 ARIXI16 ARIXI18 ARIXI19 ARIXI20 ARIXI22 ARIXI23 ARIXI24 ARIXI25 ARIXI26 ARIXI27 ARIXI28 ARIXI29 ARIXI30 ARIXI31 ARIXI32 ARIXI33 ARIXOAF ARIXOBY ARIXOB1 ARIXOB2 ARIXOCA ARIXOCK ARIXOCS ARIXOCT ARIXOCU ARIXOCY ARIXODF ARIXODM ARIXOD1 ARIXOD2 ARIXOEX ARIXOFC ARIXOFE ARIXOFF ARIXOFP ARIXOFQ ARIXOFR ARIXOFT ARIXOGA ARIXOGB ARIXOGC ARIXOGP ARIXOIN ARIXOIU ARIXOLC ARIXOLF ARIXOMA ARIXOMB ARIXOMD ARIXOMF ARIXOML ARIXOMS ARIXONV ARIXONW ARIXOOP ARIXOOS ARIXOQ1 ARIXOQ2 ARIXORM ARIXOSC ARIXOSG ARIXOSL ARIXOSO ARIXOSR ARIXOSS ARIXOST ARIXOSU ARIXOS1 ARIXOS2 ARIXOS3 ARIXOS4 ARIXOTF ARIXOTS ARIXOUS ARIXOVC ARIXOVD ARIXOV1 ARIXOW1 ARIXOW2 ARIXO1S ARIXPA0 ARIXPA1 ARIXPA2 ARIXPA3 ARIXPA4 ARIXSCF ARIXSCH ARIXSCN ARIXSCP ARIXSDB ARIXSLN ARIXSRT ARIXSUT ARIXTR1 ARIYL00 ARIYT09 ARIYT12
ARICRDC	ARIXA06 ARIXA07 ARICABEC ARICABED ARICCRAC ARICCRAD ARICIP1C ARICIP1D ARICIP2C ARICIP2D ARICPRM ARICSPMD ARICTRC ARICWSG ARIPLTP ARIPPRA ARIPPRC ARIPPRP ARIPTXA ARIPTXC ARIPTXP ARIRBRM ARIRVRMC ARIXA01 ARIXA02 ARIXA03 ARIXA04 ARIXA08 ARIXA09 ARIXA10 ARIXA11 ARIXEAD ARIXEBR ARIXECW ARIXEDB ARIXEDP ARIXEDS ARIXEER ARIXEFB ARIXELK ARIXELX ARIXEPH ARIXEPP ARIXERA ARIXERD ARIXERO ARIXERP ARIXESP ARIXEST ARIXESX ARIXETR ARIXEUH ARIXIST ARIXI01 ARIXI03 ARIXI04 ARIXI06 ARIXI07 ARIXI08 ARIXI09 ARIXI10 ARIXI13 ARIXI15 ARIXI18 ARIXI19 ARIXI20 ARIXI22 ARIXI23 ARIXI24 ARIXI25 ARIXI26 ARIXI27 ARIXI28 ARIXI29 ARIXI30 ARIXI31 ARIXI32 ARIXI33 ARIXOCA ARIXOCK ARIXOD1 ARIXOD2 ARIXOGC ARIXOGP ARIXOIU ARIXOMS ARIXOOP ARIXOSS ARIXOUS ARIXOVD ARIXPA0 ARIXPA1 ARIXPA2 ARIXPA3 ARIXPA4 ARIXSRT ARIYL00 ARIYT29
ARICTRO	ARIMTRA ARIXETR ARIYE14

Data Area Macro Name	Referencing Modules
ARICVMC	ARICCOMC ARICDSPC ARICENAC ARICINTC ARICIP1C ARICMUDC ARIRINTC ARISFM1C ARIYI07C ARIYI22C ARIYI23C ARIYI47C
ARICVMH	ARICCLAC ARICENAC ARICINTC ARICIP1C ARICSHOC ARICSHT ARIYI22C
ARICVMQ	ARICENAC ARICINTC ARICIP1C ARICSHOC ARISFM1C ARIYI22C
ARICWSA	ARICWSF ARICWSG ARICWSI ARIYT14 ARIYT19
ARIDCPA	ARIDALC ARIDALI ARIDALT ARIDBS ARIDCFI ARIDCIB ARIDCSP ARIDCTB ARIDDFI ARIDDLO ARIDDUL ARIDEXI ARIDMGE ARIDREL ARIDSEL ARIDSFA ARIDSSB ARIDUCP ARIDUNL ARIDUSQ
ARIGENIO	ARICIP1C ARICIP1D ARICTRC ARICTRI ARICTRM ARIDBS ARIDCFI ARIDDFI ARIDDLO ARIDDUL ARIDEXI ARIDMGE ARIDUCP ARIMPRT ARIMTRA ARIPLTP ARIPMSF ARIPMSH ARIPPRA ARIPPRC ARIPPRF ARIPPRP ARIPSCN ARIPSQA ARIPSQC ARIPSQF ARIPSQP ARIPTXA ARIPTXC ARIPTXF ARIPTXP ARISDBR ARISDSK ARISD00D ARISD10D ARISD11D ARISD12D ARISD13D ARISD14D ARISD15D ARISD16D ARISD20D ARISD21D ARISD22D ARISD23D ARISD24D ARISD25D ARISD26D ARISD31D ARISD32D ARISD33D ARISEGA ARISERR ARISMAI ARISPFM ARISYSDD ARISYSEC ARIXETR ARIYE14 ARIYL21 ARIYL22 ARIYL23
ARIIFMB	ARIIDQY ARIIFCC ARIIFCI ARIIFCS ARIIFET ARIIFMC ARIIFMT ARIIHDR ARIIHLP ARIIPQY ARIIPRTC ARIIPRTD ARIIRPT ARIISC
ARIIGCB	AXTCENT ARIIBIN ARIICAN ARIICICD ARIICI2D ARIICMD ARIICNV ARIIDBS ARIIDQY ARIIERS ARIIEXT ARIIFCC ARIIFCI ARIIFC ARIIFET ARIIFMC ARIIFMT ARIIFOL ARIIFOR ARIIFULD ARIIHDR ARIIHLD ARIIHLP ARIIIGN ARIILST ARIIMAP ARIIMSG ARIINITC ARIINME ARIINSQ ARIIOCI ARIIOVD ARIIPFKD ARIIPQL ARIIPQY ARIIPRF ARIIPRTC ARIIPRTD ARIIPSQ ARIIQRY ARIIREAC ARIIREC ARIIRETD ARIIRNM ARIIRPT ARIIRUN ARIIRWI ARIISCC ARIISCMD ARIISET ARIISMG ARIISTK ARIISTR ARIIST2 ARIIST3 ARIISUB ARIISYSC ARIITIOC ARIITKN ARIITRC ARIITRMD ARIIVFYC ARIIVFYD ARIIVLD ARIIWATD ARIIWRTD ARIIXITD
ARIIOCB	ARIICMD ARIIOCI
ARIIVAR	ARIIDQY ARIIFET ARIIFMT ARIIHDR ARIIPQL ARIIPQY ARIIPRTC ARIIPRTD ARIIPSQ ARIIRPT
ARIRECP	ARIRDISD ARIRECID ARIRENAD ARIRO13
ARIRMAR	ARIICI2D ARIINITC ARIIXITD ARIRDISD ARIRORMD ARIRSEND
ARIRMCV	ARIRDISD ARIRENAD ARIRMI1D ARIRORMD ARIRSEND
ARIRMGL	ARIRDISD ARIRENAD ARIRMI1D
ARIRMLA	ARIRDISD ARIRENAD ARIRORMD
ARIRMLO	ARIRBRM ARIRCL1C ARIRCL2C ARIRDISD ARIRECID ARIRENAD ARIRENTD ARIRIRM ARIRMI1D ARIRORMD ARIRO13 ARIRSEND ARIRTL1 ARIRTL2C ARIRVIRC ARIRVRMC
ARIRMLT	ARIRBRM ARIRCL1C ARIRDISD ARIRENAD ARIRIRM ARIRORMD ARIRSEND ARIRTL1C ARIRTL2C ARIRVIRC ARIRVRMC
ARIRMMR	ARIRENAD ARIRORMD
ARIRMRE	ARIRDISD ARIRENAD ARIRORMD ARIYT08 ARIYT26 ARIYT27
ARIRMSP	ARIRENAD ARIRORMD
ARIRMWL	ARIIXITD ARIRDISD ARIRENAD ARIRMI1D ARIRORMD ARIRSEND

Data Area Macro Name	Referencing Modules
ARIRMXCC	ARIIVFYC ARIRCL1C ARIRCL2C ARIRTL1C ARIRVIRC ARIRVRMC
ARIYOBA	ARIXEOC ARIYM10 ARIYT08
ARIYSCA	ARICCRAC ARICCRAD ARICIP1C ARICIP1D ARIYD00 ARIYD02 ARIYD05 ARIYD06 ARIYD08 ARIYD15 ARIYD18 ARIYD34 ARIYD44 ARIYD59 ARIYD60 ARIYD61 ARIYD87 ARIYL13 ARIYS00 ARIYT09 ARIYX00 ARIYX03 ARIYX04 ARIYX16 ARIYX19 ARIYX22 ARIYX28 ARIYZ15
DCE	ARICABEC ARICABED ARICCLAC ARICCLAD ARICCRAC ARICCRAD ARICDPT ARICDSPC ARICDSPD ARICDWT ARICENAC ARICENAD ARICINTC ARICIP1C ARICIP1D ARICMUDC ARICSHOC ARICSHT ARICWAT ARISOCMD ARIYT12 ARIYT14 ARIYT16 ARIYT18 ARIYT29
DS2CVT	ARICABEC ARICABED ARICCLAC ARICCLAD ARICCRAC ARICCRAD ARICDPT ARICDSPC ARICDSPD ARICDWT ARICENAC ARICENAD ARICINTC ARICIP1C ARICIP1D ARICIP2C ARICIP2D ARICMUDC ARICMUDD ARICONB ARICPDM ARICPRM ARICRST ARICSHOC ARICSHT ARICSPMC ARICSPMD ARICSTK ARICTRC ARICTRI ARICTRM ARICWAT ARICWFR ARICWSF ARICWSG ARICWSI ARICWSR ARIRBRM ARIRVRMC ARISDBR ARISDSK ARISEGA ARISFDB ARISFM1C ARISFM1D ARISIIO ARISOCMD ARISPFM ARISYSDC ARISYSDD ARIXEER ARIXELK ARIXEPH ARIXERD ARIXERO ARIXEST ARIXETR ARIXEUM ARIYC11 ARIYD69 ARIYD81 ARIYE01 ARIYE13 ARIYE14 ARIYI07C ARIYI22C ARIYI22D ARIYI23C ARIYI36 ARIYI47C ARIYK12 ARIYK37 ARIYK41 ARIYL00 ARIYL08 ARIYL13 ARIYL15 ARIYL21 ARIYL22 ARIYL23 ARIYM01 ARIYM02 ARIYM10 ARIYM11 ARIYM50D ARIYM51C ARIYM51D ARIYS08 ARIYT00 ARIYT01 ARIYT12 ARIYT14 ARIYT15 ARIYT18 ARIYT19 ARIYT26 ARIYT29 ARIYX04 ARIYZ00
DS2MODE	ARICIP1C ARICIP1D ARICIP2C ARICIP2D ARICMOD ARIRBRM ARIRCL1C ARIRIRM ARIRTL1C ARIRTL2C ARIRVIRC ARIRVRMC ARISOCM
SQLCA	ARICABEC ARICABED ARICIP1C ARIIBIN ARIICAN ARIICI2D ARIICMD ARIIDBS ARIIDQY ARIIERS ARIIFET ARIIFMT ARIIHDR ARIIHLP ARIILST ARIINITC ARIIOCI ARIIPQY ARIIPRTC ARIIPRTD ARIIPSQ ARIIREC ARIIRNM ARIIRUN ARIISCC ARIISMG ARIISTR ARIIST2 ARIMPT1 ARIMPT2C ARIMPT2D ARIMPT3 ARIMPT4 ARIMPT5 ARIMSMF ARIPERR ARIPLTP ARIPMSF ARIPMSH ARIPPRA ARIPPRC ARIPPRP ARIPSCN ARIPSQA ARIPSQC ARIPSQP ARIPTXA ARIPTXC ARIPTXP ARIRBRM ARIRENAD ARIRIRM ARIRORMD ARIRSEND ARIRTL2C ARIRVIRC ARIRVRMC ARIXA01 ARIXA02 ARIXA03 ARIXA04 ARIXA06 ARIXA07 ARIXA08 ARIXA09 ARIXA10 ARIXA11 ARIXCR2 ARIXCR8 ARIXC01 ARIXC02 ARIXC03 ARIXC05 ARIXC06 ARIXC07 ARIXC08 ARIXC10 ARIXC11 ARIXC13 ARIXC14 ARIXC15 ARIXC16 ARIXC17 ARIXC18 ARIXC19 ARIXC20 ARIXC21 ARIXC22 ARIXC23 ARIXC24 ARIXC25 ARIXC27 ARIXC28 ARIXC29 ARIXC30 ARIXC31 ARIXC32 ARIXC33 ARIXC34 ARIXC35 ARIXC36 ARIXC37 ARIXC38 ARIXC39 ARIXC40 ARIXC41 ARIXC42 ARIXC43 ARIXC45 ARIXC48 ARIXC51 ARIXC54 ARIXEAB ARIXEAD ARIXEBR ARIXECK ARIXECL ARIXEDC ARIXEDP ARIXEDR ARIXEDS ARIXEER ARIXEFB ARIXELX ARIXEOC ARIXEPH ARIXEPP ARIXERA ARIXERD ARIXERO ARIXERP ARIXESP ARIXEST ARIXESX ARIXIST ARIXI01 ARIXI03 ARIXI04 ARIXI06 ARIXI07 ARIXI08 ARIXI09 ARIXI10 ARIXI12 ARIXI13 ARIXI14 ARIXI15 ARIXI16 ARIXI18 ARIXI19 ARIXI20 ARIXI22 ARIXI23 ARIXI24 ARIXI25 ARIXI26 ARIXI27 ARIXI28 ARIXI29 ARIXI30 ARIXI31 ARIXI32 ARIXI33 ARIXOBY ARIXOB1 ARIXOB2 ARIXOCA ARIXOCK ARIXOCS ARIXOCT ARIXOCU ARIXOCY ARIXODF ARIXODM ARIXOD1 ARIXOD2 ARIXOEX ARIXOFC ARIXOFE ARIXOFF ARIXOFP ARIXOFQ ARIXOFR ARIXOFT ARIXOGA ARIXOGB ARIXOGC ARIXOGP ARIXOIN ARIXOIU ARIXOLC ARIXOLF ARIXOMA ARIXOMB ARIXOMD ARIXOMF ARIXOML ARIXOMS ARIXONV ARIXONW ARIXOOP ARIXOOS ARIXOQ1 ARIXOQ2 ARIXORM ARIXOSC ARIXOSG ARIXOSL ARIXOSO ARIXOSR ARIXOSS ARIXOST ARIXOSU ARIXOS2 ARIXOS3 ARIXOS4 ARIXOTF ARIXOTS ARIXOUS ARIXOVC ARIXOVD ARIXOV1 ARIXOW1 ARIXOW2 ARIXO1S ARIXPA0 ARIXPA1 ARIXPA2 ARIXPA3 ARIXSRT ARIXSUT
SQLDA	ARICABEC ARICABED ARICIMB ARICIP1C ARIDCIB ARIDCTB ARIDDLO ARIDDUL ARIDREL ARIDSEL ARIDSFA ARIDUNL ARIIBIN ARIIDQY ARIIFET ARIIFMT ARIIHDR ARIIPQY ARIIPRTC ARIIPRTD ARIIRPT ARIIRUN ARIMPT1 ARIMPT2C ARIMPT2D ARIMPT3 ARIMPT4 ARIMPT5 ARIPERR ARIPLTP ARIPMSH ARIPPRA ARIPPRC ARIPPRF ARIPPRP ARIPSCN ARIPSQA ARIPSQC ARIPSQP ARIPSSF ARIPTXA ARIPTXC ARIPTXP ARIXA01 ARIXA02 ARIXA03 ARIXA04 ARIXA06 ARIXA07 ARIXA08 ARIXA09 ARIXA10 ARIXA11 ARIXCR2 ARIXCR8 ARIXC01 ARIXC02 ARIXC03 ARIXC05 ARIXC06 ARIXC07 ARIXC08 ARIXC10 ARIXC11 ARIXC13 ARIXC14 ARIXC15 ARIXC16 ARIXC17 ARIXC18 ARIXC19 ARIXC20 ARIXC21 ARIXC22 ARIXC23 ARIXC24 ARIXC25 ARIXC27 ARIXC28 ARIXC29 ARIXC30 ARIXC31 ARIXC32 ARIXC33 ARIXC34 ARIXC35 ARIXC36 ARIXC37 ARIXC38 ARIXC39 ARIXC40 ARIXC41 ARIXC42 ARIXC43 ARIXC45 ARIXC48 ARIXC51 ARIXC54 ARIXEAB ARIXEAD ARIXEBR ARIXECK ARIXECL ARIXEDC ARIXEDP ARIXEDR ARIXEDS ARIXEER ARIXEFB ARIXELX ARIXEOC ARIXEPH ARIXEPP ARIXERA ARIXERD ARIXERO ARIXERP ARIXESP ARIXEST ARIXESX ARIXIST ARIXI01 ARIXI03 ARIXI04 ARIXI06 ARIXI07 ARIXI08 ARIXI09 ARIXI10 ARIXI12 ARIXI13 ARIXI14 ARIXI15 ARIXI16 ARIXI18 ARIXI19 ARIXI20 ARIXI22 ARIXI23 ARIXI24 ARIXI25 ARIXI26 ARIXI27 ARIXI28 ARIXI29 ARIXI30 ARIXI31 ARIXI32 ARIXI33 ARIXOBY ARIXOB1 ARIXOB2

Data Area Macro Name	Referencing Modules
	ARIXOCA ARIXOCK ARIXOCS ARIXOCT ARIXOCU ARIXOCY ARIXODF ARIXODM ARIXOD1 ARIXOD2 ARIXOEX ARIXOFC ARIXOFE ARIXOFF ARIXOFP ARIXOFQ ARIXOFR ARIXOFT ARIXOGA ARIXOGB ARIXOGC ARIXOGP ARIXOIN ARIXOIU ARIXOLC ARIXOLF ARIXOMA ARIXOMB ARIXOMD ARIXOMF ARIXOML ARIXOMS ARIXONV ARIXONW ARIXOOP ARIXOOS ARIXOQ1 ARIXOQ2 ARIXORM ARIXOSC ARIXOSG ARIXOSL ARIXOSO ARIXOSR ARIXOSS ARIXOST ARIXOSU ARIXOS2 ARIXOS3 ARIXOS4 ARIXOTF ARIXOTS ARIXOUS ARIXOVC ARIXOVD ARIXOV1 ARIXOW1 ARIXOW2 ARIXO1S ARIXPA0 ARIXPA1 ARIXPA2 ARIXPA3 ARIXSRT ARIXSUT
TRACMAP	ARIXETR ARIYE14
YCONTROL	ARIYC00 ARIYC01 ARIYC02 ARIYC03 ARIYC05 ARIYC10 ARIYC11 ARIYC13 ARIYC14 ARIYC15 ARIYC18 ARIYC19 ARIYC20 ARIYC21 ARIYC22 ARIYC23 ARIYC24 ARIYC25 ARIYC26 ARIYD01 ARIYD02 ARIYD03 ARIYD04 ARIYD05 ARIYD06 ARIYD08 ARIYD09 ARIYD10 ARIYD18 ARIYD27 ARIYD28 ARIYD29 ARIYD30 ARIYD32 ARIYD33 ARIYD38 ARIYD45 ARIYD46 ARIYD49 ARIYD57 ARIYD58 ARIYD69 ARIYD74 ARIYD75 ARIYI02 ARIYK38 ARIYK39 ARIYK40 ARIYS00 ARIYS01 ARIYS02 ARIYS03 ARIYS04 ARIYS05 ARIYS08 ARIYS11 ARIYS12 ARIYS14 ARIYS17 ARIYS21 ARIYS24 ARIYX01 ARIYX02 ARIYX03 ARIYX04 ARIYX05 ARIYX06 ARIYX07 ARIYX08 ARIYX09 ARIYX10 ARIYX12 ARIYX14 ARIYX15 ARIYX18 ARIYX20 ARIYX21 ARIYX23 ARIYX24 ARIYX25 ARIYX26 ARIYX27 ARIYX29 ARIYX30 ARIYZ03 ARIYZ04 ARIYZ05 ARIYZ16
YCUSER	ARIYC00 ARIYC01 ARIYC02 ARIYC03 ARIYC05 ARIYC06 ARIYC07 ARIYC08 ARIYC10 ARIYC11 ARIYC13 ARIYC14 ARIYC27 ARIYK39 ARIYX04
YIPAGE	ARIYI02 ARIYX01 ARIYX02 ARIYX03 ARIYX04 ARIYX05 ARIYX07 ARIYX08 ARIYX09 ARIYX10 ARIYX11 ARIYX12 ARIYX14 ARIYX15 ARIYX16 ARIYX18 ARIYX20 ARIYX21 ARIYX22 ARIYX23 ARIYX24 ARIYX25 ARIYX26 ARIYX27 ARIYX28 ARIYX29 ARIYX30 ARIYZ16
YLOG	ARIYC02 ARIYC05 ARIYC10 ARIYC11 ARIYC13 ARIYC14 ARIYC18 ARIYC19 ARIYC20 ARIYC21 ARIYC22 ARIYC23 ARIYC24 ARIYC25 ARIYC26 ARIYD01 ARIYD02 ARIYD03 ARIYD05 ARIYD10 ARIYD33 ARIYD65 ARIYD66 ARIYD67 ARIYD68 ARIYD69 ARIYD74 ARIYL07 ARIYS00 ARIYS02 ARIYS08
YLOGTMPL	ARIYL00 ARIYL01 ARIYL02 ARIYL03 ARIYL04 ARIYL05 ARIYL06 ARIYL08 ARIYL09 ARIYL10 ARIYL11 ARIYL12 ARIYL13 ARIYL19 ARIYT09 ARIYT10 ARIYT14 ARIYT19 ARIYT20 ARIYT21 ARIYT27 ARIYT29
YLRBS	ARICIP1C ARICIP1D ARIYI05 ARIYK00 ARIYK02 ARIYK09 ARIYK12 ARIYK21 ARIYK22 ARIYK23 ARIYK24 ARIYK25 ARIYK26 ARIYK2 ARIYK41 ARIYK42 ARIYT01 ARIYT10 ARIYT12 ARIYT14 ARIYT20 ARIYT21 ARIYT27
YRSSCVT	ARISIST2 ARICABEC ARICABED ARICCLAC ARICCLAD ARICCRAC ARICCRAD ARICDPT ARICDSPC ARICDSPD ARICDWT ARICINTC ARICIP1C ARICIP1D ARICIP2C ARICIP2D ARICOMB ARICPRM ARICRST ARICSHT ARICSPMC ARICSPMD ARICTRC ARICTRI ARICTRM ARICWAT ARIRBRM ARIRVRMC ARISDFL ARISDFM ARISDFR ARISDSK ARISEGA ARISEGB ARISFDB ARISFIL ARISFM1C ARISFM1D ARISFM2 ARISIIO ARISINI ARISIST ARISLAY ARISMAI ARISPFM ARISSCB ARISUPD ARISYSDD ARIXERD ARIXERO ARIXEST ARIXETR ARIYC00 ARIYC01 ARIYC02 ARIYC03 ARIYC05 ARIYC06 ARIYC07 ARIYC08 ARIYC09 ARIYC10 ARIYC11 ARIYC13 ARIYC14 ARIYC15 ARIYC18 ARIYC19 ARIYC20 ARIYC21 ARIYC22 ARIYC23 ARIYC24 ARIYC25 ARIYC26 ARIYD00 ARIYD01 ARIYD02 ARIYD03 ARIYD04 ARIYD05 ARIYD06 ARIYD08 ARIYD09 ARIYD10 ARIYD11 ARIYD33 ARIYD48 ARIYD50 ARIYD52 ARIYD53 ARIYD55 ARIYD76 ARIYD77 ARIYD87 ARIYE01 ARIYE14 ARIYI00 ARIYI01D ARIYI02 ARIYI03 ARIYI05 ARIYI06 ARIYI07C ARIYI07D ARIYI09 ARIYI10 ARIYI13 ARIYI14 ARIYI15 ARIYI17 ARIYI19 ARIYI20 ARIYI21 ARIYI22C ARIYI22D ARIYI23C ARIYI23D ARIYI24 ARIYI26 ARIYI28 ARIYI29 ARIYI31 ARIYI36 ARIYI41 ARIYI42 ARIYI45 ARIYI46 ARIYI47C ARIYI47D ARIYI48 ARIYI49 ARIYI50D ARIYI51 ARIYK00 ARIYK01 ARIYK02 ARIYK03 ARIYK04 ARIYK05 ARIYK06 ARIYK07 ARIYK08 ARIYK09 ARIYK10 ARIYK11 ARIYK12 ARIYK18 ARIYK19 ARIYK21 ARIYK22 ARIYK23 ARIYK24 ARIYK25 ARIYK26 ARIYK27 ARIYK28 ARIYK37 ARIYK38 ARIYK39 ARIYK40 ARIYK41 ARIYK42 ARIYK43 ARIYL00 ARIYL01 ARIYL02 ARIYL03 ARIYL04 ARIYL05 ARIYL06 ARIYL07 ARIYL08 ARIYL09 ARIYL10 ARIYL11 ARIYL12 ARIYL13 ARIYL14 ARIYL15 ARIYL19 ARIYL21 ARIYL22 ARIYL23 ARIYL24 ARIYL25 ARIYM00 ARIYM50D ARIYM51D ARIYS00 ARIYS01 ARIYS02 ARIYS03 ARIYS04 ARIYS05 ARIYS08 ARIYS09 ARIYS10 ARIYS11 ARIYS12 ARIYS14 ARIYS17 ARIYS18 ARIYS19 ARIYS20 ARIYS21 ARIYS22 ARIYS23 ARIYS24 ARIYS25 ARIYT00 ARIYT01 ARIYT02 ARIYT05 ARIYT06 ARIYT09 ARIYT10 ARIYT12 ARIYT14 ARIYT15 ARIYT16 ARIYT17 ARIYT18 ARIYT19 ARIYT20 ARIYT21 ARIYT22 ARIYT23 ARIYT26 ARIYT27 ARIYT28 ARIYT29 ARIYX00 ARIYX01 ARIYX02 ARIYX03 ARIYX04 ARIYX05 ARIYX06 ARIYX07 ARIYX08 ARIYX09 ARIYX10 ARIYX11 ARIYX12 ARIYX14 ARIYX15 ARIYX16 ARIYX17 ARIYX18 ARIYX19 ARIYX20 ARIYX21 ARIYX22 ARIYX23 ARIYX24 ARIYX25 ARIYX26 ARIYX27

Data Area Macro Name	Referencing Modules
	ARIYX28 ARIYX29 ARIYX30 ARIYZTP ARIYZ00 ARIYZ01 ARIYZ02 ARIYZ03 ARIYZ04 ARIYZ05 ARIYZ06 ARIYZ10 ARIYZ11 ARIYZ15 ARIYZ16 ARIYZ19 ARIYZ20 ARIYZ23 ARIYZ25 ARIYZ26 ARIYZ27
YRSSTRAN	ARICRST ARIYM00 ARIYT06 ARIYT08 ARIYT09 ARIYT10 ARIYT14 ARIYT15 ARIYT17 ARIYT26 ARIYT27 ARIYT28
YSCAN	ARIYD00 ARIYD01 ARIYD02 ARIYD03 ARIYD05 ARIYD06 ARIYD08 ARIYD09 ARIYD10 ARIYD15 ARIYD18 ARIYD34 ARIYD44 ARIYD59 ARIYD60 ARIYD61 ARIYD87 ARIYSTP ARIYS00 ARIYS24 ARIYX00 ARIYX03 ARIYX04 ARIYX05 ARIYX12 ARIYX16 ARIYX19 ARIYX20 ARIYX22 ARIYX23 ARIYX28 ARIYZ15
YSRTBASE	ARIYS00
YTAB	ARICCRAC ARICCRAD ARICIP1C ARICIP1D ARISDSK ARISEGA ARISFDB ARISFM1C ARISFM1D ARISFM2 ARISIIO ARISPFM ARISSCB ARIYC00 ARIYD06 ARIYD48 ARIYD50 ARIYD52 ARIYD53 ARIYD55 ARIYD76 ARIYD77 ARIYI00 ARIYI01D ARIYI02 ARIYI05 ARIYI06 ARIYI07C ARIYI07D ARIYI09 ARIYI10 ARIYI13 ARIYI14 ARIYI15 ARIYI17 ARIYI19 ARIYI20 ARIYI21 ARIYI22C ARIYI22D ARIYI23C ARIYI23D ARIYI24 ARIYI26 ARIYI29 ARIYI31 ARIYI36 ARIYI41 ARIYI42 ARIYI45 ARIYI46 ARIYI47C ARIYI47D ARIYI48 ARIYI49 ARIYI51 ARIYK26 ARIYL00 ARIYL09 ARIYL14 ARIYL15 ARIYL19 ARIYL21 ARIYL22 ARIYL23 ARIYT02 ARIYT14 ARIYX01 ARIYX03 ARIYX04 ARIYX09 ARIYX12 ARIYX18 ARIYX21 ARIYZ04 ARIYZ10 ARIYZ11 ARIYZ16
YTABLE1	ARICABEC ARICABED ARICCLAC ARICCLAD ARICCRAC ARICCRAD ARICDPT ARICDSPC ARICDSPD ARICDWT ARICIP1C ARICIP1D ARICIP2C ARICIP2D ARICOMB ARICPRM ARICRST ARICSHT ARICSPMC ARICTRC ARICTRI ARICTRM ARICWAT ARICWSF ARICWSG ARIRBRM ARIRVRMC ARISDBR ARISDFL ARISDFM ARISDFR ARISDSK ARISEGA ARISEGB ARISFDB ARISFIL ARISFM1C ARISFM1D ARISFM2 ARISIIO ARISINI ARISIST ARISIST2 ARISLAY ARISMAI ARISPFM ARISSCB ARISUPD ARIXECW ARIXEDB ARIXETR ARIYC00 ARIYC01 ARIYC02 ARIYC03 ARIYC05 ARIYC06 ARIYC07 ARIYC08 ARIYC09 ARIYC10 ARIYC11 ARIYC13 ARIYC14 ARIYC15 ARIYC18 ARIYC19 ARIYC20 ARIYC21 ARIYC22 ARIYC23 ARIYC24 ARIYC25 ARIYC26 ARIYC27 ARIYD00 ARIYD01 ARIYD02 ARIYD03 ARIYD04 ARIYD05 ARIYD06 ARIYD08 ARIYD09 ARIYD10 ARIYD11 ARIYD13 ARIYD14 ARIYD15 ARIYD17 ARIYD18 ARIYD19 ARIYD20 ARIYD23 ARIYD26 ARIYD27 ARIYD28 ARIYD29 ARIYD30 ARIYD32 ARIYD33 ARIYD34 ARIYD38 ARIYD39 ARIYD40 ARIYD41 ARIYD43 ARIYD44 ARIYD45 ARIYD46 ARIYD47 ARIYD48 ARIYD49 ARIYD50 ARIYD52 ARIYD53 ARIYD55 ARIYD57 ARIYD58 ARIYD59 ARIYD60 ARIYD61 ARIYD62 ARIYD63 ARIYD64 ARIYD65 ARIYD66 ARIYD67 ARIYD68 ARIYD69 ARIYD70 ARIYD71 ARIYD72 ARIYD73 ARIYD74 ARIYD75 ARIYD76 ARIYD77 ARIYD80 ARIYD81 ARIYD87 ARIYE01 ARIYE13 ARIYE14 ARIYE15 ARIYI00 ARIYI01D ARIYI02 ARIYI03 ARIYI04 ARIYI05 ARIYI06 ARIYI07C ARIYI07D ARIYI09 ARIYI10 ARIYI13 ARIYI14 ARIYI15 ARIYI17 ARIYI19 ARIYI20 ARIYI21 ARIYI22C ARIYI22D ARIYI23C ARIYI23D ARIYI24 ARIYI26 ARIYI28 ARIYI29 ARIYI31 ARIYI36 ARIYI41 ARIYI42 ARIYI45 ARIYI46 ARIYI47C ARIYI47D ARIYI48 ARIYI49 ARIYI50D ARIYI51 ARIYK00 ARIYK01 ARIYK02 ARIYK03 ARIYK04 ARIYK05 ARIYK06 ARIYK07 ARIYK08 ARIYK09 ARIYK10 ARIYK11 ARIYK12 ARIYK18 ARIYK19 ARIYK21 ARIYK22 ARIYK23 ARIYK24 ARIYK25 ARIYK26 ARIYK27 ARIYK28 ARIYK37 ARIYK38 ARIYK39 ARIYK40 ARIYK41 ARIYK42 ARIYK43 ARIYL00 ARIYL01 ARIYL02 ARIYL03 ARIYL04 ARIYL05 ARIYL06 ARIYL07 ARIYL08 ARIYL09 ARIYL10 ARIYL11 ARIYL12 ARIYL13 ARIYL14 ARIYL15 ARIYL19 ARIYL21 ARIYL22 ARIYL23 ARIYL24 ARIYL25 ARIYM00 ARIYM01 ARIYM02 ARIYM10 ARIYM11 ARIYM50D ARIYM51D ARIYSTP ARIYS00 ARIYS01 ARIYS02 ARIYS03 ARIYS04 ARIYS05 ARIYS08 ARIYS09 ARIYS10 ARIYS11 ARIYS12 ARIYS14 ARIYS17 ARIYS18 ARIYS19 ARIYS20 ARIYS21 ARIYS22 ARIYS23 ARIYS24 ARIYS25 ARIYT00 ARIYT01 ARIYT02 ARIYT05 ARIYT06 ARIYT09 ARIYT10 ARIYT12 ARIYT14 ARIYT15 ARIYT16 ARIYT17 ARIYT18 ARIYT19 ARIYT20 ARIYT21 ARIYT22 ARIYT23 ARIYT26 ARIYT27 ARIYT28 ARIYT29 ARIYXTP ARIYX00 ARIYX01 ARIYX02 ARIYX03 ARIYX04 ARIYX05 ARIYX06 ARIYX07 ARIYX08 ARIYX09 ARIYX10 ARIYX11 ARIYX12 ARIYX14 ARIYX15 ARIYX16 ARIYX17 ARIYX18 ARIYX19 ARIYX20 ARIYX21 ARIYX22 ARIYX23 ARIYX24 ARIYX25 ARIYX26 ARIYX27 ARIYX28 ARIYX29 ARIYX30 ARIYZTP ARIYZ00 ARIYZ01 ARIYZ02 ARIYZ03 ARIYZ04 ARIYZ05 ARIYZ06 ARIYZ10 ARIYZ11 ARIYZ15 ARIYZ16 ARIYZ19 ARIYZ20 ARIYZ23 ARIYZ25 ARIYZ26 ARIYZ27
YTABLE1S	ARICCOMC ARICCOMD ARICDMP ARICENAC ARICENAD ARICFSE ARICGSE ARICIMB ARICINTC ARICMMB ARICMUDC ARICMUDD ARICPDM ARICSHOC ARICSPMD ARICSTK ARICTIM ARICTKN ARICWFR ARICWSI ARICWSR ARIRB13 ARIRB51 ARIRCL2C ARIRDISD ARIRECID ARIRENAD ARIRORMD ARIRQ13 ARIRSEND ARIRTL2C ARISTIM ARISYSDD ARISYSGC ARIXERD ARIXEST ARIYD16 ARIYE02 ARIYE04 ARIYE05 ARIYE06 ARIYE07 ARIYE08 ARIYE09 ARIYE10 ARIYE11 ARIYE16 ARIYE37 ARIYM03 ARIYM04 ARIYM05 ARIYM50C ARIYM51C ARIYM52C ARIYM52D ARIYT08

Data Area Macro Name	Referencing Modules
YTABLE1U	ARIYC18 ARIYC19 ARIYC20 ARIYC21 ARIYC22 ARIYC23 ARIYC24 ARIYC25 ARIYC26 ARIYD65 ARIYD66 ARIYD67 ARIYD68 ARIYD69 ARIYD70 ARIYD71 ARIYD72 ARIYD73 ARIYD74 ARIYD75 ARIYD76 ARIYD77 ARIYD80 ARIYS02 ARIYS08
YTPMAP	ARICCLAC ARICCLAD ARICCRAC ARICDPT ARICDSPC ARICDSPD ARICDWT ARICENAC ARICINTC ARICIP1C ARICIP1D ARICOMB ARICPRM ARICSHT ARICTRC ARICTRM ARICWAT ARIRBRM ARIRORMD ARIRVRMC ARIXEDB ARIXEPH ARIXERD ARIXETR ARIXEUH ARIYD11 ARIYD80 ARIYE01 ARIYE14 ARIYE15 ARIYI05 ARIYI10 ARIYI14 ARIYI19 ARIYI29 ARIYI36 ARIYI50D ARIYK00 ARIYK02 ARIYK03 ARIYK04 ARIYK05 ARIYK06 ARIYK07 ARIYK08 ARIYK09 ARIYK10 ARIYK11 ARIYK12 ARIYK18 ARIYK19 ARIYK21 ARIYK22 ARIYK23 ARIYK24 ARIYK25 ARIYK26 ARIYK27 ARIYK37 ARIYK41 ARIYK42 ARIYL00 ARIYL01 ARIYL02 ARIYL03 ARIYL04 ARIYL05 ARIYL06 ARIYL07 ARIYL08 ARIYL09 ARIYL10 ARIYL13 ARIYL15 ARIYL19 ARIYL25 ARIYM00 ARIYM50D ARIYM51D ARIYT00 ARIYT01 ARIYT02 ARIYT05 ARIYT06 ARIYT09 ARIYT10 ARIYT12 ARIYT14 ARIYT15 ARIYT16 ARIYT17 ARIYT18 ARIYT19 ARIYT20 ARIYT21 ARIYT22 ARIYT23 ARIYT26 ARIYT27 ARIYT28 ARIYT29 ARIYZTP ARIYZ19
YUSER	ARIYD00 ARIYD01 ARIYD02 ARIYD03 ARIYD04 ARIYD05 ARIYD06 ARIYD08 ARIYD09 ARIYD10 ARIYD13 ARIYD27 ARIYD29 ARIYD30 ARIYD33 ARIYD43 ARIYD45 ARIYD49 ARIYD55 ARIYD57 ARIYD58 ARIYD81 ARIYSTP ARIYS00 ARIYS01 ARIYS03 ARIYS04 ARIYS05 ARIYS11 ARIYS12 ARIYS14 ARIYS17 ARIYS20 ARIYS21 ARIYS24 ARIYS25 ARIYZ27
YVPAGE	ARIYC00 ARIYC01 ARIYC02 ARIYC03 ARIYC05 ARIYC09 ARIYC10 ARIYC11 ARIYC15 ARIYC18 ARIYC19 ARIYC20 ARIYC21 ARIYC22 ARIYC23 ARIYC24 ARIYC25 ARIYC26 ARIYD01 ARIYD02 ARIYD03 ARIYD04 ARIYD05 ARIYD06 ARIYD08 ARIYD09 ARIYD10 ARIYD14 ARIYD15 ARIYD17 ARIYD18 ARIYD19 ARIYD20 ARIYD23 ARIYD26 ARIYD28 ARIYD30 ARIYD33 ARIYD38 ARIYD39 ARIYD40 ARIYD48 ARIYD49 ARIYD58 ARIYD62 ARIYD63 ARIYD64 ARIYD65 ARIYD66 ARIYD67 ARIYD68 ARIYD69 ARIYD70 ARIYD71 ARIYD72 ARIYD73 ARIYD77 ARIYI02 ARIYI28 ARIYK38 ARIYK40 ARIYS00 ARIYS01 ARIYS02 ARIYS04 ARIYS05 ARIYS08 ARIYS11 ARIYS12 ARIYS14 ARIYS21 ARIYS24 ARIYX06 ARIYZ04 ARIYZ06 ARIYZ11

REGISTER CONVENTIONS FOR DBSS, DSC, RDS, AND RM

Register 0 - Used to pass values to a "called" module.

Register 1 - Used to pass the pointer to a parameter list to a "called" module.
- Used to return pointers to a "caller".

Register 10 - With the exception of special initialization and termination modules, contains the pointer to YTABLE1 (DSC/DBSS) or to the RDAREA (RDS) or to RMLO (Resource Manager).

Register 13 - Contains the pointer to a module's save area.

Register 14 - Contains the return point to the "calling" module.

Register 15 - Contains the entry point to the "called" module.
- Contains the return code upon return from the "called" module.

DBS UTILITY UNLOAD PROCESSING OUTPUT RECORDS

DBS UNLOAD processing generates a SAM (sequential access method) output file that contains control records and data records.

The control records contain information that identifies the maximum logical record length of the SAM file and describes the tables that were unloaded. A DBS UNLOAD header record and a set of table description records are written to the SAM file before any data records are generated.

Data records for each table are then written. A data record is written to the SAM file for each row of a table being unloaded.

Logical record lengths exceeding 32756 may result from DBS UNLOAD processing.

A description of each record type (and the record format) generated by DBS UNLOAD processing follows; The output record field definitions follow the Type 60 record.

Type 10 - Table UNLOAD Header Record:

A single record is generated by DBS UNLOAD processing. It contains the maximum logical record length necessary to process the remainder of the sequential file as input. The record format is:

Field Description	Type	Length(bytes)
Segment Descriptor	Binary	4
Record Type (value=10)	Binary	1
UNLOAD Sequence ID (value=0)	Binary	1
Reserved		2
Maximum Logical Record Length	Binary	4

Type 20 - SQLDA Table Description Record:

A record is generated for each table. The record contains, in SQLDA format, the basic table definition information required by DBS RELOAD command processing. The record format is:

Field Description	Type	Length(bytes)
Segment Descriptor	Binary	4
Record Type (value=20)	Binary	1
UNLOAD Sequence ID	Binary	1
Creator	Character	8
Table-Name Length	Binary	2
Table-Name	Character	30
SQLDA		
SQLDA Length	Binary	2
SQLDA Data	Character	(varying)

Type 30 - Table Definition Record:

A record is generated for each table and contains table definition information in the SQL CREATE TABLE statement format. The record format is:

Field Description	Type	Length(bytes)
Segment Descriptor	Binary	4
Record Type (value=30)	Binary	1
UNLOAD Sequence ID	Binary	1
SQL Statement Length	Binary	2
SQL CREATE TABLE Statement	Character	(varying - max 8192)

Type 40 - First Table Index Definition Record:

A record is generated if the table being unloaded has at least one index defined. If no indexes exist, no record type 40 is generated. The record contains the definition of the first index for a table in SQL CREATE INDEX statement format. The unloaded table data is in the column-field-value sequence described by this table. If the table has no defined indexes, the table rows are unloaded in the system-determined order. The record format is:

Field Description	Type	Length(bytes)
Segment Descriptor	Binary	4
Record Type (value=40)	Binary	1
UNLOAD Sequence ID	Binary	1
SQL Statement Length	Binary	2
SQL CREATE INDEX Statement	Character	(varying - max 8192)

Type 50 - Other Table Index Definition Record:
A record is generated for the second through nth index defined on the table being unloaded. The record contains the definition of an index in SQL CREATE INDEX statement format. No record type 50's are generated if no index or only one index exists for the table. The record format is:

Field Description	Type	Length(bytes)
Segment Descriptor	Binary	4
Record Type (value=50)	Binary	1
UNLOAD Sequence ID	Binary	1
SQL Statement Length	Binary	2
SQL CREATE INDEX Statement	Character (varying - max 8192)	

Type 60 - Table Row Data Record:
A record is generated for each row of each table being unloaded. All record type 60's for a table are contiguous on the sequential file. No record type 60 is generated if the table has no rows. The record format is:

Field Description	Type	Length(bytes)
Segment Descriptor	Binary	4
Record Type (value=60)	Binary	1
UNLOAD Sequence ID	Binary	1
Column 1 Data Field	(varying)	
- Field Data Control	Binary	2
- Field Data	(varying)	
Column 2 Data Field	(varying)	
.		
.		
.		
Column 255 Data Field	(varying)	

Segment Descriptor - This four-byte field contains the variable-length, spanned record (SAM) segment length and segment control code. The first two bytes contain the length of the segment. The segment length includes the length of the four-byte Segment Descriptor field. The third byte of the Segment Descriptor field contains the segment control code, which specifies the relative position of the segment within the logical record. (The segment control codes are described below.) The remaining bits of the third byte and the fourth byte are zeros.

The segment control code binary values and meanings are:

- 00 - Complete logical record
- 01 - First segment of multi-segment record
- 10 - Last segment of multi-segment record
- 11 - Segment of multi-segment record other than first or last segment

(Null segments are not produced by DBS UNLOAD processing.)

Record Type - The DBS-defined identifier of UNLOAD processing output records.

UNLOAD Sequence ID - This field contains a DBS-assigned identifier (01 - 255) of each table processed during the execution of a DBS UNLOAD command. The table identifier value = 0 is used for record type 10.

Maximum Logical Record Length - This field contains the maximum length logical record contained in the job.

Creator - This field contains the SQL/DS user identifier of the creator of the table being umloaded. It will be padded on the right, if necessary, with blanks.

Table-Name Length - Actual length of table-name in the subsequent data field.

Table-Name - This field contains the SQL/DS identifier of the table being unloaded. It will be padded to the right with blanks, if necessary.

SQLDA Length - The actual length of the SQLDA structure contents in the subsequent record data field.

SQLDA Data - The contents of the SQLDA structure used by DBS for table unload processing.

SQL Statement Length - The actual length of the SQL CREATE TABLE or CREATE INDEX statement in the subsequent record data field.

SQL CREATE TABLE | INDEX Statement - This field contains the SQL CREATE TABLE or CREATE INDEX statement constructed by DBS for the table being unloaded. It does not include the EXEC SQL prefix or a terminating semi-colon. The table or index is identified by creator.table-name or creator.index-name in the SQL statement.

Column Data Field - This field contains one column value from the row of the table to which the record applies. The sequence of the Column Data Fields within the Table Row Data record is determined by SQL/DS.

Field Data Control

- Value = 0: blank row data field; no Field Data follows
- Value < 0: null field; no Field Data follows
- Value > 0: length of following Field Data

Field Data - Contains the actual row data field value if applicable.

DBS UTILITY DEBUG MODE PROCESSING

If the sequence of commands described below are supplied in the DBS Utility command file, a storage dump (partition or virtual machine dump) will be taken following the next SQL error that occurs. A SQL error is identified by a SQLCODE < 0 or a SQLCODE > +100 received after the execution of a SQL command.

The storage dump will reflect a register 15 value = X'811' and is generated by a DBS Utility call to entry point ARISYSDA in the module ARISYSDD.

The command sequence necessary to initiate the storage dump is:

```
.DEBUG
SET ERRORMODE OFF
```

NOTE:
1) .DEBUG must begin in command record column 1.
2) An error condition will occur when the .DEBUG command is processed.
3) A SET ERRORMODE OFF command must follow the the .DEBUG command.

The storage dump will be taken after the next negative SQLCODE or invalid SQLCODE is returned by SQL/DS in the SQLCA after the execution of a SQL command. DBS Utility processing will continue after the dump is generated. If subsequent storage dumps are desired during the same execution of the DBS Utility, then the special command sequence described above must be repeated.

DBS UTILITY DEBUG STORAGE DUMP ANALYSIS GUIDELINES

R15 = x'811'

R13 + 4 = ARIDSQLA register save area address for last ARIPRDID CALL

ARIDSQLA save area address:

- + 12 = ARISYSDA (storage dump routine) return address within ARIDSQLA.
- + 16 = ARISYSDA entry point.
- + 32 = Address of SQLTIE (R3 contents at time of dump).
- + 36 = DBS Utility CPA address (R4 contents at time of dump).

CPA address + X'0C' = the address of the special save area containing the R0 through R15 contents saved by the DBS Utility before executing the dump request call to ARISYSDA.

DBS UTILITY-INITIATED STORAGE DUMPS

The DBS Utility will initiate a storage dump if an illogical condition or critical error arises during execution. Before a DBS storage dump (partition or virtual machine dump) is initiated, the message ARI804E is normally generated and register 15 is set to a hexadecimal dump identification (DUMP ID) value. After a storage dump is generated, DBS processing continues.
The message ARI804E identifies:
- the DBS module initiating the dump, and
- the reason code for the dump.

There are two DBS storage dumps that are not preceded by the message ARI804E:

1. If the storage dump (DUMP ID = X'811') is initiated as a result of the .DEBUG command.

2. If the storage dump (DUMP ID = X'803') is initiated by the module ARIDBS before DBS processing terminates, the final DBS return code (R15) value will be 4 or greater than 8.

The DBS modules initiating storage dumps, the reasons for the dumps, and the hexadecimal dump identification values are:

DBS MODULE	MESSAGE ARI804E REASON CODE	DESCRIPTION OF ERROR CONDITION	DUMP ID R15 CONTENTS
ARIDALC	4	Invalid CALL to module. The call type parameter was not ALLOCATE (X'00') or FREE (X'08').	800
ARIDALI	4	Invalid CALL to module. The call type parameter was not ALLOCATE (X'00') or FREE (X'08').	801
ARIDALT	4	Invalid CALL to module. The call type parameter was not ALLOCATE (X'00') or FREE (X'08').	802
ARIDBS	N/A	Final DBS return code will not be 0 or 8.	803
ARIDCFI	8	Invalid CALL to module.	804

DBS MODULE	MESSAGE ARI804E REASON CODE	DESCRIPTION OF ERROR CONDITION	DUMP ID R15 CONTENTS
		A CALL to the module was made to read the command file: - before the command file was opened, - after the command file was closed, or - after a command file I/O error occurred.	
ARIDCSP	4	On entry to the module, the command file status and the ARIDCSP processing control fields were in an inconsistent state.	805
ARIDCSP	8	Command build logic error. During command build processing, the last non-blank position of the command file record was less than the first non-blank position.	805
ARIDDFI	8	Invalid call parameter A CALL to the module was made to read the command file: - before the command file was opened, - after the command file was closed, or - after a command file I/O error occurred.	806
ARIDDLO	4	The current command was not a DBS DATALOAD TABLE command when the module ARIDDLO was entered.	807
ARIDDLO	8	Command processing logic error. A ARIDCSP return code (R15) value > 8 was received while reading commands for actual processing (i.e. no previous errors have occurred).	807
ARIDDLO	12	A command sequence check logic error occurred during normal command sequence processing.	807
ARIDDLO	16	A logic error occurred during DATALOAD input OPEN processing. The last command processed was not an INFILE or an INMOD command.	807
ARIDDLO	20	Command processing logic error. A ARIDCSP return code (R15) value > 8 was received while reading commands after an error has been encountered.	807
ARIDDUL	4	The current command was not a DATAUNLOAD command when the module ARIDDUL was entered.	813

DBS MODULE	MESSAGE ARI804E REASON CODE	DESCRIPTION OF ERROR CONDITION	DUMP ID R15 CONTENTS
ARIDDUL	8	Command processing logic error. An ARIDCSP return code > 8 was received while reading commands for actual processing (that is, no previous errors occurred).	813
ARIDDUL	12	A command sequence check logic error occurred during normal command sequence processing.	813
ARIDDUL	16	A logic error occurred during DATALOAD output file OPEN processing. The last command processed was not an OUTFILE command.	813
ARIDDUL	20	Command processing logic error. An ARIDCSP return code > 8 was received while reading commands after an error occurred.	813
ARIDEXI	4	Invalid call parameter The call type parameter was not OPEN (X'12') or FETCH (X'02') or CLOSE (X'04').	808
ARIDSFA	4	Invalid call parameter. The call type parameter was not ALLOCATE (X'00') or COMPUTE (X'02') or FREE (X'04').	809
ARIDSQL	4	Invalid call request parameter. The call type parameter was not valid.	810
ARIDSQL	N/A	Storage dump (partition dump) requested via .DEBUG command.	811
ARIDUNL	12	Internal logic error. The SQL I/O area is too small.	812

DBS UTILITY DATALOAD INMOD SUB-COMMAND

The DBS Utility DATALOAD INMOD sub-command is currently restricted to the DL/1 EXTRACT linkage to DBS Utility DATALOAD processing and applies only to VSE. The INMOD sub-command is used in place of the DBS DATALOAD INFILE sub-command to identify that DBS DATALOAD input data file processing should not be performed and that a DL/1 EXTRACT phase should be called for data record input.

DBS DATALOAD processing will CDLOAD the phase(s) specified and call the identified DL/1 phase when an input data record is required. The call parameter list provided to and the return codes expected on return from the called DL/1 EXTRACT phase are described below.

A description of the DL/1 EXTRACT processing in support of this linkage to the DBS Utility is described in Appendix C.

The format of the DBS Utility DATALOAD INMOD sub-command is:
INMOD (phase_name1[,phase_name2] RECFM(V))

where:

phase_name1 is the name of the phase that will be CDLOADed and called by the DBS Utility. On entry to the phase identified by phase_name1, R1 will contain the address of the call parameter address list described below. This command parameter is required.

phase_name2 is the name of the second phase that will be CDLOADed by the DBS Utility and whose entry point address will be passed to the phase_name1 phase as the 1st entry in the call parameter address list described below. This command parameter is optional. If a phase_name2 is not provided, then an address = 0 will be supplied in the 1st call parameter list entry.

RECFM(V) identifies that variable length data records will be supplied to the DBS Utility by the phase_name1 phase. This command parameter is required.

REGISTER CONTENTS ON ENTRY TO PHASE_NAME1 MODULE

R1 contains the address of the following call parameter address list:

0	PTR To Phase_Name2 Entry Point Address Area
0	PTR To Current DBS Message File Lines Per Page Counter
1	PTR To Maximum DBS Message File Lines Per Page Value

NOTES:
1. The Phase_Name2 Entry Point Address Area occupies a fullword of storage.
2. The Current DBS Message File Lines Per Page Counter occupies a halfword of storage.
3. The Maximum DBS Message File Lines Per Page Value occupies a halfword of storage.
4. The high order bit of the last address in the address list is set to 1 (ON).

R13 contains the address of a 72-byte save area.
R14 contains the DBS Utility return address.
R15 contains the entry point address of the phase_name1 phase.

REGISTER CONTENTS ON RETURN FROM PHASE_NAME1 MODULE

R1 contains the address of the data record area when the R15 return code value is 0.
R2 through R14 are restored.
R15 contains one of the following return code values:
- 0 - Next data record available. R1 contains the data record area address.
- 4 - No more data records.
- 8 - A error has been encountered by the phase_name1 module.
- 12 - An invalid call has been received by the phase_name1 phase. (that is, after a return code 4 was returned on the previous call)

DATA RECORD AREA DESCRIPTION AND FORMAT

On return to the DBS Utility from the phase_name1 phase, R1 is expected to contain the address of the data record area if the return code value in R15 = 0. The data record area begins with a 4-byte Record Descriptor Word (RDW) that identifies the length of the data. The RDW length includes the length of the RDW itself.

The expected format of the data record area contents is:

```
   Positions 1-4:  RDW
               1-2:  Length of data record.
               3-4:  00
   Position  5-LL:  Data.
```

ISQLMAP COMMAND

The ISQL load map is displayed by entering the command ISQLMAP. The command may be entered by any user at anytime except during display mode or INPUT command mode.

The first line of the map shows the current storage location of the Global Control Block (GCB) and the trace table for this user, followed by the current location of all non-system dependent modules. The second section of the display shows the current location of all system-dependent modules.

ISQL DUMPS - TASK ID (VSE)

GETMAIN storage for ISQL modules can be identified by TASK in dumps. The format of storage is:

- Standard CICS Storage Accounting Area - 8 bytes
- Character String "TASK" - 4 bytes
- Numeric CICS Task ID - 4 bytes
- Usable Storage - Requested Storage - n bytes
- Standard CICS Storage Accounting Area - 8 bytes.

The modules requesting storage, via a call to ARISYSDG or ARIIGM, need not be aware of the Task ID in the dump. ARISYSDG and ARIIGM will increase the requested length by 8 bytes and adjust the pointer to the usable storage. The reverse adjustments will be done by ARISYSD2 and ARIIFM for FREEMAINs.

GETMAINs in the ISQL transaction (ARIITRM phase) do not have a task identifier in the dump. Automatic storage, for ARIICI2D, does not have a task identifier in the dump. The task identifier is stored near the end of the Global Control Block (GCB) in field IGCTSK.

ISQL TRACE

ISQL uses a variably-sized wrap-around table in core to trace most calls and returns. The initial table is obtained by ARIICIC2D (for VSE) or by ARIINITC (for VM). The size of the table can be changed by the user with the ISQLTRACE command. The initial table holds up to 50 entries. The ISQL Trace facility may not be turned off.

The following illustrates the format of one trace entry:

FIELD NAME	LENGTH	CONTENTS
TRMOD	8	Name of module generating the trace entry
TRCMOD	8	Name of module being called
TRR15	4	Register 15 contents for returns
TRLBL	2	Sequential number of the call or routine within the module generating the trace entry
TRCODE	1	'C' for Call or 'R' for return
TRCHAR	5	Module-dependent description of the call or return

For reasons of performance, not all calls/returns are traced. Module-dependent information is used only where meaningful.

The following pages contain ISQL modules and descriptions of the module-dependent data passed by each module for trace on external calls or returns. This is the data that appears in the TRCHAR field of the trace table entry.

ISQL TRACE DATA

ARIIBIN

GSQLDA - Call ARIIGM to get storage for the SQLDA
FSQLDA - Call ARIIFM to free storage for SQLDA
NOSTR - Return because there was no storage available for SQLDA
SYNTX - Returning because of syntax error
CANCL - Returning because CANCEL was entered by the user or the system
PREP - Call ARIISQL to do a PREPARE or on return PREPARE was unsuccessful
FREE - Call ARIIFM to free the SQLDA (after PREPARE)
GET - Call ARIIGM to get SQLDA (first was not big enough)
ROLLB - SQL ROLLBACK was issued
DESCR - Call ARIISQL to do a DESCRIBE or on return DESCRIBE was unsuccessful
HEADR - Call ARIIRWI to write header line for column information
CINFO - Call ARIIRWI to write column names to the screen
READ - Call ARIIRWI to issue READ for data or commands
EXECI - Call ARIISQL to issue an EXECUTE IMMEDIATE
CMDLH - Call ARIITKN to get length of command
CKCMD - Call ARIIVLD to see if the command is valid
FRCMD - Call the routine to process the input subcommand
EXPRM - Call ARIITKN to check for extra parameters
CMMIT - SQL COMMIT was issued
SQLMG - Call ARIISMG to write SQL message

ARIICAN

DELQ - Call ARIITS to delete temporary storage
ROLLB - Call ARIISQL to do rollback work
ROLLR - Call ARIISQL to do a rollback work release

ARIICMD

EXIT - Return from exit command processing
OP - Return from operator command processing
CMDTK - Call ARIITKN to get command
EXCES - Call ARIITKN to get excess data on exit command
ROLLB - Call ARIISQL to perform rollback work on exit
1ST - First call to ARIIOCI
OTHR - Other calls to ARIIOCI
FIN - Final call to ARIIOCI
OPL - Call ARIIRWI to display a single line in operator command processing
SCRN - Call ARIIRWI to display a full screen of operator command data

When a command processing routine is being called, the first 5 characters of the command being processed are issued for TRCHAR data. For example, if the command processing routine for RECALL is being called, the TRCHAR data is RECAL.

ARIIDQY

TKN2 - Call ARIITKN to get the second token on the command
TKN3 - Call ARIITKN to get the third token on the command
TKN4 - Call ARIITKN to get the fourth token on the command
CANC - Return because the command was cancelled

ARIIERS

NAME1 - Call ARIITKN for the first SQL command name
OTHRN - Call ARIITKN for the other SQL command names
DEL - Call ARIISQL1 to delete the SQL command from the stored queries table
DCOM - Call ARIISQL to perform a commit after the delete completed successfully
NFCOM - Call ARIISQL to perform a commit after the deleted did not complete successfully because the SQL command was not found

ARIIFCC

CKEY - Call ARIITKN to get the column keyword (ie. NAME, ZEROS, DPLACES, or WIDTH)
TKN1 - Call ARIITKN to get the first token after the COLUMN keyword
ZERO - Call ARIITKN to process the value after the ZEROS keyword
WIDTH - Call ARIITKN to process the value after the WIDTH keyword
NAME - Call ARIITKN to process the value after the NAME keyword
MORE - Call ARIITKN to get the next keyword to process
DPLAC - Call ARIITKN to process the value after DPLACES keyword

ARIIFCI

TKN1 - Call ARIITKN to get the first token after the format exclude or format include
TKN2 - Call ARIITKN to get the second token after the keyword being processed
PAREN - Call ARIITKN to look for parentheses found around the column list

STR2 - Call ARIITKN to process the column list found inside the parentheses
EXTRA - Call ARIITKN to look for more information found on line to process

ARIIFCS

TKN1 - Call ARIITKN to get the first token on the FORMAT KEYWORD command
EXCPT - Call ARIITKN to get the column id after the EXCEPT keyword
PAREN - Call ARIITKN to look for the parentheses around the column list
STR2 - Call ARIITKN to process the column list inside the parentheses
EXTRA - Call ARIITKN to look for more information on the command line

ARIIFET

SUBTL - Call ARIIRPT2 to do subtotal processing
TOTTL - Call ARIIRPT5 to do total processing
FETRW - Call ARIISQL to fetch a row of query result
UNDRS - Call to ARIIRPT2 to get underscores
SUBTI - Call ARIIRPT2 for and of file - subtotal call
TOTTI - Call ARIIRPT5 for end of file - total call
GETDA - Return getting data routine
ACCUM - Call ARIIRPT3 to accumulate total or subtotals

ARIIFMC

TKNI - Call ARIITKN to get the first operand of the format command (first token)
MORE - Call ARIITKN to check for more operands to process because format is now multiple operand
KEY - Call ARIITKN to get the keyword to process after the operand
BLNKS - Call ARIITKN to check for keyword blanks after a number on the format separator command
TTITL - Call ARIIRWI to display the top title
BTITL - Call ARIIRWI to display the bottom title
GROUP - Call to ARIIFCS to process GROUP command
SUBTO - Call to ARIIFCS to process SUBTOTAL command
TOTAL - Call to ARIIFCS to process TOTAL command
COLUM - Call to ARIIFCS to process COLUMN command
EXCLU - Call to ARIIFCS to process EXCLUDE command
INCLU - Call to ARIIFCS to process INCLUDE command

ARIIFMT

FLTCV - Call ARIMDFP1 to do floating point conversion

ARIIFOR

NORET - Calls to modules which are not our modules which means there will be no return entry in the trace table

ARIIHLP

TOPIC - Call ARIITKN to get the help topic
EXCES - Call ARIITKN to excess data after topic

ARIILST

TKN1 - Call ARIITKN to get the operand (SET or SQL)
NAME1 - Call ARIITKN to get the first statement name or *
OTHRN - Call ARIITKN to get the next statenent name or *
CMD - Call to ARIIRWI to display the command if a routine is executing
ISQLN - Call to ARIIRWI to display a SQL command if a SQL command name was entered
SQL* - Call to ARIIRWI to display a SQL command if list SQL * was entered
DECL - Call to ARIISQL2 to declare the cursor for list SQL * processing
OPEN - Call to ARIISQL2 to open the cursor for list SQL * processing
FETCH - Call to ARIISQL2 to perform a fetch for list SQL * processing
CLOSE - Call to ARIISQL2 to close the cursor for list SQL * processing

ARIIMAP

BLANK - Call ARIIRWI to display a blank line
LOOP - Call ARIIRWI to display other lines of ISQLMAP display
RMAIN - Call ARIIRWI to display last line of ISQLMAP display

ARIIMSG

CMD - Call to ARIIRWI to display the command line if we are in a procedure and the command line has nor previously been displayed
LOOP - Call to ARIIFOR to request a line of a message
TRACMSG- Field containing the message number when ARIIRWI is called to display a line of the message

STAT - Call to ARIIRWI to display a status area
F057E - Call to ARIIFOR to retrieve message ARI057E
R057E - Call to ARIIRWI to display message ARI057E
F7022 - Call to ARIIFOR to retrieve message ARI7022E
R7022 - Call to ARIIRWI to display message ARI7022E

ARIIOCI

CALL1 - Call ARIPRDI for the first time
CALL2 - Call ARIPRDI for the second time

ARIIPQY

NOSTR - Return because no storage for SQLDA
CONSC - Call ARIICNV to convert screen size
FINIT - Call ARIIFM to FREEMAIN initial SQLDA
PREP - Call ARIISQL to do a PREPARE
DESC - Call ARIISQL to do a DESCRIBE
F10DA - Call ARIIFM to free SQLDA with ten variables
BPTR - Call ARIIFM/ARIIGM for one row buffer
TANK - Call ARIIFM/ARIIGM for holding tank
SQLDA - Call ARIIFM/ARIIGM for SQLDA
SCRB - Call ARIIFM for screen row buffer
OPENC - Call ARIISQL to open cursor
CLOSE - Call ARIISQL to close cursor
TKN1 - Call ARIITKN to get first token
TKN2 - Call ARIITKN to get second token
LBSIZ - Call ARIIGM to get screen buffer
GCE1 - Call ARIMDFP1 to convert GCE to EBCDIC

ARIIPRF

DELQ - Call ARIITS to delete temporary storage
TKNM - Call ARIITKN to simulate run for the master profile
MASTR - Call ARIIRUN for master profile
TKNU - Call ARIITKN to simulate run for the user profile
USERP - Call ARIIRUN for user profile
TKNS - Call ARIITKN to simulate run for the user-specified routine
USERS - Call ARIIRUN for user-specified routine

ARIIPRTC

TOKIZ - Call ARIITKN to get class or copies
OPVD - Call ARIIOVD to validate operands
INVKY - Return because of invalid keyword
CANCL - Return because of a CANCEL
PRTC - Call ARIISQL to do a CLOSE CURSOR
PRSQL - Call ARIISMG to display SQL error message
CANDE - Return because of a CANCEL
PRRET - Return to caller
PRDRT - Return to display mode

ARIIPRTD

TOKIZ - Call ARIITKN to get class or copied
OPVD - Call ARIIOVD to validate operands
INVKY - Return because of invalid keyword
CANCL - Return because of a CANCEL
NODSP - Return because user not in display mode
PROTK - Call ARIITKN to get class/copies
COENQ - Call ARIIENQ to en-queue on printer
NOENQ - Return because printer is busy
INVID - Return because of invalid terminal id
TNSU - Return because of terminal not supported
RTFL - Return because route failed
PRHDR - Call ARIIHDR to print header
PRCL - Call ARIISQL to close cursor
PRDP - Call ARIISMG to display SQL error message
PRCDE - Call ARIIDEQ to de-queue printer
CLERR - Return due to error on SQL close cursor
PROP - Call ARIISQL to open cursor
PRODE - Call ARIIDEQ to de-queue printer
OPERR - Return due to power error
CANDE - Call ARIIDEQ to de-queue printer and return
PRSQL - Call ARIISMG to display SQL error message
EOFDE - Call ARIIDEQ to de-queue printer and return
PRCAA - Return due to cancel
CSQER - Call ARIISMG to display SQL error message
PRCNP - Return to display mode
PRDEQ - Call ARIIDEQ to de-queue printer
PRSP - Call ARIISPGE to send page
PRCRE - Return to display mode
PRRRE - Return to display mode
PRPOS - Return to display mode
PRCLA - Call ARIITKN to get class/copies
PRCOP - Call ARIITKN to get number of copies
PRTER - Call ARIITKN to get terminal id
PRDES - Call ARIITKN to get destination
PRWA - Call ARIIWAIT to wait for printer I/O
PRLCA - Call ARIIDEQ to de-queue printer
PRST1 - Call ARIISTXT to send text
PRDRT - Return to caller

ARIIPSQ

ROLL - Call ARIISQL and ARIISMG for ROLLBACK
COMM - Call ARIISQL and ARIISMG for COMMIT
CONN - Call ARIISQL and ARIISMG for CONNECT
TKN1 - Call ARIITKN to get first token
TKN2 - Call ARIITKN to get second token
TKN3 - Call ARIITKN to get third token

RELEX - Call to ARIITKN to find excess parameters
G/R/E - Call AXTCEXTS for GRANT/REVOKE/EXTRACE
EXTR - Call AXTCEXTS for extract
SEL - Call ARIIPQY for SELECT
EXIM - Call ARIISQL and ARIISMG for EXEC IMMEDIATE
SQL - Return, SQL statement was not permitted
WORK - RETURN, COMMIT and ROLLBACK must have an argument of WORK or WORK RELEASE

ARIIQRY

KEYWD - Call ARIITKN to get the keyword after the list set
CMD - Call ARIIRWI to write out the keyword list if a bad keyword was entered
SEP - Call ARIIRWI to write out the column separator string

ARIIREC

NAME - Call ARIITKN to get the SQL command name
CURB - Call ARIIRWI to display the current buffer
EXCSS - Call ARIITKN to check for excess data on the RECALL command
CTOP - Call ARIIPQL1 to move current buffer to previous buffer
CMMIT - Call ARIISQL to issue a COMMIT WORK
SELCT - Call ARIISQL1 to get a stored SQL command from the stored queries table
DSQL - Call ARIIRWI to display the recalled SQL command
DPREV - Call ARIIRWI to display the previous SQL command

ARIIRNM

ONAME - Call ARIITKN to get the old name on RENAME command
NNAME - Call ARIITKN to get the new name on RENAME command
EXCSS - Call ARIITKN to check for excess parameters on RENAME command
CTOP - Call ARIIPQL1 to move the current SQL command to the previous buffer
NSLCT - Call ARIISQL1 to recall the new SQL command
OSLCT - Call ARIISQL1 to recall the old SQL command
RBK - Call ARIISQL to issue a ROLLBACK WORK
CMMIT - Call ARIISQL to issue a COMMIT WORK
DELET - Call ARIISQL1 to delete the old SQL command
INSRT - Call ARIISQL1 to insert the command under the new name

ARIIRPT

GETFL - Return due to GETMAIN failure
GETST - Call ARIIGM to get accumulator storage (RPT1)
FREE - Call ARIIFM to free RPT blk (RPT1)
SUBTL - Return from ARIIRPT2
ACCUM - Return from ARIIPRT3
FREES - Call ARIIFM and return from ARIIRPT4
TOTAL - Return from a RIIRPT5

ARIIRUN

PREP - ARIISQL called for PREPARE
DESCR - ARIISQL called for DESCRIBE
OPEN - ARIISQL called for OPEN
FETCH - ARIISQL called for FETCH
CLOSE - ARIISQL called for CLOSE
COMMI - ARIISQL called for COMMIT
ROLLB - ARIISQL called for ROLLBACK
CKNAM - ARIITKN called to get userid, routine name

ARIISCC

GCHNG - Call ARIIGM to get a buffer for the CHANGE command
FCHNG - Call ARIIFM to free the buffer for the CHANGE command
STR1 - Call ARIITKN to get STRING 1 on the CHANGE command
STR2 - Call ARIITKN to get STRING 2 on the CHANGE command
CEXC - Call ARIITKN to look for excess data on the CHANGE command
SELCK - Call ARIITKN to see if SELECT is the first thing in the SQL command buffer
CDISP - Call ARIIRWI to display the changed SQL command
STKN1 - Call ARIITKN to get the first token after the START command
PRMLS - Call ARIITKN to look for a parameter list after the START command
SEXC - Call ARIITKN to look for excess data on the START command
GSTRT - Call ARIIGM to get a buffer for the START command
STRT - Call ARIIRWI to display the START command if a routine is running
SQL - Call ARIIRWI to display the SQL command after substitutions were performed
FSTRT - Call ARIIFM to free the buffer for the START command
PARMS - Call ARIITKN to look for the parameters in the parameter list

When ARIISQL1 and ARIISQL are called, the first 5 characters of the function to be performed are used for TRCHAR data. For example, if the function to be performed is ROLLBACK, the characters ROLLB are used.

ARIISET

OPER1 - Call ARIITKN to get the operand after the SET command
CLVAL - Call ARIITKN to get the class value
COVAL - Call ARIITKN to get the copies value
PR1 - Call ARIITKN to get the keyword after PRINTROUTE
PR2 - Call ARIITKN to get the value after the PR1 keyword
RNVAL - Call ARIITKN to get the runmode value
CEVAL - Call ARIITKN to get the cost estimate

ARIISMG

CMD - Call ARIIRWI to display the command line
MSG - Call ARIIRWI to display a line of the message

ARIISQL

CONCT - Call ARIITKN to parse the CONNECT command for connect
UID - Call ARIITKN to parse the CONNECT command for the userid
IDENT - Call ARIITKN to parse the CONNECT command for identified
BY - Call ARIITKN to parse the CONNECT command for by
PASS - Call ARIITKN to parse the CONNECT command for the user password
END - Call ARIIRTRN to end the ISQL session if a severe SQL error has occurred
7946 - Call ARIYM04 to display message ARI7946

The SQLCODE is used for TRCHAR data on each return from this module.

ARIISTR

NAME - Call ARIITKN to get the name after the STORE command
TKN2 - Call ARIITKN to get the token after the name on the STORE command
EXCSS - Call ARIITKN to check for excess parms
CHECK - Call ARIISQLI to see if the stored SQL command exists
DEL - Call ARIISQLI to delete the existing stored SQL command
REPLY - Call ARIITKN to process the message reply to message ARI7576D or ARI7567D
INS - Call ARIISQLI to insert the stored SQL command into the stored queries table
ROLL - Call ARIISQL to perform a rollback work
COMM - Call ARIISQL to perform a commit work

ARIIST2

AUTOT - Call ARIITKN to get the token after AUTOCOMMIT
CASET - Call ARIITKN to get the token after CASE
CONTT - Call ARIITKN to get the token after CONTINUE
DECT - Call ARIITKN to get the token after DECIMAL

ARIIST3

NULLT - Call ARIITKN to get the token after AUTOCOMMIT
PTKN1 - Call ARIITKN to get the token after PAGESIZE
PNVAL - Call ARIITKN to get the token after width on the PAGESIZE command
PWNEX - Call ARIITKN to get the token after the width value
PLVAL - Call ARIITKN to get the token after length on the PAGESIZE command
PLNEX - Call ARIITKN to get the token after the length value
SEP1 - Call ARIITKN to get the first token after separator
SEP2 - Call ARIITKN to get the second token after separator
DISPO - Call ARIIRWI to display the old separator value
DISPN - Call ARIIRWI to display the new separator value
VART - Call ARIITKN to get the token after VARCHAR

ARIITRC

TKN1 - Call ARIITKN to get the first operand of the FORMAT command (first token)
MORE - Call ARIITKN to check for more operands to process because format is now multiple operand
KEY - Call ARIITKN to get the keyword to process after the operand
BLNKS - Call ARIITKN to check for keyword blanks after a number on the format separator command
TTITL - Call ARIIRWI to display the top title
BTITL - Call ARIIRWI to display the bottom title
GROUP - Call to ARIIFCS to process GROUP command
SUBTO - Call to ARIIFCS to process SUBTOTAL command
TOTAL - Call to ARIIFCS to process TOTAL command
COLUM - Call to ARIIFCC to process COLUMN command
EXCLU - Call to ARIIFCS to process EXCLUDE command
INCLU - Call to ARIIFCS to process INCLUDE command

ARIIVLD

On return, the first five characters of the command being processed are used as TRCHAR data.

The following pages contain SQL/DS Trace information regarding:

- Invoking
- Formatting and Printing
- Format of Output
- DUMP Option of TRACE Command
- Using Service Temporary Trace Points
- Trace Point Descriptor Records
  Trace Point Descriptor Records are in numerical order except for the DBSS Service Temporary trace points (9900-9909). They follow the Update Statistics trace points (2900-2942).

INVOKING THE SQL/DS TRACE FACILITY

SQL/DS Trace Facility can be invoked to trace RDS and/or DBSS activity within the SQL/DS partition or virtual machine in two ways:

- Via the TRACDBSS and TRACRDS SQL/DS initialization parameters. These are described in SQL/DS Operation, listed in the Preface, Section 3. The detailed meanings of the parameter positions are described under the TRACE ON operator command in Section 4 - "Problem Determination and Isolation". When SQL/DS Tracing is invoked by this method, tracing continues until the SQL/DS partition or virtual machine terminates normally or abnormally or until the TRACE OFF operator command (available only in multi-partition mode or multiple virtual machine mode) is entered. TRACE OFF is described in SQL/DS Operation, Section 4.

  Note: If you specify a 2 in the position for the Statistics subcomponent on TRACDBSS, you will also get Working Storage Manager tracing within DBSS. If you specify a 2 in the position for the Parser subcomponent on TRACRDS, you will also get Working Storage Manager tracing within RDS.

- Via the TRACE ON SQL/DS Operator command. Tracing continues until the TRACE OFF command is issued or until the SQL/DS partition or virtual machine terminates normally or abnormally. TRACE ON and TRACE OFF commands are described in SQL/DS Operation, Section 4, "Problem Determination and Isolation". SQL/DS termination always attempts to close the SQL/DS Trace output tape file and issues a message indicating successful (or unsuccessful) close.

  Note: If, in response to the TRACE ON prompt, you respond with "STAT 2", you will get DBSS Working Storage Manager tracing in addition to the Statistics subcomponent tracing. If you respond "PA 2", you will get RDS Working Storage Manager tracing in addition to the Parser subcomponent tracing.

Note that SQL/DS Trace output is always to tape.

For VSE: To execute SQL/DS Trace, a TLBL statement with the filename ARITRAC (all other TLBL parameters are optional) must be part of the SQL/DS job JCL. The tape unit must not be assigned (no // ASSGN statement) since SQL/DS dynamically assigns and unassigns the Trace tape unit. Also, SQL/DS never rewinds the Trace tape except on the end of the volume. This allows the possibility of multi-file output on the tape volume but leaves the operator or the JCL responsible for ensuring proper tape positioning, rewinding, unloading, etc. The Trace Tape files are always written with IBM (EBCDIC) standard labels.

For VM: The CMS EXEC which starts SQL/DS issues the following CMS FILEDEF command for the trace output file:

```
FILEDEF ARITRAC TAP2 SL (BLOCK 4096 NOCHANGE
```

If different FILEDEF options are desired, a CMS FILEDEF command (with the ddname ARITRAC) can be entered before invoking the CMS EXEC which starts SQL/DS. If the tape volume is unlabeled, only the TAPn and NL parameters are required. If the tape volume(s) are standard label (required for multi-volume output), the TAPn and SL parameters are required. Note that CMS will ABEND if end of volume is reached on an unlabeled output tape file. If multi-volume output is desired (or possible) the FILEDEF LEAVE and NOEOV parameters must be omitted. The LEAVE parameter suppresses rewinding before OPEN processing and after CLOSE processing. For standard label tapes,

the CMS LABELDEF command is optional (allowing specification and checking of file header label fields). If the CMS LABELDEF command is desired, it must be entered (with the ddname ARITRAC) before invoking the CMS EXEC which starts SQL/DS.

FORMATTING AND PRINTING THE SQL/DS TRACE

SQL/DS provides a utility program ARIMTRA to format and print the SQL/DS Trace tape file. The utility provides options to select all or only specified subsets of the trace output file. Use of the SQL/DS Trace Formatter utility program is described in SQL/DS Operation, Section 4, "Problem Determination and Isolation" and in SQL/DS Planning and Administration, "Problem Handling".

FORMAT OF SQL/DS TRACE OUTPUT

Each time a trace point is encountered in a function/subcomponent that is activated for trace (and the agent or userid is active for trace), trace point output is produced. Trace point output has the following format:

The first printed line is the Trace Header and it has the following format:

TRACEPOINT=nnnn
  Identifies the trace point number

DBSS_OP=nn|RDS_OP=nn
  Identifies trace point as DBSS or RDS and gives the current DBSS or RDS OPCODE being executed. Special "OPCODE" numbers are assigned to DBSS non-OPCODE functions such as checkpoint, archive, warm start, etc.

  RDS OPCODEs come from the RDIIN control block on external calls to RDS (in RDIIN field RDICTYPE). RDS places the OPCODE in field RDAOPCOD in the RDAREA control block for Trace and for Problem Determination. These OPCODEs are described in Section 5 - RDIIN Note 1 (page 161).

  DBSS OPCODEs (excluding special "OPCODEs" - see below) come from DBSI calls (calls to module ARIYM00) as the OPCODE parameter and are placed in field YT1OPCOD in the YTABLE1 control block for Trace and for Problem Determination. These DBSI call OPCODEs are described in Section 6 - DBSS Op Codes (page 267).

  Certain DBSS functions are executed without formal DBSI calls to the agent which executes that function. DBSS sets (in YT1OPCOD) special "pseudo-OPCODEs" to cover a number of these situations as follows:

  OPCODE=99: DBSS is performing warm start SQL/DS Initialization as a result of the Initialization parameter STARTUP=W.

  OPCODE=98: DBSS is performing data base generation and initialization as a result of the SQL/DS Initialization parameter STARTUP=C.

  OPCODE=97: DBSS is being initialized and is restoring the data base from an archive tape as a result of the SQL/DS Initialization parameter STARTUP=R.

  OPCODE=96: DBSS is being initialized and is adding new DBSPACE(s) to the data base as a result of the SQL/DS Initialization parameter STARTUP=S.

  OPCODE=95: DBSS is being initialized and is adding new DBEXTENT(s) to the data base as a result of the SQL/DS Initialization parameter STARTUP=E.

  OPCODE=94: DBSS is being initialized and is redefining/formatting the log data set(s) (COLDLOG) as a result of the SQL/DS Initialization parameter STARTUP=L.

  OPCODE=93: DBSS is performing a SQL/DS checkpoint or checkpoint and archive in the checkpoint agent (agent 2).

  OPCODE=92: DBSS is performing an asynchronous rollback or commit of an LUW. This can be caused by conditions such as deadlock and the FORCE operator command.

  OPCODE=91: DBSS is executing (in the operator agent) a SQL/DS operator command from the SQL/DS operator.

USERID=cccccccc
  SQL/DS User-ID
    The operator agent (agent 1) always has the User-ID 'OPERATOR'. SQL/DS Initialization occurs under that User-ID.

    The checkpoint agent (agent 2) always has the User-ID 'CHECKPT'.

The Ready/Recovery agent (agent 3, Multi-user mode only) always has the User-ID 'RECOVERY'.

COMP=compname
Name of subcomponent/function being traced. Same as name specified via the TRACE ON operator command.

AGENT=nn
nn is the ordinal of the SQL/DS Agent Structure.
The operator is always agent 1.
The checkpoint is always agent 2.
In Single-user Mode, agent 3 is the (only) user agent.
In Multi-user Mode, agent 3 is always the Ready/Recovery agent.
In Multi-user Mode, agents 4 through n are the user agents (where n is the NCUSERS parameters value plus three).

DATE=mm-dd-yy
Date of trace point activation.

TIME=hh:mm:ss
Time of trace point activation. Time is adjusted to local time as specified by VSE Job Control via the ZONE JCS or as specified by the VM SYSTIME system generation macro. The time is in 24 hour notation. Thus 2:00 p.m. is 14:00:00. The time zone adjustment is made at Trace Format and Print time.

[LUW_ID='hhhhhhhh'X]
DBSS Logical-Unit-of-Work-ID. Appears for DBSS trace points only.

Except for DBSS Entry and DBSS Exit trace points, the Trace Header is always followed by (for both level 1 and 2 tracing):

- If module entry trace point:
  MOD_CALLED='entry point name'
- If module exit trace point:
  MOD_RETURNED='module name' (followed by)
  RETCODE=[-]n...n (present if module passed a return code)
- If neither module entry nor module exit (middle of some module):
  MOD_REPORT='module name'

For DBSS Entry trace points, the Trace Header is always followed by (for both level 1 and 2 tracing):
DBSS ENTRY: L_OPCODE='DBSS_opcode_name'

For DBSS Exit trace points, the Trace Header is always followed by (for both level 1 and 2 tracing):
DBSS EXIT: L_OPCODE='DBSS_opcode_name' RETCODE=[-]n...n
(RETCODE is the DBSS Return Code, see Section 6, DBSS Return Codes page 268)

For all trace points, only if trace level is 2, variables (if any), of processing and debugging significance are displayed after the above information. Variables are displayed in the general form:

- L_varname=value-or-string, or
- G_varname=value-or-string

The L or G prefix indicates that the variable is only locally addressable to the issuing module (L), or is globally addressable (via SQL/DS control blocks) to all modules in the SQL/DS partition or virtual machine (G).

Varname should be the name of the data item (simple entity, structure, or substructure) by which the issuing module addresses the data item.

value-or-string is of the form:

- [-]n...n if displaying decimal data (binary internally)
- 'c...c' if displaying (unconverted) character data
- 'h...h'X if displaying hex data (binary internally)
- L/G_varname=HEXADECIMAL DUMP: (line 1)
  - 0000 hhhhhhhh (up to 12 hex words) (line 2)
  - 0030 hhhhhhhh (up to 12 hex words) (line 3 if needed)

  More lines are included as needed for up to 32K of hex data. For lines 2 to n, the first 4 characters are the hex offset of displayed hex dump data (0 for first line, 30 for second line, etc.). The rest of the line is hex output displayed in groups of 8 hex characters (each a binary word) with 12 groups per line (less only if end of output data reached).

Trace output may also contain printable character strings that are trace point dependent and clarify processing being performed or the data being displayed.

THE DUMP OPTION OF THE TRACE COMMAND

This option is described under the TRACE ON SQL/DS operator command in SQL/DS Operation, Section 4, "Problem Determination and Isolation". When the DUMP option is chosen (and a trace point number is entered), the first time the specified trace point produces trace output, a snap dump of the SQL/DS partition or virtual machine is displayed on SYSLST (VSE) or to the virtual printer (VM). The dump occurs only on the first activation of the specified trace point number. The amount of dump output is controlled by the SQL/DS Initialization parameter DUMPTYPE=P|F|N where P is the default. If F is specified, a full partition or

virtual machine dump occurs. If P or N is specified, SQL/DS phases (load modules) are bypassed in the partition or virtual machine dump. If you are console debugging SQL/DS with Trace active you can cause the DUMP option to occur by locating the DS2CVT control block (described in Section 5, see page 55) and turn on bit flag DS2TRFLD in flag byte DS2TRFLG (offset hex BE), and store (in binary) a trace point number in field DS2TRDNT (halfword at hex offset BC). Note that the function/subcomponent of the stored trace point number must be active for tracing.

Note: To locate the general register values of the module containing the trace point that caused the snap dump, you must use the standard method of following register save areas. The displayed register 13 points to the register save area of module ARISYSD, which in turn points back to the register save area of module ARICPDM. The ARICPDM save area points back to the ARIXETR save area (for RDS trace point) or ARIYE14 (DBSS trace point). This save area contains the register values of the module containing the trace point (their values at the point of the trace call BALR).

USING SERVICE TEMPORARY TRACE POINTS

Trace point numbers for Service Temporary trace points are reserved. They have the numbers 9900 through 9999. Predefined trace point descriptors for the DBSS/DSC Linkage Edit (phase/load module ARISQLDS) are contained in module ARIYZTP and are described at the end of DBSS trace points (page 435). Predefined trace point descriptors for the RDS Linkage Edit (phase/load module ARIXRDS) are contained in module ARIXTR5 and are described at the end of RDS trace points (page 487).

These predefined trace points allow SQL/DS service personnel to insert either via source code or via patching object code (for example, using the PDZAP utility program) temporary trace points for any purpose (normally problem isolation). Note: DBSS predefined Service Trace points are displayed as DBSS trace points with COMP=STAT in the trace header. RDS predefined Service trace points are displayed as RDS trace points with COMP=CODE in the trace header.

If inserting a Source temporary trace point:
- Use the TRACE macro if DBSS or DSC module
- Use the XTRACE macro if RDS module
- Insert the macro at the point where output is desired
- The first positional parameter is the trace point number
- Following this are variable names as positional parameters for each local variable to be displayed. There must be as many local variable (or ADDR(0) in place of variable name if no variable desired) as are defined in the corresponding trace point descriptor.
- If the trace point is to be activated only if a specific DBSS (for TRACE macro) function or RDS (for XTRACE macro) function is being traced, add the keyword COMP(n) to the macro where n is:
  - 1 - RDS Executives (EXEC)
  - 2 - RDS Parser (PA)
  - 3 - RDS Optimizer (OPT)
  - 4 - RDS Code Generator (CODE)
  - 5 - RDS Interpreter and Authorization (INT)
  - 6 - RDS Security/Audit (AU)
  - 9 - DBSS Entry (Call ARIYM00) (ENTRY)
  - 10 - DBSS Exit (ARIYM00 Return) (EXIT)
  - 11 - DBSS Log/Recovery (LOG)
  - 12 - DBSS Lock (LOCK)
  - 13 - Reserved for DBSS LUW
  - 14 - DBSS Data Control (DC)
  - 15 - DBSS Data Manipulation (DM)
  - 16 - DBSS Storage (I/O) Management (STOR)
  - 17 - DBSS Sort (SORT)
  - 18 - DBSS Index (INDEX)
  - 19 - DBSS Update Statistics (STAT)
- If the trace point is to be activatd if any SQL/DS tracing is active, add the NOCOMP keyword (omitting the COMP keyword).
- See the prolog of the TRACE or XTRACE macro for more details if needed.

If inserting an object temporary trace point:
- Choose the logic point in the module where you want trace output to occur and service temporary trace point number that suits your requirements.
- Using the module patch area (@PSPACE in assembler listing) or the DBSS or RDS patch module (See "SQL/DS Patch Areas", page 262 in this volume), insert code that does the following (in addition to repeating any displayed instructions):
  - Loads a parameter list of fullwords with the addresses of local variables to be displayed. There must be as many words in the list as there are local variables in the trace point descriptor used (with 0 address loaded into any variable position for which no variable is to be displayed). This parameters list (standard PLS call format) should be in the module automatic (DSECT) storage area. Pick existing automatic storage currently useable, or patch the module assembler constant @SIZDATD that defines the size of the module automatic storage. This ensures reentrant coding. Register 1 must point to this parameter list when the BALR R14, R15 below is executed. No parameter list is required if the chosen service trace point descriptor displays only global variables such as YTABLE1 or RDAREA.

- Loads Register 15 with the address of the DBSS trace service module (ARIYE14) if in DBSS link edit or the RDS trace service module (ARIXETR) if in RDS link edit. This address is already in the module if it contains trace points. Otherwise:
  a. For DBSS/DSC, the address is in YTABLE1 fullword YT1TRCSP (hex displacement 4BC from Register 10).
  b. For RDS, the address is in RDAREA fullword RDATRCSP (hex displacement 90 from Register 10).
- Does a BALR R14, R15, followed by a halfword containing the service temporary trace point number in binary.
- Branches back to the main line code.

Note that you need not test for trace active since the trace service module will do this and return (to the instruction following the halfword trace point number), if trace is not active (with no error indication).

TRACE POINT NUMBER ASSIGNMENTS

- DBSS is assigned trace point numbers 0 through 3999. They are suballocated to the functions/subcomponents of DBSS as follows:
  - DBSS Entry (ENTRY) page 387 — 0 - 149
  - DBSS Exit (EXIT) page 390 — 150 - 299
  - Log/Recovery component (LOG) page 393 — 300 - 599
  - Lock Management (LOCK) page 396 — 600 - 899
  - Reserved for LUW Transaction (LUW) — 900 - 1199
  - Data Control (DC) page 399 — 1200 - 1499
  - Data Manipulation (DM) page 406 — 1500 - 1999
  - Storage (I/O) component (STOR) page 418 — 2000 - 2299
  - Sort component (SORT) page 424 — 2300 - 2599
  - Index component (INDEX) page 428 — 2600 - 2899
  - Update Statistics component (STAT) page 433 — 2900 - 2999
  - DBSS reserved — 3000 - 3999
- RDS is assigned trace point numbers 4000 through 7999. They are suballocated to the functions/subcomponets of RDS as follows:
  - Executives (EXEC) page 438 — 4000 - 4399
  - Parser (PA) page 444 — 4400 - 4599
  - Optimizer (OPT) page 445 — 4600 - 5199
  - Code Generator (CODE) page 448 — 5200 - 5799
  - Interpreter and Authorization (INT) page 454 — 5800 - 6399
  - Security/Audit (AU) page 484 — 6400 - 6415
  - RDS reserved — 6420 - 7999
- SQL/DS reserved — 8000 - 9899
- Service Temporary
  - DBSS Service Temporary page 435 — 9900 - 9949
  - RDS Service Temporary page 487 — 9950 - 9999

  These trace point numbers are intended for inserting temporary trace points for problems which cannot be isolated with permanent trace points.

ARIYE15 (DBSS Entry and DBSS Exit Trace Point Descriptor)

Trace point: 0000
Subcomponent: DBSS Entry
Module: ARIYM00
Trace level: 1 or 2
Type: DBSS entry
Description: Invoked when DBSS is called with an invalid OPCODE. Indicates invalid OPCODE on DBSS entry and displays parameters pointer.

Trace point: 0001
Subcomponent: DBSS Entry
Module: ARIYE01
Trace level: 1 or 2
Type: DBSS entry
Description: Invoked when DBSS is called and DBSS entry trace level is 1 or there is no OPCODE-specific trace handling routine. Indicates DBSS entry and displays OPCODE name.

Trace point: 0002
Subcomponent: Storage I/O
Module: ARIYI03
Trace Level: 2
Type: DBSS entry
Description: Invoked when DBSS is called and DBSS entry trace level is 2. Displays input parameters for DBSPACE operations, ACQUIRE and RELEASE.

Trace Point: 0003
Subcomponent: Sort
Module: ARIYS25
Trace Level: 2
Type: DBSS entry
Description: Invoked when DBSS is called and DBSS entry trace level is 2. Displays input parameters and BASE and auxiliary structures for SORT call. (Structure BASE in copy macro YUSER)

```
Trace Point:  0004
Subcomponent: Data Manipulation
Module:       ARIYD81
Trace Level:  2
Type:         DBSS entry
Description:  Invoked when DBSS is called and DBSS entry
              trace level is 2.  Displays input parameters
              and BASE and auxiliary structures for data
              manipulation call. (Structure BASE in copy
              macro YUSER)

Trace Point:  0005
Subcomponent: Lock
Module:       ARIYD812
Trace Level:  2
Type:         DBSS entry
Description:  Invoked when DBSS is called and DBSS entry
              trace level is 2.  Displays input parameters
              and base structure for LOCK, UNLOCK call.
              (Structure LBASE in copy macro YRSSLOCK)

Trace Point:  0010
Subcomponent: Data Control
Module:       ARIYC08
Trace Level:  2
Type:         DBSS entry (insert control record)
Description:  DBSS entry to ARIYC02 (CBASE and SCR)

Trace Point:  0011
Subcomponent: Data Control
Module:       ARIYC08
Trace Level:  2
Type:         DBSS entry (insert control record)
Description:  DBSS entry to ARIYC02 (CBASE and MCR)

Trace point:  0012
Subcomponent: Data Control
Module:       ARIYC08
Trace level:  2
Type:         DBSS entry (insert control record)
Description:  DBSS entry to ARIYC02 (CBASE and LCR)

Trace point:  0013
Subcomponent: Data Control
Module:       ARIYC08
Trace level:  2
Type:         DBSS entry (insert control record)
Description:  DBSS entry to ARIYC02 (CBASE and ICR)

Trace point:  0014
Subcomponent: Data Control
Module:       ARIYC08
Trace level:  2
Type:         DBSS entry (update control record)
Description:  DBSS entry to ARIYC10 (CBASE and SCR)

Trace point:  0015
Subcomponent: Data Control
Module:       ARIYC08
Trace level:  2
Type:         DBSS entry (update control record)
Description:  DBSS entry to ARIYC10 (CBASE and MCR)

Trace point:  0016
Subcomponent: Data Control
Module:       ARIYC08
Trace level:  2
Type:         DBSS entry (update control record)
Description:  DBSS entry to ARIYC10 (CBASE and ICR)

Trace point:  0017
Subcomponent: Data Control
Module:       ARIYC08
Trace level:  2
Type:         DBSS entry (fetch next control record)
Description:  DBSS entry to ARIYC00 (CBASE and LCR)

Trace point:  0018
Subcomponent: Data Control
Module:       ARIYC08
Trace level:  2
Type:         DBSS entry (fetch control record)
Description:  DBSS entry to ARIYC01 (CBASE and LCR)

Trace point:  0019
Subcomponent: Data Control
Module:       ARIYC08
Trace level:  2
Type:         DBSS entry (delete control record)
Description:  DBSS entry to ARIYC11 (CBASE and LCR)

Trace point:  0020
Subcomponent: Data Control
Module:       ARIYC08
Trace level:  2
Type:         DBSS entry (fetch next control record)
Description:  DBSS entry to ARIYC00 (CBASE and PLCR)
```

Trace point: 0021
Subcomponent: Data Control
Module: ARIYC08
Trace level: 2
Type: DBSS entry (fetch control record)
Description: DBSS entry to ARIYC01 (CBASE and PLCR)

Trace point: 0022
Subcomponent: Data Control
Module: ARIYC08
Trace level: 2
Type: DBSS entry (delete control record)
Description: DBSS entry to ARIYC11 (CBASE and PLCR)

Trace point: 0023
Subcomponent: Data Control
Module: ARIYC08
Trace level: 2
Type: DBSS entry (fetch control record)
Description: DBSS entry to ARIYC01 (CBASE and ICR)

Trace point: 0024
Subcomponent: Data Control
Module: ARIYC08
Trace level: 2
Type: DBSS entry (fetch next control record)
Description: DBSS entry to ARIYC00 (CBASE and ICR)

Trace point: 0025
Subcomponent: Data Control
Module: ARIYC08
Trace level: 2
Type: DBSS entry (delete control record)
Description: DBSS entry to ARIYC11 (CBASE and ICR)

Trace point: 0026
Subcomponent: Data Control
Module: ARIYC08
Trace level: 2
Type: DBSS entry (fetch next control record)
Description: DBSS entry to ARIYC00 (CBASE and MCR)

Trace point: 0027
Subcomponent: Data Control
Module: ARIYC08
Trace level: 2
Type: DBSS entry (fetch control record)
Description: DBSS entry to ARIYC01 (CBASE and MCR)

Trace point: 0028
Subcomponent: Data Control
Module: ARIYC08
Trace level: 2
Type: DBSS entry (fetch control record)
Description: DBSS entry to ARIYC01 (CBASE and SCR)

Trace point: 0030
Subcomponent: Work Management
Module: ARIYT08
Trace level: 2
Type: DBSS entry
Description: Invoked when DBSS is called and DBSS entry trace level is 2 and the operation code is BEGIN WORK or COMMIT or RETRIEVE IN-DOUBT LIST. Display opcode name and agent TPMAP (LUW status).

Trace point: 0031
Subcomponent: Work Management
Module: ARIYT08
Trace level: 2
Type: DBSS entry
Description: Invoked when DBSS is called and DBSS entry trace level is 2 and the operation code is PREPARE TO COMMIT. Displays opcode name, RMRE (Resource Manager Recovery List), and agent TPMAP (LUW status).

Trace point: 0032
Subcomponent: Work Management
Module: ARIYT08
Trace level: 2
Type: DBSS entry
Description: Invoked when DBSS is called and DBSS entry trace level is 2 and the operation code is ROLLBACK or SAVE WORK. Displays opcode name, SAVENO (field in TBASE), and agent TPMAP (LUW status).

Trace point: 0033
Subcomponent: Work Management
Module: ARIYT08
Trace level: 2
Type: DBSS entry
Description: Invoked when DBSS is called and DBSS entry trace level is 2 and the operation code is SCHEDULE USERID. Displays opcode name, LDATA and TBAEXTID (fields in TBASE), and agent TPMAP (LUW status).

Trace point: 0034
Subcomponent: Work Management
Module: ARIYT08
Trace level: 2
Type: DBSS entry
Description: Invoked when DBSS is called and DBSS entry trace level is 2 and the operation code is TERMINATE AGENT. Displays Opcode name, LDATA (field in TBASE), and agent TPMAP (LUW status).

Trace point: 0035
Subcomponent: Operator Communication
Module: ARIYT08
Trace level: 2
Type: DBSS entry
Description: Invoked when DBSS is called and DBSS entry trace level is 2 and the operation code is PROCESS OPERATOR COMMAND. Displays opcode name, OBAAUXPA and OBAEOPCT and OBACMMD (fields in OBASE parameter structure).

Trace Point: 0150
Subcomponent: DBSS Exit
Module: ARIYM00
Trace level: 1 or 2
Type: DBSS exit
Description: Invoked when DBSS is called with an invalid opcode. Indicates invalid opcode on DBSS exit and displays return code and parameters pointer.

Trace point: 0151
Subcomponent: DBSS Exit
Module: ARIYE01
Trace level: 1 or 2
Type: DBSS exit
Description: Invoked when DBSS is called and DBSS exit trace level is 1 or there is no opcode-specific trace handling routine. Indicates DBSS exit and displays opcode name and return code.

Trace point: 0152
Subcomponent: Storage (I/O) Component
Module: ARIYI031
Trace level: 2
Type: DBSS exit
Description: Invoked when DBSS is called and DBSS exit trace level is 2. Displays output parameters for DBSPACE operations, ACQUIRE and RELEASE.

Trace point: 0153
Subcomponent: Sort
Module: ARIYS251
Trace level: 2
Type: DBSS exit
Description: Invoked when DBSS is called and DBSS exit trace level is 2. Displays output parameters for SORT operations (from structure BASE and auxiliary structures).

Trace point: 0154
Subcomponent: Data Manipulation
Module: ARIYD811
Trace level: 2
Type: DBSS exit
Description: Invoked when DBSS is called and DBSS exit trace level is 2. Displays output parameters for data manipulation operations. (Structure BASE and auxiliary structures in copy macro YUSER)

Trace point: 0155
Subcomponent: Lock Management
Module: ARIYD813
Trace level: 2
Type: DBSS exit
Description: Invoked when DBSS is called and DBSS exit trace level is 2. Displays output parameters for LOCK/UNLOCK operations. (Structure LBASE in copy macro YRSSLOCK)

Trace point: 0160
Subcomponent: Data Control
Module: ARIYC081
Trace level: 2
Type: DBSS exit (fetch next control record)
Description: DBSS exit from ARIYC00 (CBASE)

Trace point: 0161
Subcomponent: Data Control
Module: ARIYC081
Trace level: 2
Type: DBSS exit (fetch control record)
Description: DBSS exit from ARIYC01 (CBASE)

Trace point: 0162
Subcomponent: Data Control
Module: ARIYC081
Trace level: 2
Type: DBSS exit (insert control record)
Description: DBSS exit from ARIYC02 (CBASE)

```
Trace point:  0163
Subcomponent: Data Control
Module:       ARIYC081
Trace level:  2
Type:         DBSS exit (update control record)
Description:  DBSS exit from ARIYC10 (CBASE)

Trace point:  0164
Subcomponent: Data Control
Module:       ARIYC081
Trace level:  2
Type:         DBSS exit (delete control record)
Description:  DBSS exit from ARIYC11 (CBASE)

Trace point:  0165
Subcomponent: Data Control
Module:       ARIYC081
Trace level:  2
Type:         DBSS exit (fetch control record)
Description:  DBSS exit from ARIYC01 (CBASE and SCR)

Trace point:  0166
Subcomponent: Data Control
Module:       ARIYC081
Trace level:  2
Type:         DBSS exit (fetch next control record)
Description:  DBSS exit from ARIYC00 (CBASE and SCR)

Trace point:  0167
Subcomponent: Data Control
Module:       ARIYC081
Trace level:  2
Type:         DBSS exit (update control record)
Description:  DBSS exit from ARIYC10 (CBASE and SCR)

Trace point:  0168
Subcomponent: Data Control
Module:       ARIYC081
Trace level:  2
Type:         DBSS exit (fetch next control record)
Description:  DBSS exit from ARIYC00 (CBASE and MCR)

Trace point:  0169
Subcomponent: Data Control
Module:       ARIYC081
Trace level:  2
Type:         DBSS exit (fetch control record)
Description:  DBSS exit from ARIYC01 (CBASE and MCR)
```

```
Trace point:  0170
Subcomponent: Data Control
Module:       ARIYC081
Trace level:  2
Type:         DBSS exit (update control record)
Description:  DBSS exit from ARIYC10 (CBASE and MCR)

Trace point:  0171
Subcomponent: Data Control
Module:       ARIYC081
Trace level:  2
Type:         DBSS exit (fetch next control record)
Description:  DBSS exit from ARIYC00 (CBASE and LCR)

Trace point:  0172
Subcomponent: Data Control
Module:       ARIYC081
Trace level:  2
Type:         DBSS exit (fetch control record)
Description:  DBSS exit from ARIYC01 (CBASE and LCR)

Trace point:  0173
Subcomponent: Data Control
Module:       ARIYC081
Trace level:  2
Type:         DBSS exit (update control record)
Description:  DBSS exit from ARIYC10 (CBASE and LCR)

Trace point:  0174
Subcomponent: Data Control
Module:       ARIYC081
Trace level:  2
Type:         DBSS exit (fetch next control record)
Description:  DBSS exit from ARIYC00 (CBASE and PLCR)

Trace point:  0175
Subcomponent: Data Control
Module:       ARIYC081
Trace level:  2
Type:         DBSS exit (fetch control record)
Description:  DBSS exit from ARIYC01 (CBASE and PLCR)

Trace point:  0176
Subcomponent: Data Control
Module:       ARIYC081
Trace level:  2
Type:         DBSS exit (update control record)
Description:  DBSS exit from ARIYC10 (CBASE and PLCR)
```

Trace point: 0177
Subcomponent: Data Control
Module: ARIYC081
Trace level: 2
Type: DBSS exit (fetch next control record)
Description: DBSS exit from ARIYC00 (CBASE and ICR)

Trace point: 0178
Subcomponent: Data Control
Module: ARIYC081
Trace level: 2
Type: DBSS exit (fetch control record)
Description: DBSS exit from ARIYC01 (CBASE and ICR)

Trace point: 0179
Subcomponent: Data Control
Module: ARIYC081
Trace level: 2
Type: DBSS exit (update control record)
Description: DBSS exit from ARIYC10 (CBASE and ICR)

Trace point: 0180
Subcomponent: Data Control
Module: ARIYC081
Trace level: 2
Type: DBSS exit (insert control record)
Description: DBSS exit from ARIYC02 (CBASE and SCR)

Trace point: 0181
Subcomponent: Data Control
Module: ARIYC081
Trace level: 2
Type: DBSS exit (insert control record)
Description: DBSS exit from ARIYC02 (CBASE and MCR)

Trace point: 0182
Subcomponent: Data Control
Module: ARIYC081
Trace level: 2
Type: DBSS exit (insert control record)
Description: DBSS exit from ARIYC02 (CBASE and LCR)

Trace point: 0183
Subcomponent: Data Control
Module: ARIYC081
Trace level: 2
Type: DBSS exit (insert control record)
Description: DBSS exit from ARIYC02 (CBASE and ICR)

Trace point: 0190
Subcomponent: Work Management
Module: ARIYT08
Trace level: 2
Type: DBSS exit
Description: Invoked when DBSS is returning to its caller and DBSS exit trace level is 2 and the operation code is COMMIT WORK or ROLLBACK WORK. Displays opcode name, DBSS return code, TRANSACT (field in TBASE parameter structure), and agent TPMAP (LUW status).

Trace point: 0191
Subcomponent: Work Management
Module: ARIYT08
Trace level: 2
Type: DBSS exit
Description: Invoked when DBSS is returning to its caller and DBSS exit trace level is 2 and the operation code is BEGIN WORK, SAVE WORK, PREPARE TO COMMIT, SCHEDULE USERID, INIT AGENT, or TERMINATE AGENT. Displays opcode name, DBSS return code, and agent TPMAP (LUW status).

Trace point: 0192
Subcomponent: Work Management
Module: ARIYT08
Trace level: 2
Type: DBSS exit
Description: Invoked when DBSS is returning to its caller and DBSS exit trace level is 2 and the operation code is RETRIEVE in-doubt list. Displays opcode name, DBSS return code, RMRE (Resource Manager Recovery List), and agent TPMAP (LUW status).

Trace point: 0193
Subcomponent: Operator Communication
Module: ARIYT08
Trace level: 2
Type: DBSS exit
Description: Invoked when DBSS is returning to its caller and DBSS exit trace level is 2 and the operation code is PROCESS OPERATOR command. Displays opcode name, DBSS return code, OBATOTCT and OBAFDBK1 (fields in OBASE parameter list structure).

Trace point: 0195
Subcomponent: Update Statistics
Module: ARIYZ27
Trace level: 2
Type: DBSS exit
Description: Invoked when DBSS is returning to its caller and DBSS exit trace level is 2 and the operation code is END UPDATE STATISTICS. Displays opcode name, DBSS return code, NREL, NSEG, RELPTR, SEGPTR, REQORPH, REQSTSEG, LORCODE, LDFBACK1-4, RELS ARRAY, and SEGMS ARRAY. (fields in LDBASE parameter structure and output from update statistics - see ZUSER copy macro).

ARIYL25 (DBSS Log/Recovery Component Trace Point Descriptor)

Trace point: 0300
Subcomponent: Log/Recovery
Module: ARIYL00
Trace level: 1
Type: Module entry
Description: Displays input parameter COLDFLAG, TPMAP fields TLOGBEG and TLOGEND, YTABLE1 TLOGxxx fields, and YRSSCVT SLOGxxx fields (all level 2).

Trace point: 0301
Subcomponent: Log/Recovery
Module: ARIYL00
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 0302
Subcomponent: Log/Recovery
Module: ARIYL01
Trace level: 1
Type: Module entry
Description: Displays input parameters LEN and TYPE, TPMAP fields TLOGBEG and TLOGEND, YTABLE1 TLOGxxx fields, and YRSSCVT SLOGxxx fields (all level 2).

Trace point: 0303
Subcomponent: Log/Recovery
Module: ARIYL01
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 0304
Subcomponent: Log/Recovery
Module: ARIYL02
Trace level: 1
Type: Module entry
Description: Displays input parameters LEN and SOURCE, TPMAP fields TLOGBEG and TLOGEND, YTABLE1 TLOGxxx fields, and YRSSCVT SLOGxxx fields (all level 2).

Trace point: 0305
Subcomponent: Log/Recovery
Module: ARIYL02
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 0306
Subcomponent: Log/Recovery
Module: ARIYL03
Trace level: 1
Type: Module entry
Description: Displays input parameter FLUSH, TPMAP fields TLOGBEG and TLOGEND, YTABLE1 TLOGxxx fields, and YRSSCVT SLOGxxx fields (all level 2).

Trace point: 0307
Subcomponent: Log/Recovery
Module: ARIYL03
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 0308
Subcomponent: Log/Recovery
Module: ARIYL04
Trace level: 1
Type: Module entry
Description: Displays input parameter SADDR, TPMAP fields TLOGBEG and TLOGEND, YTABLE1 TLOGxxx fields, and YRSSCVT SLOGxxx fields (all level 2).

Trace point: 0309
Subcomponent: Log/Recovery
Module: ARIYL04
Trace level: 1
Type: Module exit
Description: Displays return code and output parameter PREDADDR (level 2).

Trace point: 0310
Subcomponent: Log/Recovery
Module: ARIYL05
Trace level: 1
Type: Module entry
Description: Displays input parameter LEN, TPMAP fields TLOGBEG and TLOGEND, YTABLE1 TLOGxxx fields, and YRSSCVT SLOGxxx fields (all level 2).

Trace point: 0311
Subcomponent: Log/Recovery
Module: ARIYL05
Trace level: 1
Type: Module exit
Description: Displays return code and output parameter TARGET (level 2).

Trace point: 0312
Subcomponent: Log/Recovery
Module: ARIYL06
Trace level: 1
Type: Module entry
Description: Displays input parameter DELTA, TPMAP fields TLOGBEG and TLOGEND, and YTABLE1 field TLOGCURS (all level 2)

Trace point: 0313
Subcomponent: Log/Recovery
Module: ARIYL06
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 0314
Subcomponent: Log/Recovery
Module: ARIYL07
Trace level: 1
Type: Module entry
Description: Displays input parameters LEN and TYPE, TPMAP fields TLOGBEG and TLOGEND, YTABLE1 TLOGxxx fields, and YRSSCVT SLOGxxx fields (all level 2).

Trace point: 0315
Subcomponent: Log/Recovery
Module: ARIYL07
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 0316
Subcomponent: Log/Recovery
Module: ARIYL08
Trace level: 1
Type: Module entry
Description: Displays input parameter REASON, TPMAP fields TLOGBEG and TLOGEND, YTABLE1 TLOGxxx fields, and YRSSCVT SLOGxxx fields (all level 2).

Trace point: 0317
Subcomponent: Log/Recovery
Module: ARIYL08
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 0318
Subcomponent: Log/Recovery
Module: ARIYL09
Trace level: 1
Type: Module entry
Description: Displays no variables

Trace point: 0319
Subcomponent: Log/Recovery
Module: ARIYL09
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 0320
Subcomponent: Log/Recovery
Module: ARIYL10
Trace level: 1
Type: Module entry
Description: Displays YTABLE1 fields TLOGCURS and TLOGPAGE (both level 2).

Trace point: 0321
Subcomponent: Log/Recovery
Module: ARIYL10
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 0322
Subcomponent: Log/Recovery
Module: ARIYL11
Trace level: 1
Type: Module entry
Description: Displays input parameter ENDOFLOG (level 2).

```
Trace point:  0323
Subcomponent: Log/Recovery
Module:       ARIYL11
Trace level:  1
Type:         Module exit
Description:  Displays return code

Trace point:  0324
Subcomponent: Log/Recovery
Module:       ARIYL12
Trace level:  1
Type:         Module entry
Description:  Displays input parameters UWADDR, UWOFFSET and
              UWVALUE (all level 2).

Trace point:  0325
Subcomponent: Log/Recovery
Module:       ARIYL12
Trace level:  1
Type:         Module exit
Description:  Displays return code

Trace point:  0326
Subcomponent: Log/Recovery
Module:       ARIYL13
Trace level:  1
Type:         Module entry
Description:  Displays no variables

Trace point:  0327
Subcomponent: Log/Recovery
Module:       ARIYL13
Trace level:  1
Type:         Module exit
Description:  Displays return code

Trace point:  0328
Subcomponent: Log/Recovery
Module:       ARIYL19
Trace level:  1
Type:         Module entry
Description:  Displays input parameter LOGPAGEN (level 2).

Trace point:  0329
Subcomponent: Log/Recovery
Module:       ARIYL19
Trace level:  1
Type:         Module exit
Description:  Displays return code

Trace point:  0330
Subcomponent: Log/Recovery
Module:       ARIYL21
Trace level:  1
Type:         Module entry
Description:  Displays no variables

Trace point:  0331
Subcomponent: Log/Recovery
Module:       ARIYL21
Trace level:  1
Type:         Module exit
Description:  Displays return code

Trace point:  0332
Subcomponent: Log/Recovery
Module:       ARIYL22
Trace level:  1
Type:         Module entry
Description:  Displays no variables

Trace point:  0333
Subcomponent: Log/Recovery
Module:       ARIYL22
Trace level:  1
Type:         Module exit
Description:  Displays return code

Trace point:  0334
Subcomponent: Log/Recovery
Module:       ARIYL23
Trace level:  1
Type:         Module entry
Description:  Displays no variables

Trace point:  0335
Subcomponent: Log/Recovery
Module:       ARIYL23
Trace level:  1
Type:         Module exit
Description:  No return code

Trace point:  0336
Subcomponent: Log/Recovery
Module:       ARIYL24
Trace level:  1
Type:         Module entry
Description:  Displays no variables
```

Trace point: 0337
Subcomponent: Log/Recovery
Module: ARIYL24
Trace level: 1
Type: Module exit
Description: No return code

ARIYK43 (DBSS Lock Management Component Trace Point Descriptor)

Trace point: 0600
Subcomponent: Lock Management
Module: ARIYK00
Trace level: 1
Type: Module entry
Description: Displays GATENAM and GATENAME (level 2)

Trace point: 0601
Subcomponent: Lock Management
Module: ARIYK00
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 0602
Subcomponent: Lock Management
Module: ARIYK01
Trace level: 1
Type: Module entry
Description: Displays LSEGMENT, LRID, LTID and LMODE (all level 2)

Trace point: 0603
Subcomponent: Lock Management
Module: ARIYK01
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 0604
Subcomponent: Lock Management
Module: ARIYK02
Trace level: 1
Type: Module entry
Description: Displays LOCKNAME (level 2)

Trace point: 0605
Subcomponent: Lock Management
Module: ARIYK02
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 0606
Subcomponent: Lock Management
Module: ARIYK03
Trace level: 1
Type: Module entry
Description: Displays GNAME (level 2)

Trace point: 0607
Subcomponent: Lock Management
Module: ARIYK03
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 0608
Subcomponent: Lock Management
Module: ARIYK04
Trace level: 1
Type: Module entry
Description: Displays LATCHPTR (level 2)

Trace point: 0609
Subcomponent: Lock Management
Module: ARIYK04
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 0610
Subcomponent: Lock Management
Module: ARIYK05
Trace level: 1
Type: Module entry
Description: Displays LATCHPTR (level 2)

Trace point: 0611
Subcomponent: Lock Management
Module: ARIYK05
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 0612
Subcomponent: Lock Management
Module: ARIYK06
Trace level: 1
Type: Module entry
Description: Displays LATCHPTR and MODE (both level 2)

Trace point: 0613
Subcomponent: Lock Management
Module: ARIYK06
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 0614
Subcomponent: Lock Management
Module: ARIYK07
Trace level: 1
Type: Module entry
Description: Displays LATCHPTR (level 2)

Trace point: 0615
Subcomponent: Lock Management
Module: ARIYK07
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 0616
Subcomponent: Lock Management
Module: ARIYK08
Trace level: 1
Type: Module entry
Description: Displays no variables

Trace point: 0617
Subcomponent: Lock Management
Module: ARIYK08
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 0618
Subcomponent: Lock Management
Module: ARIYK09
Trace level: 1
Type: Module entry
Description: Displays no variables

Trace point: 0619
Subcomponent: Lock Management
Module: ARIYK09
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 0620
Subcomponent: Lock Management
Module: ARIYK10
Trace level: 1
Type: Module entry
Description: Displays no variables

Trace point: 0621
Subcomponent: Lock Management
Module: ARIYK10
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 0622
Subcomponent: Lock Management
Module: ARIYK11
Trace level: 1
Type: Module entry
Description: Displays no variables

Trace point: 0623
Subcomponent: Lock Management
Module: ARIYK11
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 0624
Subcomponent: Lock Management
Module: ARIYK12
Trace level: 1
Type: Module entry
Description: Displays no variables

Trace point: 0625
Subcomponent: Lock Management
Module: ARIYK12
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 0626
Subcomponent: Lock Management
Module: ARIYK18
Trace level: 1
Type: Module entry
Description: Displays WAITTYPE (level 2)

Trace point: 0627
Subcomponent: Lock Management
Module: ARIYK18
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 0628
Subcomponent: Lock Management
Module: ARIYK19
Trace level: 1
Type: Module entry
Description: Displays PROK (level 2)

Trace point: 0629
Subcomponent: Lock Management
Module: ARIYK19
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 0630
Subcomponent: Lock Management
Module: ARIYK21
Trace level: 1
Type: Module entry
Description: Displays GATENAM (level 2)

Trace point: 0631
Subcomponent: Lock Management
Module: ARIYK21
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 0632
Subcomponent: Lock Management
Module: ARIYK22
Trace level: 1
Type: Module entry
Description: Displays no variables

Trace point: 0633
Subcomponent: Lock Management
Module: ARIYK22
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 0634
Subcomponent: Lock Management
Module: ARIYK23
Trace level: 1
Type: Module entry
Description: Displays no variables

Trace point: 0635
Subcomponent: Lock Management
Module: ARIYK23
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 0636
Subcomponent: Lock Management
Module: ARIYK24
Trace level: 1
Type: Module entry
Description: Displays no variables

Trace point: 0637
Subcomponent: Lock Management
Module: ARIYK24
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 0638
Subcomponent: Lock Management
Module: ARIYK25
Trace level: 1
Type: Module entry
Description: Displays no variables

Trace point: 0639
Subcomponent: Lock Management
Module: ARIYK25
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 0640
Subcomponent: Lock Management
Module: ARIYK26
Trace level: 1
Type: Module entry
Description: Displays no variables

Trace point: 0641
Subcomponent: Lock Management
Module: ARIYK26
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 0642
Subcomponent: Lock Management
Module: ARIYK27
Trace level: 1
Type: Module entry
Description: Displays no variables

Trace point: 0643
Subcomponent: Lock Management
Module: ARIYK27
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 0644
Subcomponent: Lock Management
Module: ARIYK39
Trace level: 1
Type: Module entry
Description: Displays SEGN (level 2)

Trace point: 0645
Subcomponent: Lock Management
Module: ARIYK39
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 0650
Subcomponent: Lock Management
Module: ARIYK40
Trace level: 1
Type: Module entry
Description: Displays INSEGM and HEADER (both level 2)

Trace point: 0651
Subcomponent: Lock Management
Module: ARIYK40
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 0652
Subcomponent: Lock Management
Module: ARIYK41
Trace level: 1
Type: Module entry
Description: Displays GNAME, HISMODE, HISDUR and CONTROL (all level 2)

Trace point: 0653
Subcomponent: Lock Management
Module: ARIYK41
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 0654
Subcomponent: Lock Management
Module: ARIYK42
Trace level: 1
Type: Module entry
Description: Displays no variables

Trace point: 0655
Subcomponent: Lock Management
Module: ARIYK42
Trace level: 1
Type: Module exit
Description: Displays return code

ARIYC27 (DBSS Data Control Component Trace Point Descriptor)

Trace point: 1200
Subcomponent: Data Control
Module: ARIYC00
Trace level: 1
Type: Module entry
Description: Displays CBASE pointer (level 2)

Trace point: 1201
Subcomponent: Data Control
Module: ARIYC00
Trace level: 2
Type: Module report
Description: Displays input parameters (data in CBASE structure)

Trace point: 1202
Subcomponent: Data Control
Module: ARIYC00
Trace level: 2
Type: Module report
Description: Displays input parameters: DBSPACE description auxiliary structure (SCR) pointed to by AUXPTR in CBASE structure (YP->CBASE->AUXPTR->SCR)

Trace point: 1203
Subcomponent: Data Control
Module: ARIYC00
Trace level: 2
Type: Module report
Description: Displays input parameters: description of Type 1 row auxiliary structure (MCR1) pointed to by AUXPTR in CBASE structure (YP->CBASE->AUXPTR->MCR1)

Trace point: 1204
Subcomponent: Data Control
Module: ARIYC00
Trace level: 2
Type: Module report
Description: Displays input parameters: description of 1 link (child-LCR) auxiliary structure (LCR) pointed to by AUXPTR in CBASE structure (YP->CBASE->AUXPTR->LCR)

Trace point: 1205
Subcomponent: Data Control
Module: ARIYC00
Trace level: 2
Type: Module report
Description: Displays input parameters: description of 1 link (parent-PLCR) auxiliary structure (PLCR) pointed to by AUXPTR in CBASE structure (YP->CBASE->AUXPTR->PLCR)

Trace point: 1206
Subcomponent: Data Control
Module: ARIYC00
Trace level: 2
Type: Module report
Description: Displays input parameters: description of 1 index auxiliary structure (ICR) pointed to by AUXPTR in CBASE structure (YP->CBASE->AUXPTR->ICR)

Trace point: 1207
Subcomponent: Data Control
Module: ARIYC00
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 1210
Subcomponent: Data Control
Module: ARIYC01
Trace level: 1
Type: Module entry
Description: Displays pointer to CBASE structure (level 2)

Trace point: 1211
Subcomponent: Data Control
Module: ARIYC01
Trace level: 2
Type: Module report
Description: Displays input parameters: data in CBASE structure

Trace point: 1212
Subcomponent: Data Control
Module: ARIYC01
Trace level: 2
Type: Module report
Description: Displays input parameters: DBSPACE description auxiliary structure (SCR) pointed to by AUXPTR in CBASE structure (YP->CBASE->AUXPTR->SCR)

Trace point: 1213
Subcomponent: Data Control
Module: ARIYC01
Trace level: 2
Type: Module report
Description: Displays input parameters: description of Type 1 row auxiliary structure (MCR1) pointed to by AUXPTR in CBASE structure (YP->CBASE->AUXPTR->MCR1)

Trace point: 1214
Subcomponent: Data Control
Module: ARIYC01
Trace level: 2
Type: Module report
Description: Displays input parameters: description of 1 link (child-LCR) auxiliary structure (LCR) pointed to by auxptr in CBASE structure (YP->CBASE->AUXPTR->LCR)

Trace point: 1215
Subcomponent: Data Control
Module: ARIYC01
Trace level: 2
Type: Module report
Description: Displays input parameters: description of 1 link (parent-PLCR) auxiliary structure (LCR) pointed to by AUXPTR in CBASE structure (YP->CBASE->AUXPTR->PLCR)

Trace point: 1216
Subcomponent: Data Control
Module: ARIYC01
Trace level: 2
Type: Module report
Description: Displays input parameters: description of 1 image (INDEX) auxiliary structure (ICR) pointed to by AUXPTR in CBASE structure (YP->CBASE->AUXPTR->ICR)

Trace point: 1217
Subcomponent: Data Control
Module: ARIYC01
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 1220
Subcomponent: Data Control
Module: ARIYC02
Trace level: 1
Type: Module entry
Description: Displays pointer to CBASE structure (level 2)

Trace point: 1221
Subcomponent: Data Control
Module: ARIYC02
Trace level: 2
Type: Module report
Description: Displays input parameters: data in CBASE structure

Trace point: 1222
Subcomponent: Data Control
Module: ARIYC02
Trace level: 2
Type: Module report
Description: Displays input parameters: DBSPACE description auxiliary structure (SCR) pointed to by AUXPTR in CBASE structure (YP->CBASE->AUXPTR->SCR)

Trace point: 1223
Subcomponent: Data Control
Module: ARIYC02
Trace level: 2
Type: Module report
Description: Displays input parameters: description of Type 1 row auxiliary structure (MCR1) pointed to by AUXPTR in CBASE structure (YP->CBASE->AUXPTR->MCR1)

Trace point: 1224
Subcomponent: Data Control
Module: ARIYC02
Trace level: 2
Type: Module report
Description: Displays input parameters: description of 1 link (child-LCR) auxiliary structure (LCR) pointed to by AUXPTR in CBASE structure (YP->CBASE->AUXPTR->LCR)

Trace point: 1226
Subcomponent: Data Control
Module: ARIYC02
Trace level: 2
Type: Module report
Description: Displays input parameters: description of 1 index auxiliary structure (ICR) pointed to by AUXPTR in CBASE structure (YP->CBASE->AUXPTR->ICR)

Trace point: 1227
Subcomponent: Data Control
Module: ARIYC02
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 1230
Subcomponent: Data Control
Module: ARIYC03
Trace level: 1
Type: Module entry
Description: Displays input parameters: PNI and HLGTH (both level 2)

Trace point: 1231
Subcomponent: Data Control
Module: ARIYC03
Trace level: 2
Type: Module report
Description: Displays parameter returned from call to ARIYC03, RTID

Trace point: 1232
Subcomponent: Data Control
Module: ARIYC03
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 1235
Subcomponent: Data Control
Module: ARIYC03 (ENTRY POINT ARIYC031)
Trace level: 1
Type: Module entry
Description: Displays input parameters: R2PTR, NRETID, and LNGTH (all level 2)

Trace point: 1236
Subcomponent: Data Control
Module: ARIYC03 (Exit from entry point ARIYC031)
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 1240
Subcomponent: Data Control
Module: ARIYC05
Trace level: 1
Type: Module entry
Description: Displays fields in CBASE structure and LCR auxiliary structure (all level 2)

Trace point: 1241
Subcomponent: Data Control
Module: ARIYC05
Trace level: 2
Type: Module report
Description: Displays input parameters: CLINKSC and CLINKSP (number of links)

Trace point: 1242
Subcomponent: Data Control
Module: ARIYC05
Trace level: 2
Type: Module report
Description: Displays module output: PLINKP (child link ID)

Trace point: 1248
Subcomponent: Data Control
Module: ARIYC05
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 1250
Subcomponent: Data Control
Module: ARIYC07
Trace level: 1
Type: Module entry
Description: Displays input values: TRANMODE from YTABLE1 and CQUALF from CBASE (both level 2)

Trace point: 1255
Subcomponent: Data Control
Module: ARIYC07
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 1256
Subcomponent: Data Control
Module: ARIYC09
Trace level: 1
Type: Module entry
Description: Displays input parameters: PAGENO, PAGEPTR, and BLOCKPTR (all level 2)

Trace point: 1259
Subcomponent: Data Control
Module: ARIYC09
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 1260
Subcomponent: Data Control
Module: ARIYC10
Trace level: 1
Type: Module entry
Description: Displays input parameter: P (pointer to CBASE) (level 2)

Trace point: 1261
Subcomponent: Data Control
Module: ARIYC10
Trace level: 2
Type: Module report
Description: Displays input parameters: fields in CBASE structure

Trace point: 1262
Subcomponent: Data Control
Module: ARIYC10
Trace level: 2
Type: Module report
Description: Displays input parameters: DBSPACE description auxiliary structure (SCR) pointed to by AUXPTR in CBASE structure (YP->CBASE->AUXPTR->SCR)

Trace point: 1263
Subcomponent: Data Control
Module: ARIYC10
Trace level: 2
Type: Module report
Description: Displays input parameters: description of type 1 table auxiliary structure (MCR1) pointed to by AUXPTR in CBASE structure (YP->CBASE->AUXPTR->MCR1)

Trace point: 1264
Subcomponent: Data Control
Module: ARIYC10
Trace level: 2
Type: Module report
Description: Displays input parameters: description of 1 index auxiliary structure (ICR) pointed to by AUXPTR IN CBASE structure (YP->CBASE->AUXPTR->ICR)

Trace point: 1269
Subcomponent: Data Control
Module: ARIYC10
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 1270
Subcomponent: Data Control
Module: ARIYC11
Trace level: 1
Type: Module entry
Description: Displays input parameter: P (pointer to CBASE structure) (level 2)

Trace point: 1271
Subcomponent: Data Control
Module: ARIYC11
Trace level: 2
Type: Module report
Description: Displays input parameters: fields in CBASE structure

Trace point: 1272
Subcomponent: Data Control
Module: ARIYC11
Trace level: 2
Type: Module report
Description: Displays input parameters: DBSPACE description auxiliary structure (SCR) pointed to by AUXPTR in CBASE structure (YP->CBASE->AUXPTR->SCR)

Trace point: 1273
Subcomponent: Data Control
Module: ARIYC11
Trace level: 2
Type: Module report
Description: Displays input parameters: description of type 1 table auxiliary structure (MCR1) pointed to by AUXPTR in CBASE structure (YP->CBASE->AUXPTR->MCR1)

Trace point: 1274
Subcomponent: Data Control
Module: ARIYC11
Trace level: 2
Type: Module report
Description: Displays input parameters: description of 1 link (child-LCR) auxiliary structure (LCR) pointed to by AUXPTR in CBASE structure (YP->CBASE->AUXPTR->LCR)

Trace point: 1275
Subcomponent: Data Control
Module: ARIYC11
Trace level: 2
Type: Module report
Description: Displays input parameters: description of 1 link (parent-PLCR) auxiliary structure (PLCR) pointed to by AUXPTR in CBASE structure (YP->CBASE->AUXPTR->PLCR)

Trace point: 1276
Subcomponent: Data Control
Module: ARIYC11
Trace level: 2
Type: Module report
Description: Displays input parameters: description of 1 index auxiliary structure (ICR) pointed to by AUXPTR in CBASE structure (YP->CBASE->AUXPTR->ICR)

Trace point: 1284
Subcomponent: Data Control
Module: ARIYC11
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 1285
Subcomponent: Data Control
Module: ARIYC13
Trace level: 1
Type: Module entry
Description: Displays input parameters: DCRIG, DCSEG, DLIDL, and LBPTR (all level 2)

Trace point: 1289
Subcomponent: Data Control
Module: ARIYC13
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 1290
Subcomponent: Data Control
Module: ARIYC14
Trace level: 1
Type: Module entry
Description: Displays input parameters: R3MR, DIIDI, and LBPTR (all level 2)

Trace point: 1294
Subcomponent: Data Control
Module: ARIYC14
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 1295
Subcomponent: Data Control
Module: ARIYC15
Trace level: 1
Type: Module entry
Description: Displays input parameters: RR3, RR2, PID, and TLEN (all level 2)

Trace point: 1297
Subcomponent: Data Control
Module: ARIYC15
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 1298
Subcomponent: Data Control
Module: ARIYC15 (entry point ARIYC151)
Trace level: 1
Type: Module entry
Description: Displays input parameter INCODE (level 2)

Trace point: 1299
Subcomponent: Data Control
Module: ARIYC15 (entry point ARIYC151)
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 1300
Subcomponent: Data Control
Module: ARIYC18
Trace level: 1
Type: Module entry
Description: Displays input values: TRANMODE and LOGL from YTABLE1 (both level 2)

Trace point: 1301
Subcomponent: Data Control
Module: ARIYC18
Trace level: 2
Type: Module report
Description: Displays data read from LOG (DBSPACE control row header)

Trace point: 1302
Subcomponent: Data Control
Module: ARIYC18
Trace level: 2
Type: Module report
Description: Displays data read from LOG (DBSPACE control row descriptors)

Trace point: 1304
Subcomponent: Data Control
Module: ARIYC18
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 1305
Subcomponent: Data Control
Module: ARIYC19
Trace level: 1
Type: Module entry
Description: Displays input values: TRANMODE and LOGL from YTABLE1 (both level 2)

Trace point: 1306
Subcomponent: Data Control
Module: ARIYC19
Trace level: 2
Type: Module report
Description: Displays data read from LOG (type 1 table row header)

Trace point: 1307
Subcomponent: Data Control
Module: ARIYC19
Trace level: 2
Type: Module report
Description: Displays data read from LOG (type 1 table row descriptors)

Trace point: 1309
Subcomponent: Data Control
Module: ARIYC19
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 1310
Subcomponent: Data Control
Module: ARIYC20
Trace level: 1
Type: Module entry
Description: Displays input values: TRANMODE and LOGL from YTABLE1 (level 2)

Trace point: 1311
Subcomponent: Data Control
Module: ARIYC20
Trace level: 2
Type: Module report
Description: Displays data read from LOG (ICR row header)

Trace point: 1319
Subcomponent: Data Control
Module: ARIYC20
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 1320
Subcomponent: Data Control
Module: ARIYC21
Trace level: 1
Type: Module entry
Description: Displays input values: TRANMODE and LOGL from YTABLE1 (both level 2)

Trace point: 1321
Subcomponent: Data Control
Module: ARIYC21
Trace level: 2
Type: Module report
Description: Displays data read from LOG (LINK row header)

Trace point: 1329
Subcomponent: Data Control
Module: ARIYC21
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 1330
Subcomponent: Data Control
Module: ARIYC22
Trace level: 1
Type: Module entry
Description: Displays input values: TRANMODE and LOGL from YTABLE1 (both level 2)

Trace point: 1331
Subcomponent: Data Control
Module: ARIYC22
Trace level: 2
Type: Module report
Description: Displays data read from LOG (update control row)

Trace point: 1335
Subcomponent: Data Control
Module: ARIYC22
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 1336
Subcomponent: Data Control
Module: ARIYC23
Trace level: 1
Type: Module entry
Description: Displays input values: TRANMODE and LOGL from YTABLE1 (both level 2)

Trace point: 1337
Subcomponent: Data Control
Module: ARIYC23
Trace level: 2
Type: Module report
Description: Displays data read from LOG (delete control row)

Trace point: 1339
Subcomponent: Data Control
Module: ARIYC23
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 1340
Subcomponent: Data Control
Module: ARIYC24
Trace level: 1
Type: Module entry
Description: Displays input values: TRANMODE and LOGL from YTABLE1 (both level 2)

Trace point: 1341
Subcomponent: Data Control
Module: ARIYC24
Trace level: 2
Type: Module report
Description: Displays data read from LOG (delete MCR1 control row)

Trace point: 1349
Subcomponent: Data Control
Module: ARIYC24
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 1350
Subcomponent: Data Control
Module: ARIYC25
Trace level: 1
Type: Module entry
Description: Displays input values: TRANMODE and LOGL from YTABLE1 (both level 2)

Trace point: 1351
Subcomponent: Data Control
Module: ARIYC25
Trace level: 2
Type: Module report
Description: Displays data read from LOG (delete ICR control row)

Trace point: 1359
Subcomponent: Data Control
Module: ARIYC25
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 1360
Subcomponent: Data Control
Module: ARIYC26
Trace level: 1
Type: Module entry
Description: Displays input values: TRANMODE and LOGL from YTABLE1 (both level 2)

Trace point: 1361
Subcomponent: Data Control
Module: ARIYC26
Trace level: 2
Type: Module report
Description: Displays data read from LOG (delete LINK control row)

Trace point: 1369
Subcomponent: Data Control
Module: ARIYC26
Trace level: 1
Type: Module exit
Description: Displays return code

ARIYD80 (DBSS Data Manipulation Trace Point Descriptor)

Trace point: 1500
Subcomponent: Data Manipulation
Module: ARIYD00
Trace level: 1
Type: Module entry
Description: Traces entry to module ARIYD00 (DBSI call for Close Scan). No variables are displayed since DBSS entry tracing displays all input.

Trace point: 1501
Subcomponent: Data Manipulation
Module: ARIYD00
Trace level: 1
Type: Module exit
Description: Traces exit from module ARIYD00 (DBSI call for Close Scan). Other than return code, no variables are displayed since DBSS exit tracing displays all output (to DBSI caller).

Trace point: 1504
Subcomponent: Data Manipulation
Module: ARIYD01
Trace level: 1
Type: Module entry
Description: Traces entry to module ARIYD01 (DBSI call for Connect Row). No variables are displayed since DBSS entry tracing displays all input.

Trace point: 1505
Subcomponent: Data Manipulation
Module: ARIYD01
Trace level: 1
Type: Module exit
Description: Traces exit from module ARIYD01 (DBSI call for Connect Row). Other than return code, no variables are displayed since DBSS exit tracing displays all output (to DBSI caller).

Trace point: 1508
Subcomponent: Data Manipulation
Module: ARIYD02
Trace level: 1
Type: Module entry
Description: Traces entry to module ARIYD02 (DBSI call for Delete Row). No variables are displayed since DBSS entry tracing displays all input.

Trace point: 1509
Subcomponent: Data Manipulation
Module: ARIYD02
Trace level: 1
Type: Module exit
Description: Traces exit from module ARIYD02 (DBSI call for Delete Row). Other than return code, no variables are displayed since DBSS exit tracing displays all output (to DBSI caller).

Trace point: 1512
Subcomponent: Data Manipulation
Module: ARIYD03
Trace level: 1
Type: Module entry
Description: Traces entry to module ARIYD03 (DBSI call for Disconnect Row). No variables are displayed since DBSS entry tracing displays all input.

Trace point: 1513
Subcomponent: Data Manipulation
Module: ARIYD03
Trace level: 1
Type: Module exit
Description: Traces exit from module ARIYD02 (DBSI call for Disconnect Row). Other than return code, no variables are displayed since DBSS exit tracing displays all output (to DBSI caller).

Trace point: 1516
Subcomponent: Data Manipulation
Module: ARIYD04
Trace level: 1
Type: Module entry
Description: Traces entry to module ARIYD04 (DBSI call for Fetch Row). No variables are displayed since DBSS entry tracing displays all output (to DBSI caller).

Trace point: 1517
Subcomponent: Data Manipulation
Module: ARIYD04
Trace level: 1
Type: Module exit
Description: Traces exit from module ARIYD04 (DBSI call for Fetch Row). Other than return code, no variables are displayed since DBSS exit tracing displays all output (to DBSI caller).

Trace point: 1520
Subcomponent: Data Manipulation
Module: ARIYD05
Trace level: 1
Type: Module entry
Description: Traces entry to module ARIYD05 (DBSI call for Insert Row). No variables are displayed since DBSS entry tracing displays all input.

Trace point: 1521
Subcomponent: Data Manipulation
Module: ARIYD05
Trace level: 1
Type: Module exit
Description: Traces exit from module ARIYD05 (DBSI call for Insert Row). Other than return code, no variables are displayed since DBSS exit tracing displays all output (to DBSI caller).

Trace point: 1524
Subcomponent: Data Manipulation
Module: ARIYD06
Trace level: 1
Type: Module entry
Description: Traces entry to module ARIYD06 (DBSI call for Get Next Row). No variables are displayed since DBSS entry tracing displays all input.

Trace point: 1525
Subcomponent: Data Manipulation
Module: ARIYD06
Trace level: 1
Type: Module exit
Description: Traces exit from module ARIYD06 (DBSI call for Get Next Row). Other than return code, no variables are displayed since DBSS exit tracing displays all output (to DBSI caller).

Trace point: 1528
Subcomponent: Data Manipulation
Module: ARIYD08
Trace level: 1
Type: Module entry
Description: Traces entry to module ARIYD08 (DBSI call for Open Scan). No variables are displayed since DBSS entry tracing displays all input.

Trace point: 1529
Subcomponent: Data Manipulation
Module: ARIYD08
Trace level: 1
Type: Module exit
Description: Traces exit from module ARIYD08 (DBSI call for Open Scan). Other than return code, no variables are displayed since DBSS exit tracing displays all output (to DBSI caller).

Trace point: 1532
Subcomponent: Data Manipulation
Module: ARIYD09
Trace level: 1
Type: Module entry
Description: Traces entry to module ARIYD09 (DBSI call for Get Parent). No variables are displayed since DBSS entry tracing displays all input.

Trace point: 1533
Subcomponent: Data Manipulation
Module: ARIYD09
Trace level: 1
Type: Module exit
Description: Traces exit from module ARIYD09 (DBSI call for Get Parent). Other than return code, no variables are displayed since DBSS exit tracing displays all output (to DBSI caller).

Trace point: 1536
Subcomponent: Data Manipulation
Module: ARIYD10
Trace level: 1
Type: Module entry
Description: Traces entry to module ARIYD10 (DBSI call for Update Row). No variables are displayed since DBSS entry tracing displays all input.

Trace point: 1537
Subcomponent: Data Manipulation
Module: ARIYD10
Trace level: 1
Type: Module exit
Description: Traces exit from module ARIYD10 (DBSI call for Update Row). Other than return code, no variables are displayed since DBSS exit tracing displays all output (to DBSI caller).

Trace point: 1540
Subcomponent: Data Manipulation
Module: ARIYD11
Trace level: 1
Type: Module entry
(primary entry point ARIYD11)
Description: Traces entry to module ARIYD11 primary entry point. Displays YTABLE1 variable BACKUPCT and TPMAP flag word STATE (both level 2).

Trace point: 1541
Subcomponent: Data Manipulation
Module: ARIYD11
Trace level: 1
Type: Module entry
(secondary entry point ARIYD111)
Description: Traces entry to module ARIYD11 secondary entry point ARIYD111. Displays YTABLE1 variable BACKUPCT and TPMAP flag word STATE (both level 2).

Trace point: 1542
Subcomponent: Data Manipulation
Module: ARIYD11
Trace level: 1
Type: Module exit
Description: Traces exit from module ARIYD11. Displays return code (level 1) and output parameters RECODE and RFBACK(1) (both level 2).

Trace point: 1545
Subcomponent: Data Manipulation
Module: ARIYD14
Trace level: 1
Type: Module entry
Description: Traces entry to module ARIYD14. Displays input parameters PAGEPT, VHEADER (page header), and TIDFLAG (all level 2).

Trace point: 1546
Subcomponent: Data Manipulation
Module: ARIYD14
Trace level: 1
Type: Module exit
Description: Traces exit from module ARIYD14. Displays VHEADER (page header) (level 2). ARIYD14 does not set a return code (so none displayed).

Trace point: 1549
Subcomponent: Data Manipulation
Module: ARIYD15
Trace level: 1
Type: Module entry
Description: Trace entry to module ARIYD15. Displays input parameters LOGB and LCONFLG (both level 2).

Trace point: 1550
Subcomponent: Data Manipulation
Module: ARIYD15
Trace level: 1
Type: Module exit
Description: Traces exit from module ARIYD15. Displays return code (level 1). Displays input/output parameters LOGB and LCONFLG and output parameters T2PAG and T2TUP (all level 2).

Trace point: 1555
Subcomponent: Data Manipulation
Module: ARIYD18
Trace level: 1
Type: Module entry
Description: Traces entry to module ARIYD18. Displays input parameters LOGB and LCONFLG (both level 2).

Trace point: 1556
Subcomponent: Data Manipulation
Module: ARIYD18
Trace level: 1
Type: Module exit
Description: Traces exit from module ARIYD18. Displays return code (level 1). Displays input/output parameters LOGB and LCONFLG (both level 2).

Trace point: 1559
Subcomponent: Data Manipulation
Module: ARIYD19
Trace level: 1
Type: Module entry
Description: Traces entry to module ARIYD19. Displays input parameters PAGENO, PAGEPTR, BLOCKPTR (all level 2).

Trace point: 1560
Subcomponent: Data Manipulation
Module: ARIYD19
Trace level: 1
Type: Module exit
Description: Traces exit from module ARIYD19. Displays return code (level 1). Displays input/output parameter BLOCKPTR (level 2).

Trace point: 1563
Subcomponent: Data Manipulation
Module: ARIYD20
(primary entry point)
Trace level: 1
Type: Module entry
Description: Traces entry to module ARIYD20 (primary entry point). Displays input parameter INTID and YTABLE1 input EPPTR(1) (both level 2).

Trace point: 1565
Subcomponent: Data Manipulation
Module: ARIYD20
Trace level: 1
Type: Module exit
Description: Traces exit from module ARIYD20. Displays return code (level 1). Displays output parameter OUTTID (level 2).

Trace point: 1568
Subcomponent: Data Manipulation
Module: ARIYD23
Trace level: 1
Type: Module entry
(primary entry point ARIYD23)
Description: Traces primary entry point for module ARIYD23. Displays input parameter INTIDH (level 2).

Trace point: 1569
Subcomponent: Data Manipulation
Module: ARIYD23
Trace level: 1
Type: Module entry
(secondary entry point ARIYD231)
Description: Traces secondary entry point ARIYD231. Displays input parameters INTIDF, RWCODE (both level 2).

Trace point: 1570
Subcomponent: Data Manipulation
Module: ARIYD23
Trace level: 1
Type: Module entry
(secondary entry point ARIYD232)
Description: Traces secondary entry point ARIYD232. Displays input parameters INTIDF, RWCODE (both level 2).

Trace point: 1571
Subcomponent: Data Manipulation
Module: ARIYD23
Trace level: 1
Type: Module exit
Description: Traces exit from module ARIYD23 Displays return code (level 1). Displays output Registers 2 and 3 (both level 2).

Trace point: 1574
Subcomponent: Data Manipulation
Module: ARIYD26
Trace level: 1
Type: Module entry
Description: Traces entry to module ARIYD23. Displays input parameters INTID, RWCODE, TIDTYPE (all level 2).

Trace point: 1575
Subcomponent: Data Manipulation
Module: ARIYD26
Trace level: 1
Type: Module exit
Description: Traces exit from module ARIYD26. Displays return code (level 1). Displays output Registers 2 and 3 (both level 2).

Trace point: 1578
Subcomponent: Data Manipulation
Module: ARIYD27
Trace level: 1
Type: Module entry
Description: Traces entry to module ARIYD27. Displays input structures IMAGEREC and DOMREC (both level 2). Note: KDOMAINS input is displayed by DBSS entry tracing.

Trace point: 1579
Subcomponent: Data Manipulation
Module: ARIYD27
Trace level: 1
Type: Module exit
Description: Traces exit from module ARIYD27. Displays return code (level 1). Displays YTABLE1 output values IKEYGEN, IKEYLTH, IKEYVAL (all level 2).

Trace point: 1582
Subcomponent: Data Manipulation
Module: ARIYD28
Trace level: 1
Type: Module entry
Description: Traces entry to module ARIYD28. Displays input parameter TUPLEPTR, input structure VTUPLE (input stored row), input structures IMAGEREC and DOMREC (all level 2).

Trace point: 1583
Subcomponent: Data Manipulation
Module: ARIYD28
Trace level: 1
Type: Module exit
Description: Traces exit from module ARIYD28. Displays return code (level 1). Displays YTABLE1 output values IKEYLTH, IKEYVAL (both level 2).

Trace point: 1586
Subcomponent: Data Manipulation
Module: ARIYD29
Trace level: 1
Type: Module entry
Description: Traces entry to module ARIYD29. Displays input structures IMAGEREC and DOMREC (both level 2). Note: DOMAINS input is displayed by DBSS entry tracing.

Trace point: 1587
Subcomponent: Data Manipulation
Module: ARIYD29
Trace level: 1
Type: Module exit
Description: Traces exit from module ARIYD29. Displays return code (level 1). Displays YTABLE1 output values IKEYLTH, IKEYVAL (both level 2).

Trace point: 1590
Subcomponent: Data Manipulation
Module: ARIYD30
Trace level: 1
Type: Module entry
Description: Traces entry to module ARIYD30. Displays input parameter TUPLEPTR, input structure VTUPLE (input stored row), input structures IMAGEREC and DOMREC (all level 2). Note: DOMAINS input is displayed by DBSS entry tracing.

Trace point: 1591
Subcomponent: Data Manipulation
Module: ARIYD30
Trace level: 1
Type: Module exit
Description: Traces exit from module ARIYD30. Displays return code (level 1). Displays YTABLE1 output values IKEYLTH, IKEYVAL (both level 2).

Trace point: 1594
Subcomponent: Data Manipulation
Module: ARIYD32
Trace level: 1
Type: Module entry
Description: Traces entry to module ARIYD32. Displays input parameter INRID, and global (YTABLE1) inputs LASTRELN, GLOCKREQ, SEGSTATE, and RELSTATE (all level 2).

Trace point: 1595
Subcomponent: Data Manipulation
Module: ARIYD32
Trace level: 1
Type: Module exit
Description: Traces exit from module ARIYD32. Displays return code (level 1). Displays YTABLE1 output fields RELSTATE and YMASTREC (both level 2).

Trace point: 1598
Subcomponent: Data Manipulation
Module: ARIYD33
Trace level: 1
Type: Module entry
Description: Traces entry to module ARIYD33. Displays input parameter LOGHEAD, VHEADER (page header of input page), VTUPLE (input row), DOMREC, and YTABLE1 fields: ELTH, ETIDS, EPPTR, SEGSTATE, RELSTATE, YSFLAGS, YLPREFIX, YDEGREE, and SEGMATTR (all level 2).

Trace point: 1599
Subcomponent: Data Manipulation
Module: ARIYD33
Trace level: 1
Type: Module exit
Description: Traces exit from module ARIYD33. Displays return code (level 1). Displays VHEADER (page header of output page), VTUPLE (output row), and YTABLE1 fields: ETIDS, EPPTR (all level 2).

Trace point: 1602
Subcomponent: Data Manipulation
Module: ARIYD34
Trace level: 1
Type: Module entry
Description: Traces entry to module ARIYD34. Displays input parameter LTH, SCANS (scan table header) (both level 2).

Trace point: 1603
Subcomponent: Data Manipulation
Module: ARIYD34
Trace level: 1
Type: Module exit
Description: Traces exit from module ARIYD34. Displays return code (level 1). Displays output parameter SCBPT, SCANS (scan table header), and allocated SCB (first 3 bytes) (all level 2).

Trace point: 1606
Subcomponent: Data Manipulation
Module: ARIYD38
Trace level: 1
Type: Module entry
Description: Traces entry to module ARIYD38. Displays input parameter INSEGM, and YTABLE1 fields: LASTSEGN, LASTSEST, GLOCKREQ, SYSMODE (all level 2).

Trace point: 1607
Subcomponent: Data Manipulation
Module: ARIYD38
Trace level: 1
Type: Module exit
Description: Traces exit from module ARIYD38. Displays return code (level 1). Displays YTABLE1 fields: SEGMATTR, SEGSTATE, YSHEADER (all level 2).

Trace point: 1610
Subcomponent: Data Manipulation
Module: ARIYD39
Trace level: 1
Type: Module entry
Description: Traces entry to module ARIYD39. Displays input parameters TPAGE and TIDTYPE, and YTABLE1 fields: YSFREE, YSFLAGS, SEGSTATE, RELSTATE (all level 2).

Trace point: 1611
Subcomponent: Data Manipulation
Module: ARIYD39
Trace level: 1
Type: Module exit
Description: Traces exit from module ARIYD39. Displays return code (level 1). Displays output parameters PAGEPTR, BLOCKPTR, RETID, updated page header (VHEADER), and YTABLE1 fields: PAGECODE, TIDCODE (all level 2).

Trace point: 1614
Subcomponent: Data Manipulation
Module: ARIYD40
Trace level: 1
Type: Module entry
Description: Traces entry to module ARIYD40. Displays input parameters INTID, RWCODE, TIDTYPE (all level 2).

Trace point: 1615
Subcomponent: Data Manipulation
Module: ARIYD40
Trace level: 1
Type: Module exit
Description: Traces exit from module ARIYD40. Displays return code (level 1). Displays output Register 3 and the header of the output page (VHEADER) (both level 2).

Trace point: 1618
Subcomponent: Data Manipulation
Module: ARIYD41
Trace level: 1
Type: Module entry
(primary entry point ARIYD41)
Description: Traces primary entry to module ARIYD41. Displays input parameter INTID and YTABLE1 fields: SEGSTATE, RELSTATE, YSFLAGS, CLEVEL (all level 2)

Trace point: 1619
Subcomponent: Data Manipulation
Module: ARIYD41
Trace level: 1
Type: Module entry
(secondary entry point ARIYD411)
Description: Traces secondary entry to module ARIYD41 (entry point ARIYD411). Displays input parameter INTID and YTABLE1 fields: SEGSTATE, RELSTATE, YSFLAGS, CLEVEL (all level 2).

Trace point: 1620
Subcomponent: Data Manipulation
Module: ARIYD41
Trace level: 1
Type: Module exit
Description: Traces exit from module ARIYD41. Displays return code (level 1).

Trace point: 1622
Subcomponent: Data Manipulation
Module: ARIYD44
Trace level: 1
Type: Module entry
Description: Traces entry to module ARIYD44. Displays input parameters OPCODE and OBJECTID, global SCFREE (in scan table header), and YTABLE1 value SEGN (all level 2).

Trace point: 1623
Subcomponent: Data Manipulation
Module: ARIYD44
Trace level: 1
Type: Module exit
Description: Traces exit from module ARIYD44. Displays global variable SCFREE (in scan table header) (level 2).

Trace point: 1626
Subcomponent: Data Manipulation
Module: ARIYD45
Trace level: 1
Type: Module entry
Description: Traces entry to module ARIYD45. Displays input global objects DOMREC, IMAGEREC, and YTABLE1 fields: IKEYOLTH, IKEYOVAL (all level 2).

Trace point: 1627
Subcomponent: Data Manipulation
Module: ARIYD45
Trace level: 1
Type: Module exit
Description: Traces exit from module ARIYD45. Displays return code (level 1).

Trace point: 1630
Subcomponent: Data Manipulation
Module: ARIYD46
Trace level: 1
Type: Module entry
Description: Traces entry to module ARIYD46. Displays input parameter SAVECODE and YTABLE1 values: IID, LASTIID, SEGSTATE, RELSTATE, GLOCKREQ (all level 2).

Trace point: 1631
Subcomponent: Data Manipulation
Module: ARIYD46
Trace level: 1
Type: Module exit
Description: Traces exit from module ARIYD46. Displays return code (level 1). Displays YTABLE1 fields: TID, LASTIID, ILOCK (all level 2).

Trace point: 1634
Subcomponent: Data Manipulation
Module: ARIYD47
Trace level: 1
Type: Module entry
Description: Traces entry to module ARIYD47. Displays input parameters INTID and INSTATE, and YTABLE1 values: PAGECODE, TIDCODE (all level 2).

Trace point: 1635
Subcomponent: Data Manipulation
Module: ARIYD47
Trace level: 1
Type: Module exit
Description: Traces exit from module ARIYD47. Displays return code (level 1).

Trace point: 1638
Subcomponent: Data Manipulation
Module: ARIYD48
Trace level: 1
Type: Module entry
Description: Traces entry to module ARIYD48. Displays input parameters: OBTYPE, CMRCODE, CTIDP, PMRCODE, PTIDP, and YTABLE1 values: YSHEADER and YMASTREC (all level 2).

Trace point: 1639
Subcomponent: Data Manipulation
Module: ARIYD48
Trace level: 1
Type: Module exit
Description: Traces exit from module ARIYD48. Displays return code (level 1).

Trace point: 1642
Subcomponent: Data Manipulation
Module: ARIYD49
Trace level: 1
Type: Module entry
Description: Traces entry to module ARIYD49. Displays input value VTUPLE (pointed to be Register 3) and global input structure DOMREC (both level 2).

Trace point: 1643
Subcomponent: Data Manipulation
Module: ARIYD49
Trace level: 1
Type: Module exit
Description: Traces exit from module ARIYD49. Displays return code (level 1).

Trace point: 1646
Subcomponent: Data Manipulation
Module: ARIYD50
Trace level: 1
Type: Module entry
(primary entry point ARIYD50)
Description: Traces primary entry to module ARIYD50 (entry point ARIYD50). Displays input parameter INTID and YTABLE1 substructure YMASTREC (both level 2).

Trace point: 1648
Subcomponent: Data Manipulation
Module: ARIYD50
Trace level: 1
Type: Module exit
Description: Traces exit from module ARIYD50. Displays return code (level 2). Displays output parameter OUTTID and output row VTUPLE (both level 2).

Trace point: 1660
Subcomponent: Data Manipulation
Module: ARIYD57
Trace level: 1
Type: Module entry
Description: Traces entry to module ARIYD57. Displays global input structure DOMREC (level 2).

Trace point: 1661
Subcomponent: Data Manipulation
Module: ARIYD57
Trace level: 1
Type: Module exit
Description: Traces exit from module ARIYD57. Displays return code (level 1).

Trace point: 1664
Subcomponent: Data Manipulation
Module: ARIYD58
Trace level: 1
Type: Module entry
Description: Traces entry to module ARIYD58. Displays input value VTUPLE and global input structure DOMREC (both level 2).

Trace point: 1665
Subcomponent: Data Manipulation
Module: ARIYD58
Trace level: 1
Type: Module exit
Description: Traces exit from module ARIYD58. Displays return code (level 1).

Trace point: 1668
Subcomponent: Data Manipulation
Module: ARIYD59
Trace level: 1
Type: Module entry
Description: Traces entry to module ARIYD59. Displays YTABLE1 field SCANID (level 2).

Trace point: 1669
Subcomponent: Data Manipulation
Module: ARIYD59
Trace level: 1
Type: Module exit
Description: Traces exit from module ARIYD59. Displays return code (level 1). Displays SCBHEAD (processed SCB) and YTABLE1 fields: SEGMENT, RID, TID, and SCBPTR (all level 2).

Trace point: 1672
Subcomponent: Data Manipulation
Module: ARIYD60
Trace level: 1
Type: Module entry
Description: Traces entry to module ARIYD60. Displays YTABLE1 field SCANID (level 2).

Trace point: 1673
Subcomponent: Data Manipulation
Module: ARIYD60
Trace level: 1
Type: Module exit
Description: Traces exit from module ARIYD60. Displays return code (level 1). Displays SCBHEAD (processed SCB) and YTABLE1 fields: PSEGMENT, PRID, PTID, and SCBPTR (all level 2).

Trace point: 1676
Subcomponent: Data Manipulation
Module: ARIYD61
Trace level: 1
Type: Module entry
Description: Traces entry to module ARIYD61. Displays scan table header (SCANS) (level 2).

Trace point: 1677
Subcomponent: Data Manipulation
Module: ARIYD61
Trace level: 1
Type: Module exit
Description: Traces exit from module ARIYD61. Displays scan table header (SCANS) (level 2).

Trace point: 1680
Subcomponent: Data Manipulation
Module: ARIYD62
Trace level: 1
Type: Module entry
Description: Traces entry to module ARIYD62. Displays input parameters: INTID, LOCKMODE, TIDTYPE, and YTABLE1 values: SEGSTATE, RELSTATE, YSFLAGS (all level 2).

Trace point: 1681
Subcomponent: Data Manipulation
Module: ARIYD62
Trace level: 1
Type: Module exit
Description: Traces exit from module ARIYD62. Displays return code (level 1). Displays output Register 2 and output row VTUPLE and YTABLE1 fields: PAGECODE, TIDCODE, and ETIDS (all level 2).

Trace point: 1684
Subcomponent: Data Manipulation
Module: ARIYD63
Trace level: 1
Type: Module entry
Description: Traces entry to module ARIYD63. Displays input parameters: BLOCKPTR, VHEADER (page header) (both level 2).

Trace point: 1685
Subcomponent: Data Manipulation
Module: ARIYD63
Trace level: 1
Type: Module exit
Description: Traces exit from module ARIYD63. Displays return code (level 1). Displays output parameter BLOCKPTR (level 2).

Trace point: 1688
Subcomponent: Data Manipulation
Module: ARIYD64
Trace level: 1
Type: Module entry
Description: Traces entry to module ARIYD64. Displays input parameters: INTID, MODCODE, TIDTYPE, and YTABLE1 values: SEGSTATE, RELSTATE, YSFLAGS (all level 2).

Trace point: 1689
Subcomponent: Data Manipulation
Module: ARIYD64
Trace level: 1
Type: Module exit
Description: Traces exit from module ARIYD64. Displays return code (level 1). Displays output page header (VHEADER) and output row (VTUPLE) and YTABLE1 fields: PAGECODE, TIDCODE, and ETIDS (all level 2).

Trace point: 1692
Subcomponent: Data Manipulation
Module: ARIIYD65
Trace level: I
Type: Module entry
Description: Traces entry to module ARIYD65. Displays YTABLE1 value TRANMODE (level 2).

Trace point: 1693
Subcomponent: Data Manipulation
Module: ARIYD65
Trace level: 2
Type: Module report (call input)
Description: Traces log input to module ARIYD65. Displays log input record (LCONNECT).

Trace point: 1694
Subcomponent: Data Manipulation
Module: ARIYD65
Trace level: 1
Type: Module exit
Description: Traces exit from module ARIYD65. Displays return code (level 1). Displays local log data LCONNECT and YTABLE1 substructure YTABLE1U (remap of YBASE for recovery) (both level 2).

Trace point: 1697
Subcomponent: Data Manipulation
Module: ARIYD66
Trace level: 1
Type: Module entry
Description: Traces entry to module ARIYD66. Displays YTABLE1 value TRANMODE (level 2).

Trace point: 1698
Subcomponent: Data Manipulation
Module: ARIYD66
Trace level: 2
Type: Module report (call input)
Description: Traces log input to module ARIYD66. Displays log input record (LDELETE).

Trace point: 1699
Subcomponent: Data Manipulation
Module: ARIYD66
Trace level: 1
Type: Module exit
Description: Traces exit from module ARIYD66. Displays return code (level 1). Displays YTABLE1 substructure YTABLE1U (remap of YBASE for recovery) (level 2).

Trace point: 1702
Subcomponent: Data Manipulation
Module: ARIYD67
Trace level: 1
Type: Module entry
Description: Traces entry to module ARIYD67. Displays YTABLE1 value TRANMODE (level 2).

Trace point: 1703
Subcomponent: Data Manipulation
Module: ARIYD67
Trace level: 2
Type: Module report (call input)
Description: Traces log input to module ARIYD66. Displays log input record (LDISCONN).

Trace point: 1704
Subcomponent: Data Manipulation
Module: ARIYD67
Trace level: 1
Type: Module exit
Description: Traces exit from module ARIYD67. Displays return code (level 1). Displays YTABLE1 substructure YTABLE1U (remap of YBASE for recovery) (level 2).

Trace point: 1707
Subcomponent: Data Manipulation
Module: ARIYD68
Trace level: 1
Type: Module entry
Description: Traces entry to module ARIYD68. Displays YTABLE1 value TRANMODE (level 2).

Trace point: 1708
Subcomponent: Data Manipulation
Module: ARIYD68
Trace level: 2
Type: Module report (call input)
Description: Traces log input to module ARIYD68. Displays log input record (LINSERT).

Trace point: 1709
Subcomponent: Data Manipulation
Module: ARIYD68
Trace level: 1
Type: Module exit
Description: Traces exit from module ARIYD68. Displays return code (level 1). Displays YTABLE1 substructure YTABLE1U (remap of YBASE for recovery) (level 2).

Trace point: 1712
Subcomponent: Data Manipulation
Module: ARIYD69
Trace level: 1
Type: Module entry
Description: Traces entry to module ARIYD69. Displays YTABLE1 value TRANMODE (level 2).

Trace point: 1713
Subcomponent: Data Manipulation
Module: ARIYD69
Trace level: 2
Type: Module report (call input)
Description: Traces log input to module ARIYD69. Displays log input record (LUPDATE).

Trace point: 1714
Subcomponent: Data Manipulation
Module: ARIYD69
Trace level: 1
Type: Module exit
Description: Traces exit from module ARIYD69. Displays return code (level 1). Displays YTABLE1 substructure YTABLE1U (remap of YBASE for recovery) (level 2).

Trace point: 1717
Subcomponent: Data Manipulation
Module: ARIYD70
Trace level: 1
Type: Module entry
Description: Traces entry to module ARIYD70. Displays input page header (VHEADER) and input parameters LTH and IDN and YTABLE1 values: EPAGELKU (YTABLE1U) and SEGN (all level 2).

Trace point: 1718
Subcomponent: Data Manipulation
Module: ARIYD70
Trace level: 1
Type: Module exit
Description: Traces exit from module ARIYD70. Displays output page header (VHEADER) and local variable TIDLOCI (both level 2).

Trace point: 1721
Subcomponent: Data Manipulation
Module: ARIYD71
Trace level: 1
Type: Module entry
Description: Traces entry to module ARIYD71. Displays input parameters: LCONHEAD and LCONPREF and YTABLE1 value TRANMODE (all level 2).

Trace point: 1722
Subcomponent: Data Manipulation
Module: ARIYD71
Trace level: 1
Type: Module exit
Description: Traces exit from module ARIYD71. Displays return code (level 1).

Trace point: 1725
Subcomponent: Data Manipulation
Module: ARIYD72
Trace level: 1
Type: Module entry
Description: Traces entry to module ARIYD72. Displays input parameters: LDISCOM, LOGB, and YTABLE1 value TRANMODE (all level 2).

Trace point: 1726
Subcomponent: Data Manipulation
Module: ARIYD72
Trace level: 1
Type: Module exit
Description: Traces exit from module ARIYD72. Displays return code (level 1).

Trace point: 1729
Subcomponent: Data Manipulation
Module: ARIYD73
Trace level: 1
Type: Module entry
Description: Traces entry to module ARIYD73. Displays input parameter LCONTUPL and YTABLE1 values: TRANMODE, SEGN (all level 2).

Trace point: 1730
Subcomponent: Data Manipulation
Module: ARIYD73
Trace level: 1
Type: Module exit
Description: Traces exit from module ARIYD73. Displays return code (level 1). Displays local variable HW (log input), output page header (VHEADER), and output row header (VTHEADER) (all level 2).

Trace point: 1733
Subcomponent: Data Manipulation
Module: ARIYD74
Trace level: 1
Type: Module entry
Description: Traces entry to module ARIYD74. Displays input parameter LCONHEAD and YTABLE1 value TRANMODE (both level 2).

Trace point: 1734
Subcomponent: Data Manipulation
Module: ARIYD74
Trace level: 2
Type: Module report (call input)
Description: Traces retrieved control row input to ARIYD74. Displays DCR (DDMREC) global structure and YTABLE1 structure updates YSHEADER (SCR) and YMASTREC (MCR).

Trace point: 1735
Subcomponent: Data Manipulation
Module: ARIYD74
Trace level: 1
Type: Module exit
Description: Traces exit from module ARIYD74. Displays return code (level 1). Displays YTABLE1 fields SEGN and YTABLE1U (YTABLE1U is an overlay of YBASE for recovery) (both level 2).

Trace point: 1738
Subcomponent: Data Manipulation
Module: ARIYD75
Trace level: 1
Type: Module entry
Description: Traces entry to module ARIYD75. Displays input parameters OPMODE, MODTID, and MODPTR (all level 2).

Trace point: 1739
Subcomponent: Data Manipulation
Module: ARIYD75
Trace level: 1
Type: Module exit
Description: Traces exit from module ARIYD75. Displays return code (level 1).

Trace point: 1744
Subcomponent: Data Manipulation
Module: ARIYD77
Trace level: 1
Type: Module entry
Description: Traces entry to module ARIYD77. Displays input parameters: INTID0 and INLTH0, input Register 2, and YTABLE1 fields: TRANMODE and YTABLE1U (YTABLE1U is an overlay of YBASE for recovery) (all level 2).

Trace point: 1745
Subcomponent: Data Manipulation
Module: ARIYD77
Trace level: 1
Type: Module exit
Description: Traces exit from module ARIYD77. Displays return code (level 1). Displays output page header (VHEADER), output row (VTUPLE), and YTABLE1 field YTABLE1U (YTABLE1U is an overlay of YBASE for recovery) (all level 2).

Trace point: 1749
Subcomponent: Data Manipulation
Module: ARIYD87
Trace level: 1
Type: Module entry
(primary entry point ARIYD87)
Description: Traces entry to module ARIYD87 (primary entry point ARIYD87). Displays input parameter LENG and global variable SCANS (scan table header) (both level 2).

Trace point: 1750
Subcomponent: Data Manipulation
Module: ARIYD87
Trace level: 1
Type: Module exit
(for primary entry point ARIYD87)
Description: Traces exit from module ARIYD87 (for primary entry point ARIYD87). Displays no variables.

Trace point: 1751
Subcomponent: Data Manipulation
Module: ARIYD87
Trace level: 1
Type: Module entry
(secondary entry point ARIYD871)
Description: Traces entry to module ARIYD87 (secondary entry point ARIYD871). Displays global variable SCANS (scan table header) (level 2).

Trace point: 1752
Subcomponent: Data Manipulation
Module: ARIYD87
Trace level: 1
Type: Module exit
(for secondary entry point ARIYD871)
Description: Traces exit from module ARIYD87 (for secondary entry point ARIYD871). Displays global variable SCANS (scan table header) (level 2).

Trace point: 1753
Subcomponent: Data Manipulation
Module: ARIYD87
Trace level: 1
Type: Module entry
(secondary entry point ARIYD872)
Description: Traces entry to module ARIYD87 (secondary entry point ARIYD872). Displays global variable SCANS (scan table header) (level 2).

Trace point: 1754
Subcomponent: Data Manipulation
Module: ARIYD87
Trace level: 1
Type: Module exit
(for secondary entry point ARIYD872)
Description: Traces exit from module ARIYD87 (for secondary entry point ARIYD872). Displays global variable SCANS (scan table header) (level 2).

ARIYI52 (DBSS Storage (I/O) Component Trace Point Descriptor)

Trace point: 2000
Subcomponent: Storage (I/O)
Module: ARIYI22
Trace level: 1
Type: Module entry
Description: Displays input variable DB (level 2)

Trace point: 2001
Subcomponent: Storage (I/O)
Module: ARIYI22
Trace level: 1
Type: Module exit
Description: Displays no variables (no return code)

Trace point: 2002
Subcomponent: Storage (I/O)
Module: ARISFM1
Trace level: 1
Type: Module entry
Description: Displays input variable TYPE (disk type) (level 2).

Trace point: 2003
Subcomponent: Storage (I/O)
Module: ARISFM1
Trace level: 1
Type: Module exit
Description: Displays no variables (no return code)

Trace point: 2004
Subcomponent: Storage (I/O)
Module: ARISFM2
Trace level: 1
Type: Module entry
Description: Displays input variables: I, TYPE, J, BN, and POOL (all level 2)

Trace point: 2005
Subcomponent: Storage (I/O)
Module: ARISFM2
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 2006
Subcomponent: Storage (I/O)
Module: ARIYI29
Trace level: 1
Type: Module entry
Description: Displays input variables: PNUM, WCODE, BPTR, SN, BUFC, AND R2 (all level 2).

Trace point: 2007
Subcomponent: Storage (I/O)
Module: ARIYI29
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 2008
Subcomponent: Storage (I/O)
Module: ARISFDB
Trace level: 1
Type: Module entry
Description: Displays input variable PMAXBLKS (level 2)

Trace point: 2009
Subcomponent: Storage (I/O)
Module: ARISFDB
Trace level: 1
Type: Module exit
Description: Displays no variables (no return code)

Trace point: 2010
Subcomponent: Storage (I/O)
Module: ARIYI07
Trace level: 1
Type: Module entry
Description: Displays input variables: REQUEST and RECNO (all level 2).

Trace point: 2011
Subcomponent: Storage (I/O)
Module: ARIYI07
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 2012
Subcomponent: Storage (I/O)
Module: ARIYI23
Trace level: 1
Type: Module entry
Description: Displays input variables: EXTENT and FUNCTION (all level 2)

Trace point: 2013
Subcomponent: Storage (I/O)
Module: ARIYI23
Trace level: 1
Type: Module exit
Description: Displays no variables (no return code)

Trace point: 2014
Subcomponent: Storage (I/O)
Module: ARISIIO
Trace level: 1
Type: Module entry
Description: Displays no variables

Trace point: 2015
Subcomponent: Storage (I/O)
Module: ARISIIO (entry point ARISIIO1)
Trace level: 1
Type: Module entry
Description: Displays input variables: VTCURXT, VTMXXNT, VTMXDBS, VTMXPGS, and VTMXPLS (all level 2)

Trace point: 2016
Subcomponent: Storage (I/O)
Module: ARISIIO
Trace level: 1
Type: Module exit (for entry points ARISIIO and ARISIIO1)
Description: Displays no variables (no return code)

Trace point: 2017
Subcomponent: Storage (I/O)
Module: ARIYI00
Trace level: 1
Type: Module entry
Description: Displays input variables: CASE and V (both level 2)

Trace point: 2018
Subcomponent: Storage (I/O)
Module: ARIYI00
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 2019
Subcomponent: Storage (I/O)
Module: ARIYI05
Trace level: 1
Type: Module entry
Description: Displays no variables

Trace point: 2020
Subcomponent: Storage (I/O)
Module: ARIYI05
Trace level: 1
Type: Module exit
Description: Displays no variables (no return code)

Trace point: 2021
Subcomponent: Storage (I/O)
Module: ARIYI06
Trace level: 1
Type: Module entry
Description: Displays input variables: SEG and BLOCKNO (both level 2)

Trace point: 2022
Subcomponent: Storage (I/O)
Module: ARIYI06
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 2023
Subcomponent: Storage (I/O)
Module: ARIYI10
Trace level: 1
Type: Module entry
Description: Displays input variable SEG (level 2)

Trace point: 2024
Subcomponent: Storage (I/O)
Module: ARIYI10
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 2025
Subcomponent: Storage (I/O)
Module: ARIYI13
Trace level: 1
Type: Module entry
Description: Displays no variables

Trace point: 2026
Subcomponent: Storage (I/O)
Module: ARIYI13
Trace level: 1
Type: Module exit
Description: Displays no variables (no return code)

Trace point: 2027
Subcomponent: Storage (I/O)
Module: ARIYI14
Trace level: 1
Type: Module entry (entry point ARIYI14)
Description: Displays input variables: BLOCKN, RWCODE, BPTR, and CSEGNO (all level 2)

Trace point: 2028
Subcomponent: Storage (I/O)
Module: ARIYI14
Trace level: 1
Type: Module entry (entry point ARIYI141)
Description: Displays input variables: BLOCKN, RWCODE, BPTR, and SN (all level 2)

Trace point: 2029
Subcomponent: Storage (I/O)
Module: ARIYI14
Trace level: 1
Type: Module entry (entry point ARIYI142)
Description: Displays input variables: BLOCKN, RWCODE, BPTR, and SN (all level 2)

Trace point: 2030
Subcomponent: Storage (I/O)
Module: ARIYI14
Trace level: 1
Type: Module exit (for entry points ARIYI14, ARIYI141, ARIYI142)
Description: Displays return code

Trace point: 2031
Subcomponent: Storage (I/O)
Module: ARIYI17
Trace level: 1
Type: Module entry (entry point ARIYI17)
Description: Displays no variables

Trace point: 2032
Subcomponent: Storage (I/O)
Module: ARIYI17
Trace level: 1
Type: Module entry (entry point ARIYI171)
Description: Displays no variables

Trace point: 2033
Subcomponent: Storage (I/O)
Module: ARIYI17
Trace level: 1
Type: Module exit (for entry points ARIYI17 and ARIYI171)
Description: Displays return code

Trace point: 2034
Subcomponent: Storage (I/O)
Module: ARIYI19
Trace level: 1
Type: Module entry
Description: Displays input variables: P3, RWCODE, and BPTR (all level 2)

Trace point: 2035
Subcomponent: Storage (I/O)
Module: ARIYI19
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 2036
Subcomponent: Storage (I/O)
Module: ARIYI20
Trace level: 1
Type: Module entry
Description: Displays input variable SEG (level 2)

Trace point: 2037
Subcomponent: Storage (I/O)
Module: ARIYI20
Trace level: 1
Type: Module exit
Description: Displays output variable PSLOT (level 2). No return code

Trace point: 2038
Subcomponent: Storage (I/O)
Module: ARIYI21
Trace level: 1
Type: Module entry
Description: Displays no variables

Trace point: 2039
Subcomponent: Storage (I/O)
Module: ARIYI21
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 2040
Subcomponent: Storage (I/O) (Traced under LOG)
Module: ARIYI24
Trace level: 1
Type: Module entry
Description: Displays no variables. This trace point is for the I/O component module ARIYI24.
Note: This module is a linkage module for the Log component to access information from the MASTER (an I/O control block) thus the Log component ID is used.

Trace point: 2041
Subcomponent: Storage (I/O)
Module: ARIYI24
Trace level: 1
Type: Module exit
Description: Displays output variables: LREDO and LUNDO (level 2). No return code

```
Trace point:  2042
Subcomponent: Storage (I/O)
Module:       ARIYI26
Trace level:  1
Type:         Module entry (entry point ARIYI26)
Description:  Displays no variables

Trace point:  2043
Subcomponent: Storage (I/O)
Module:       ARIYI26
Trace level:  1
Type:         Module entry (entry point ARIYI261)
Description:  Displays no variables

Trace point:  2044
Subcomponent: Storage (I/O)
Module:       ARIYI26
Trace level:  1
Type:         Module entry (entry point ARIYI262)
Description:  Displays input variable V (level 2)

Trace point:  2045
Subcomponent: Storage (I/O)
Module:       ARIYI26
Trace level:  1
Type:         Module exit (for entry points ARIYI26,
              ARIYI261, and ARIYI262)
Description:  Displays return code

Trace point:  2046
Subcomponent: Storage (I/O)
Module:       ARIYI31
Trace level:  1
Type:         Module entry (entry point ARIYI311)
Description:  Displays input variable RW (level 2)

Trace point:  2047
Subcomponent: Storage (I/O)
Module:       ARIYI31
Trace level:  1
Type:         Module entry (entry point ARIYI312)
Description:  Displays input variable RW (level 2)

Trace point:  2048
Subcomponent: Storage (I/O)
Module:       ARIYI31
Trace level:  1
Type:         Module entry (entry point ARIYI313)
Description:  Displays input variable RW (level 2)

Trace point:  2049
Subcomponent: Storage (I/O)
Module:       ARIYI31
Trace level:  1
Type:         Module entry (entry point ARIYI314)
Description:  Displays input variable RW (level 2)

Trace point:  2050
Subcomponent: Storage (I/O)
Module:       ARIYI31
Trace level:  1
Type:         Module entry (entry point ARIYI315)
Description:  Displays input variable RW (level 2)

Trace point:  2051
Subcomponent: Storage (I/O)
Module:       ARIYI31
Trace level:  1
Type:         Module exit (for entry points ARIYI31,
              ARIYI311, ARIYI312, ARIYI313, ARIYI314, and
              ARIYI315)
Description:  Displays no variables. No return code

Trace point:  2052
Subcomponent: Storage (I/O)
Module:       ARIYI41
Trace level:  1
Type:         Module entry
Description:  Displays input variables: INSEGM and K (both
              level 2)

Trace point:  2053
Subcomponent: Storage (I/O)
Module:       ARIYI41
Trace level:  1
Type:         Module exit
Description:  Displays return code

Trace point:  2054
Subcomponent: Storage (I/O)
Module:       ARIYI42
Trace level:  1
Type:         Module entry (entry point ARIYI42)
Description:  Displays input variable SN (level 2)

Trace point:  2055
Subcomponent: Storage (I/O)
Module:       ARIYI42
Trace level:  1
Type:         Module entry (entry point ARIYI421)
Description:  Displays input variable SN (level 2)
```

Trace point: 2056
Subcomponent: Storage (I/O)
Module: ARIYI42
Trace level: 1
Type: Module entry (entry point ARIYI422)
Description: Displays input variable SN (level 2)

Trace point: 2057
Subcomponent: Storage (I/O)
Module: ARIYI42
Trace level: 1
Type: Module entry (entry point ARIYI423)
Description: Displays input variable SN (level 2)

Trace point: 2058
Subcomponent: Storage (I/O)
Module: ARIYI42
Trace level: 1
Type: Module exit (for entry points ARIYI42, ARIYI421, ARIYI422, and ARIYI423)
Description: Displays return code (level 1) and output variable B (level 2)

Trace point: 2059
Subcomponent: Storage (I/O)
Module: ARIYI45
Trace level: 1
Type: Module entry
Description: Displays input variable PTRBUFC (level 2)

Trace point: 2060
Subcomponent: Storage (I/O)
Module: ARIYI45
Trace level: 1
Type: Module exit
Description: Displays no variables

Trace point: 2061
Subcomponent: Storage (I/O)
Module: ARIYI46
Trace level: 1
Type: Module entry
Description: Displays input variables: P and TYPE (both level 2)

Trace point: 2062
Subcomponent: Storage (I/O)
Module: ARIYI46
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 2063
Subcomponent: Storage (I/O)
Module: ARIYI48
Trace level: 1
Type: Module entry
Description: Displays input variable PADDR (level 2)

Trace point: 2064
Subcomponent: Storage (I/O)
Module: ARIYI48
Trace level: 1
Type: Module exit
Description: Displays no variables. No return code

Trace point: 2065
Subcomponent: Storage (I/O)
Module: ARIYI49
Trace level: 1
Type: Module entry
Description: Displays input variables: PAGENO, BUFFADDR, and WHICHLOG (all level 2)

Trace point: 2066
Subcomponent: Storage (I/O)
Module: ARIYI49
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 2067
Subcomponent: Storage (I/O)
Module: ARIYI51
Trace level: 1
Type: Module entry
Description: Displays no variables

Trace point: 2068
Subcomponent: Storage (I/O)
Module: ARIYI51
Trace level: 1
Type: Module exit
Description: Displays no variables. No return code

Trace point: 2069
Subcomponent: Storage (I/O)
Module: ARIYI36
Trace level: 1
Type: Module entry (entry point ARIYI36)
Description: Displays no variables

Trace point: 2070
Subcomponent: Storage (I/O)
Module: ARIYI36
Trace level: 1
Type: Module entry (entry point ARIYI361)
Description: Displays no variables

Trace point: 2071
Subcomponent: Storage (I/O)
Module: ARIYI36
Trace level: 1
Type: Module entry (entry point ARIYI362)
Description: Displays input variable V (level 2)

Trace point: 2072
Subcomponent: Storage (I/O)
Module: ARIYI36
Trace level: 1
Type: Module entry (entry point ARIYI363)
Description: Displays input variable V (level 2)

Trace point: 2073
Subcomponent: Storage (I/O)
Module: ARIYI36
Trace level: 1
Type: Module entry (entry point ARIYI364)
Description: Displays input variable V (level 2)

Trace point: 2074
Subcomponent: Storage (I/O)
Module: ARIYI36
Trace level: 1
Type: Module entry (entry point ARIYI365)
Description: Displays input variables: LUNDO and LREDO (all level 2)

Trace point: 2075
Subcomponent: Storage (I/O)
Module: ARIYI36
Trace level: 1
Type: Module entry (entry point ARIYI366)
Description: Displays input variables: LUNDO and LREDO (all level 2)

Trace point: 2076
Subcomponent: Storage (I/O)
Module: ARIYI36
Trace level: 1
Type: Module exit (for entry points ARIYI36, ARIYI361, ARIYI362, ARIYI363, ARIYI364, ARIYI365, ARIYI366)
Description: Displays return code

ARIYSTP (DBSS Sort Trace Point Descriptor Module)

Trace point: 2300
Subcomponent: SORT
Module: ARIYS00
Trace level: 1
Type: Module entry (for entry point ARIYS00)
Description: No input variables are displayed. This is entry point for SORT operation code call (from ARIYM00) and all inputs are displayed by DBSS entry tracing.

Trace point: 2301
Subcomponent: SORT
Module: ARIYS00
Trace level: 1
Type: Module entry (for entry point ARIYS001)
Description: Displays input variables: fields in structure BASE which is copied into YTABLE1, and structures which are pointed to by BASE (all are displayed at level 2).

Trace point: 2302
Subcomponent: SORT
Module: ARIYS00
Trace level: 1
Type: Module entry (for entry point ARIYS002)
Description: Displays input variables: fields in structure BASE which is copied into YTABLE1, and structures which are pointed to by BASE (all are displayed at level 2).

Trace Point: 2303
Subcomponent: SORT
Module: ARIYS00
Trace level: 1
Type: Module exit (for entry points ARIYS00, ARIYS001, and ARIYS002)
Description: Displays return code

Trace Point: 2304
Subcomponent: SORT
Module: ARIYS02
Trace level: 1
Type: Module entry
Description: Displays input values TRANMODE and LOGL in YTABLE1 (all at level 2).

Trace Point: 2305
Subcomponent: SORT
Module: ARIYS02
Trace level: 1
Type: Module exit
Description: Displays return code

Trace Point: 2306
Subcomponent: SORT
Module: ARIYS08
Trace level: 1
Type: Module entry
Description: Displays input values: TRANMODE and LOGL in YTABLE1 (all at level 2).

Trace Point: 2307
Subcomponent: SORT
Module: ARIYS08
Trace level: 1
Type: Module exit
Description: No variables are displayed. Return code is always 0.

Trace point: 2308
Subcomponent: SORT
Module: ARIYS09
Trace level: 1
Type: Module entry
Description: Displays input parameters: PPTR, INTPPTR, and NTUPLES (all at level 2).

Trace point: 2309
Subcomponent: SORT
Module: ARIYS09
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 2310
Subcomponent: SORT
Module: ARIYS10
Trace level: 1
Type: Module entry
Description: Displays input parameters: IDATA, PPTR, BPTR, KEYPPTR, NTUPLES, NSTRINGS, PREVSTRG, and INTQUALF (all level 2).

Trace point: 2311
Subcomponent: SORT
Module: ARIYS10
Trace level: 2
Type: Module report
Description: Displays output values (returned to ARIYS24): STUPLES, KLTH, and TKEY.

Trace point: 2312
Subcomponent: SORT
Module: ARIYS10
Trace level: 2
Type: Module report
Description: Displays return code from internal procedure KEYCOMP.

Trace point: 2313
Subcomponent: SORT
Module: ARIYS10
Trace level: 2
Type: Module report
Description: Displays working variables: CKEY, IPAGENO, and DIRECT(I).

Trace point: 2314
Subcomponent: SORT
Module: ARIYS10
Trace level: 2
Type: Module report
Description: Displays return code from internal procedure KEYCOMP.

Trace point: 2315
Subcomponent: SORT
Module: ARIYS10
Trace level: 2
Type: Module report
Description: Displays working variables: CKEY, IPAGENO, and DIRECT(I).

Trace point: 2316
Subcomponent: SORT
Module: ARIYS10
Trace level: 2
Type: Module report
Description: Displays working variables: CKEY and IPAGENO.

Trace point: 2317
Subcomponent: SORT
Module: ARIYS10
Trace level: 2
Type: Module report
Description: Displays working variables: CKEY and PREVSTRG.

Trace point: 2318
Subcomponent: SORT
Module: ARIYS10
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 2319
Subcomponent: SORT
Module: ARIYS10
Trace level: 2
Type: Module report
Description: Displays input variables to internal procedure KEYCOMP: KLTH1, KLTH2, TKEY1, TKEY2, and NSTRINGS.

Trace point: 2320
Subcomponent: SORT
Module: ARIYS11
Trace level: 1
Type: Module entry
Description: Displays input parameters: LASTPAGE and MSRTPAGE (all level 2).

Trace point: 2321
Subcomponent: SORT
Module: ARIYS11
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 2322
Subcomponent: SORT
Module: ARIYS12
Trace level: 1
Type: Module entry (for entry point ARIYS12)
Description: Displays input parameters: STRING and TUPLEPTR (all level 2).

Trace point: 2323
Subcomponent: SORT
Module: ARIYS12
Trace level: 1
Type: Module entry (for entry point ARIYS121)
Description: Displays input parameters: STRING, TUPLEPTR, and TLTH (all level 2).

Trace point: 2324
Subcomponent: SORT
Module: ARIYS12
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 2325
Subcomponent: SORT
Module: ARIYS14
Trace level: 1
Type: Module entry (for entry point ARIYS14)
Description: Displays input parameters: STRING, INTPPTR, and IFREEPTR (all level 2).

Trace point: 2326
Subcomponent: SORT
Module: ARIYS14
Trace level: 1
Type: Module entry (for entry point ARIYS141)
Description: Displays input parameters: INTPPTR and IFREEPTR (all level 2).

Trace point: 2327
Subcomponent: SORT
Module: ARIYS14
Trace level: 1
Type: Module entry (for entry point ARIYS142)
Description: Displays input parameters: INTPPTR, IFREEPTR, TLTH, and STRING (first 30 characters) (all level 2).

Trace point: 2328
Subcomponent: SORT
Module: ARIYS14
Trace level: 2
Type: Module report
Description: Displays variables: INTDATA and IFREEPTR.

Trace point: 2329
Subcomponent: SORT
Module: ARIYS14
Trace level: 1
Type: Module exit (for entry points ARIYS14, ARIYS141, and ARIYS142)
Description: Displays return code

Trace point: 2330
Subcomponent: SORT
Module: ARIYS17
Trace level: 1
Type: Module entry
Description: No variables are displayed

Trace point: 2331
Subcomponent: SORT
Module: ARIYS17
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 2332
Subcomponent: SORT
Module: ARIYS18
Trace level: 1
Type: Module entry
Description: Variable N (number of displacements in TOSORT) is displayed (level 2).

Trace point: 2333
Subcomponent: SORT
Module: ARIYS18
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 2334
Subcomponent: SORT
Module: ARIYS19
Trace level: 1
Type: Module entry
Description: Variable N (number of displacements in TOSORT) and FIXEDKEY are displayed (level 2).

Trace point: 2335
Subcomponent: SORT
Module: ARIYS19.
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 2336
Subcomponent: SORT
Module: ARIYS20.
Trace level: 1
Type: Module entry
Description: Variables INTSEGM, NSTRINGS, MINTPAGE, SRTPAGE, and PREVSTRG are displayed (all level 2).

Trace point: 2337
Subcomponent: SORT
Module: ARIYS20
Trace level: 2
Type: Intermediate
Description: Displays variables NSTRINGS, STRNGCNT, STRPGCNT, MERGPASS, and CSTRINGS.

Trace point: 2338
Subcomponent: SORT
Module: ARIYS20.
Trace level: 2
Type: Intermediate
Description: Displays variables NSTRINGS, STRNGCNT, STRPGCNT, MERGPASS, and CSTRINGS.

Trace point: 2339
Subcomponent: SORT
Module: ARIYS20
Trace level: 2
Type: Intermediate
Description: Displays variables CINSEGM, and LOGPAGE.

Trace point: 2340
Subcomponent: SORT
Module: ARIYS20
Trace level: 2
Type: Intermediate
Description: It displays variables NEXTSTRG, and NEXTPAGE.

Trace point: 2341
Subcomponent: SORT
Module: ARIYS20
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 2342
Subcomponent: SORT
Module: ARIYS21
Trace level: 1
Type: Module entry
Description: Variables TUPLEPTR, RPTR, and the first 30 bytes of the row are displayed (all level 2).

Trace point: 2343
Subcomponent: SORT
Module: ARIYS21
Trace level: 2
Type: Intermediate
Description: Displays the entire output row (STRING1) and the pointer to it (RPTR).

Trace point: 2344
Subcomponent: SORT
Module: ARIYS21
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 2345
Subcomponent: SORT
Module: ARIYS22
Trace level: 1
Type: Module entry
Description: Variables COUTSEGM and COUTPAGE are displayed (all level 2).

Trace point: 2346
Subcomponent: SORT
Module: ARIYS22
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 2347
Subcomponent: SORT
Module: ARIYS23
Trace level: 1
Type: Module entry
Description: Variables COUTSEGM and COUTPAGE are displayed (both level 2).

Trace point: 2348
Subcomponent: SORT
Module: ARIYS23
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 2349
Subcomponent: SORT
Module: ARIYS24
Trace level: 1
Type: Module entry
Description: Variables SCBGATE and INTICOMP are displayed (both level 2).

Trace point: 2350
Subcomponent: SORT
Module: ARIYS24
Trace level: 1
Type: Module entry (for entry point ARIYS241)
Description: Input parameters SCBGATE, and INTICOMP are displayed (both level 2).

Trace point: 2351
Subcomponent: SORT
Module: ARIYS24
Trace level: 1
Type: Module entry (for entry point ARIYS242)
Description: Displays variables MSRTPAGE and INDEXCRN (both level 2).

Trace point: 2352
Subcomponent: SORT
Module: ARIYS24
Trace level: 1
Type: Module entry (for entry point ARIYS243)
Description: Input parameters LPAGE and LDISP are displayed (both level 2).

Trace point: 2353
Subcomponent: SORT
Module: ARIYS24
Trace level: 1
Type: Module exit
Description: Return code (level 1) and variables KEYPPTR, PPTR, and BPTR are displayed (all level 2).

ARIYXTP (DBSS Index Trace Point Descriptor Module)

Trace point: 2600
Subcomponent: Index
Module: ARIYX00
Trace level: 1
Type: Module entry
Description: No input variables are displayed

Trace point: 2601
Subcomponent: Index
Module: ARIYX00
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 2602
Subcomponent: Index
Module: ARIYX01
Trace level: 1
Type: Module entry
Description: Displays input parameter SAVECODE and implicit input variables from YTABLE1: PRESIID, ITID, IKEYLTH, and IKEYVAL (all Level 2).

Trace point: 2603
Subcomponent: Index
Module: ARIYX01
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 2604
Subcomponent: Index
Module: ARIYX02
Trace level: 1
Type: Module entry
Description: No input variables are displayed.

Trace point: 2605
Subcomponent: Index
Module: ARIYX02
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 2606
Subcomponent: Index
Module: ARIYX03
Trace level: 1
Type: Module entry
Description: No input variables are displayed.

Trace point: 2607
Subcomponent: Index
Module: ARIYX03
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 2608
Subcomponent: Index
Module: ARIYX04
Trace level: 1
Type: Module entry
Description: No input variables are displayed.

Trace point: 2609
Subcomponent: Index
Module: ARIYX04
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 2610
Subcomponent: Index
Module: ARIYX05
Trace level: 1
Type: Module entry
Description: Implicit input variables PRESIID, YTCMPCOD, IKEYLTH, IKEYVAL and IKEYGEN from YTABLE1 are displayed (all level 2).

Trace point: 2611
Subcomponent: Index
Module: ARIYX05
Trace level: 1
Type: Module exit
Description: Displays output variables: return code (level 1) and ITID in YTABLE1 (level 2).

Trace point: 2612
Subcomponent: Index
Module: ARIYX06
Trace level: 1
Type: Module entry
Description: No input variables are displayed.

Trace point: 2613
Subcomponent: Index
Module: ARIYX06
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 2614
Subcomponent: Index
Module: ARIYX07
Trace level: 1
Type: Module entry
Description: No input variables are displayed.

Trace point: 2615
Subcomponent: Index
Module: ARIYX07
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 2616
Subcomponent: Index
Module: ARIYX08
Trace level: 1
Type: Module entry
Description: No input variables are displayed.

Trace point: 2617
Subcomponent: Index
Module: ARIYX08
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 2618
Subcomponent: Index
Module: ARIYX09
Trace level: 1
Type: Module entry
Description: No input variables are displayed.

Trace point: 2619
Subcomponent: Index
Module: ARIYX09
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 2620
Subcomponent: Index
Module: ARIYX10
Trace level: 1
Type: Module entry
Description: No input variables are displayed.

Trace point: 2621
Subcomponent: Index
Module: ARIYX10
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 2622
Subcomponent: Index
Module: ARIYX11
Trace level: 1
Type: Module entry
Description: No input variables are displayed.

Trace point: 2623
Subcomponent: Index
Module: ARIYX11
Trace level: 1
Type: Module exit
Description: Displays return code (level 1) and NPAGE (level 2).

Trace point: 2624
Subcomponent: Index
Module: ARIYX12
Trace level: 1
Type: Module entry (for entry point ARIYX12)
Description: Displays no variables

Trace point: 2625
Subcomponent: Index
Module: ARIYX12
Trace level: 1
Type: Module entry (for entry point ARIYX121)
Description: Displays no variables

Trace point: 2626
Subcomponent: Index
Module: ARIYX12
Trace level: 1
Type: Module exit (for entry points ARIYX12 and ARIYX121)
Description: Displays output variables: return code (level 1), and YTABLE1 values IKEYOLTH, IKEYOVAL, and ITID (all level 2).

Trace point: 2627
Subcomponent: Index
Module: ARIYX14
Trace level: 1
Type: Module entry
Description: No input variables are displayed.

Trace point: 2628
Subcomponent: Index
Module: ARIYX14
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 2629
Subcomponent: Index
Module: ARIYX15
Trace level: 1
Type: Module entry
Description: Input variables NEWSON and PAGEID are displayed (both level 2).

Trace point: 2630
Subcomponent: Index
Module: ARIYX15
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 2631
Subcomponent: Index
Module: ARIYX16
Trace level: 1
Type: Module entry
Description: No variables are displayed

Trace point: 2632
Subcomponent: Index
Module: ARIYX16
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 2633
Subcomponent: Index
Module: ARIYX18
Trace level: 1
Type: Module entry
Description: Input parameter SAVECODE and YTABLE1 variables IKEYLTH and IKEYVAL are displayed (all level 2).

Trace point: 2634
Subcomponent: Index
Module: ARIYX18
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 2635
Subcomponent: Index
Module: ARIYX19
Trace level: 1
Type: Module entry
Description: Displays input parameter LTH (level 2)

Trace point: 2636
Subcomponent: Index
Module: ARIYX19
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 2637
Subcomponent: Index
Module: ARIYX20
Trace level: 1
Type: Module entry
Description: Displays Input parameters: QUALCODE, SAVECODE, and COMPCODE (all level 2). Displays implicit inputs from YTABLE1: PRESIID, IKEYLTH, IKEYVAL, and IKEYGEN (all level 2).

Trace point: 2638
Subcomponent: Index
Module: ARIYX20
Trace level: 1
Type: Module exit
Description: Displays return code (level 1) and YTABLE1 field ITID (level 2).

Trace point: 2639
Subcomponent: Index
Module: ARIYX21
Trace level: 1
Type: Module entry
Description: Input parameters OPCODE and SAVECODE, and YTABLE1 fields PRESIID, IKEYLTH, and IKEYVAL are displayed (all level 2).

Trace point: 2640
Subcomponent: Index
Module: ARIYX21
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 2641
Subcomponent: Index
Module: ARIYX22
Trace level: 1
Type: Module entry
Description: No variables are displayed

Trace point: 2642
Subcomponent: Index
Module: ARIYX22
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 2643
Subcomponent: Index
Module: ARIYX23
Trace level: 1
Type: Module entry
Description: YTABLE1 fields ITID, IKEYOLTH, and IKEYOVAL are displayed (all level 2).

Trace point: 2644
Subcomponent: Index
Module: ARIYX23
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 2645
Subcomponent: Index
Module: ARIYX24
Trace level: 1
Type: Module entry
Description: No variables are displayed

Trace point: 2646
Subcomponent: Index
Module: ARIYX24
Trace level: 2
Type: Module report
Description: Displays local variables: FLKEY, and FKEYLEN, from page header pointed to by register 2 and ITID (from YTABLE1). Note: Trace point is invoked if a key is found during the page search.

Trace point: 2647
Subcomponent: Index
Module: ARIYX24
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 2648
Subcomponent: Index
Module: ARIYX25
Trace level: 1
Type: Module entry
Description: No variables are displayed

Trace point: 2649
Subcomponent: Index
Module: ARIYX25
Trace level: 1
Type: Module exit
Description: Displays return code. Local variables FNLSON, FNLKEY and YTABLE1 field FKEYLEN are displayed (all level 2).

Trace point: 2650
Subcomponent: Index
Module: ARIYX26
Trace level: 1
Type: Module entry
Description: No variables are displayed

Trace point: 2651
Subcomponent: Index
Module: ARIYX26
Trace level: 2
Type: Module report
Description: Displays local variables: VLKLTH and VLKEY, from page header pointed to by register 2, and ITID (from YTABLE1). Note: trace point is invoked if a key is found during the page search.

Trace point: 2652
Subcomponent: Index
Module: ARIYX26
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 2653
Subcomponent: Index
Module: ARIYX27
Trace level: 1
Type: Module entry
Description: No fields are displayed.

Trace point: 2654
Subcomponent: Index
Module: ARIYX27
Trace level: 1
Type: Module exit
Description: Return code (level 1), and local variables VNLSON, VNLKYLTH, VNLKEY, and VNLKYLN5 (all level 2) are displayed.

Trace point: 2655
Subcomponent: Index
Module: ARIYX28
Trace level: 1
Type: Module entry
Description: No variables are displayed

Trace point: 2656
Subcomponent: Index
Module: ARIYX28
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 2657
Subcomponent: Index
Module: ARIYX29
Trace level: 1
Type: Module entry
Description: Local variable PAGEID is displayed (level 2).

Trace point: 2658
Subcomponent: Index
Module: ARIYX29
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 2659
Subcomponent: Index
Module: ARIYX30
Trace level: 1
Type: Module entry
Description: No variables are displayed

Trace point: 2660
Subcomponent: Index
Module: ARIYX30
Trace level: 1
Type: Module exit
Description: Displays return code

ARIYZTP (DBSS Update Statistics Trace Point Descriptor)

Note: ARIYZTP also contains Service Temporary trace points (9900 - 9909). See page 435.

Trace point: 2900
Subcomponent: Update Statistics
Module: ARIYZ00
Trace level: 1
Type: Module entry
Description: Displays YTABLE1 variable LDRFLAGS (level 2).

Trace point: 2901
Subcomponent: Update Statistics
Module: ARIYZ00
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 2902
Subcomponent: Update Statistics
Module: ARIYZ01.
Trace level: 1
Type: Module entry
Description: No variables are displayed.

Trace point: 2903
Subcomponent: Update Statistics
Module: ARIYZ01.
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 2904
Subcomponent: Update Statistics
Module: ARIYZ02.
Trace level: 1
Type: Module entry
Description: No variables are displayed.

Trace point: 2905
Subcomponent: Update Statistics
Module: ARIYZ02.
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 2906
Subcomponent: Update Statistics
Module: ARIYZ03.
Trace level: 1
Type: Module entry
Description: Local variables LDSEGMNO, LDRID, LDPID, LDTYPE are displayed (all level 2).

Trace point: 2907
Subcomponent: Update Statistics
Module: ARIYZ03
Trace level: 1
Type: Module exit
Description: Displays return code (Level 1). Displays output variables: TEMPAREA, PARNTREL, PARNTSEG, TEMPCPRE, AND TNOFREL (all level 2).

Trace point: 2908
Subcomponent: Update Statistics
Module: ARIYZ04.
Trace level: 1
Type: Module entry
Description: No variables are displayed.

Trace point: 2909
Subcomponent: Update Statistics
Module: ARIYZ04.
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 2910
Subcomponent: Update Statistics
Module: ARIYZ05.
Trace level: 1
Type: Module entry
Description: No variables are displayed.

Trace point: 2911
Subcomponent: Update Statistics
Module: ARIYZ05.
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 2912
Subcomponent: Update Statistics
Module: ARIYZ06
Trace level: 1
Type: Module entry (for entry point ARIYZ06)
Description: Displays no variables.

Trace point: 2913
Subcomponent: Update Statistics
Module: ARIYZ06
Trace level: 1
Type: Module entry (for entry point ARIYZ061)
Description: Displays input variable MODEFLAG (Level 2).

Trace point: 2914
Subcomponent: Update Statistics
Module: ARIYZ06
Trace level: 1
Type: Module entry (for entry point ARIYZ062)
Description: Displays input variable MODEFLAG (level 2).

Trace point: 2915
Subcomponent: Update Statistics
Module: ARIYZ06
Trace level: 1
Type: Module entry (entry point ARIYZ063)
Description: Displays input variable MODEFLAG (level 2).

Trace point: 2916
Subcomponent: Update Statistics
Module: ARIYZ06
Trace level: 1
Type: Module exit (for entry points ARIYZ06, ARIYZ061, ARIYZ062, and ARIYZ063)
Description: Displays return code

Trace point: 2917
Subcomponent: Update Statistics
Module: ARIYZ10
Trace level: 1
Type: Module entry
Description: No variables are displayed.

Trace point: 2918
Subcomponent: Update Statistics
Module: ARIYZ10
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 2919
Subcomponent: Update Statistics
Module: ARIYZ11
Trace level: 1
Type: Module entry
Description: No variables are displayed.

Trace point: 2920
Subcomponent: Update Statistics
Module: ARIYZ11
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 2921
Subcomponent: Update Statistics
Module: ARIYZ15
Trace level: 1
Type: Module entry
Description: No variables are displayed.

Trace point: 2922
Subcomponent: Update Statistics
Module: ARIYZ15
Trace level: 1
Type: Module exit
Description: Displays return code

Trace point: 2923
Subcomponent: Update Statistics
Module: ARIYZ19
Trace level: 1
Type: Module entry
Description: No variables are displayed.

Trace point: 2924
Subcomponent: Update Statistics
Module: ARIYZ16
Trace level: 1
Type: Module exit
Description: Displays return code

End of DBSS Update Statistics Trace Points. Start Working Storage (analysis) Trace Points

Trace point: 2930
Subcomponent: Working Storage (Analysis). Traced under Update Statistics.
Module: ARICWSG
Trace level: 2
Type: Module Report (error case in ARICWSG)
Description: Displays working storage analysis element (WSAELEM)

Trace point: 2931
Subcomponent: Working Storage (Analysis). Traced under Update Statistics.
Module: ARICWSG
Trace level: 2
Type: Module Report (successful case in ARICWSG)
Description: Displays working storage analysis element (WSAELEM)

Trace point: 2935
Subcomponent: Working Storage (Analysis). Traced under Update Statistics.
Module: ARICWSF
Trace level: 2
Type: Module Report (error case in ARICWSF)
Description: Displays working storage analysis element (WSAELEM)

Trace point: 2936
Subcomponent: Working Storage (Analysis). Traced under Update Statistics.
Module: ARICWSF
Trace level: 2
Type: Module Report (successful case in ARICWSF)
Description: Displays working storage analysis element (WSAELEM)

End of Working Storage (analysis) Trace Points

DBSS Service Temporary Trace Point Descriptors

Note: DBSS Service Temporary Trace Points (9900 - 9909) are located here because they are contained in module ARIYZTP.

Trace point: 9900 (hex 26AC)
Subcomponent: DBSS Service Temporary (any subcomponent) (Displays as Update Statistics: COMP=STAT)
Module: Any DBSS/DSC (Displays 'DBSSANY1')
Trace level: 1
Type: Report
Description: Provides trace output for a temporary trace point inserted into any DBSS or DSC module. Displays up to four passed local variables:
- The first is called VARNRA1 and is assumed 4 bytes, displayed hex.
- The second is called VARNRA2 and is assumed to be 16 bytes, displayed hexdump.
- The third is called VARNRA3 and is assumed to be 16 bytes, displayed hexdump.
- The fourth is called VARNRA4 and is assumed to be 64 bytes, displayed hexdump.
- To suppress display of any of the above, pass 0 data address in call parameter array.

Displays all of YTABLE1, YRSSCVT and Agent TPMAP in hexdump. All output is displayed level 1 (only requiring trace to be active for the current agent).

Trace point: 9901 (hex 26AD)
Subcomponent: DBSS Service Temporary (any subcomponent)
(Displays as Update Statistics: COMP=STAT)
Module: Any DBSS/DSC (Displays 'DBSSANY2')
Trace level: 1
Type: Report
Description: Provides trace output for a temporary trace point inserted into any DBSS or DSC module. Displays up to four passed local variables:
- The first is called VARNRB1 and is assumed 4 bytes, displayed hex.
- The second is called VARNRB2 and is assumed to be 16 bytes, displayed hexdump.
- The third is called VARNRB3 and is assumed to be 16 bytes, displayed hexdump.
- The fourth is called VARNRB4 and is assumed to be 64 bytes, displayed hexdump.
- To suppress display of any of the above, pass 0 data address in call parameter array.

Displays all of YTABLE1, YRSSCVT and Agent TPMAP in hexdump. All output is displayed level 1 (only requiring trace to be active for the current agent)

Trace point: 9902 (hex 26AE)
Subcomponent: DBSS Service Temporary (any subcomponent)
(Displays as Update Statistics: COMP=STAT)
Module: Any DBSS/DSC (Displays 'DBSSANY3')
Trace level: 1
Type: Report
Description: Provides trace output for a temporary trace point inserted into any DBSS or DSC module. Displays up to four passed local variables:
- The first is called VARNRC1 and is assumed 4 bytes, displayed hex.
- The second is called VARNRC2 and is assumed to be 16 bytes, displayed hexdump.
- The third is called VARNRC3 and is assumed to be 16 bytes, displayed hexdump.
- The fourth is called VARNRC4 and is assumed to be 64 bytes, displayed hexdump.
- To suppress display of any of the above, pass 0 data address in call parameter array.

All output is displayed level 1 (only requiring trace to be active for the current agent).

Trace point: 9903 (hex 26AF)
Subcomponent: DBSS Service Temporary (any subcomponent)
(Displays as Update Statistics: COMP=STAT)
Module: Any DBSS/DSC (Displays 'DBSSANY4')
Trace level: 1
Type: Report
Description: Provides trace output for a temporary trace point inserted into any DBSS or DSC module. Displays up to four passed local variables:
- The first is called VARNRD1 and is assumed 4 bytes, displayed hex.
- The second is called VARNRD2 and is assumed to be 16 bytes, displayed hexdump.
- The third is called VARNRD3 and is assumed to be 16 bytes, displayed hexdump.
- The fourth is called VARNRD4 and is assumed to be 64 bytes, displayed hexdump.
- To suppress display of any of the above, pass 0 data address in call parameter array.

All output is displayed level 1 (only requiring trace to be active for the current agent).

Trace point: 9904 (hex 26B0)
Subcomponent: DBSS Service Temporary (any subcomponent)
(Displays as Update Statistics: COMP=STAT)
Module: Any DBSS/DSC (Displays 'DBSSANY5')
Trace level: 1
Type: Report
Description: Provides trace output for a temporary trace point inserted into any DBSS or DSC module. Displays no local variables. Displays all of YTABLE1, YRSSCVT and Agent TPMAP in hexdump. All output is displayed level 1 (only requiring trace to be active for the current agent).

Trace point: 9905 (hex 26B1)
Subcomponent: DBSS Service Temporary (any subcomponent)
(Displays as Update Statistics: COMP=STAT)
Module: Any DBSS/DSC (Displays 'DBSSANY6')
Trace level: 1
Type: Report
Description: Provides trace output for a temporary trace point inserted into any DBSS or DSC module.
Displays up to two passed local variables:
- The first is called VARNRE1 and is assumed 4 bytes, displayed hex.
- The second is called VARNRE2 and is assumed to be 16 bytes, displayed hexdump.
- To suppress display of any of the above, pass 0 data address in call parameter array.

All output is displayed level 1 (only requiring trace to be active for the current agent).

Trace point: 9906 (hex 26B2)
Subcomponent: DBSS Service Temporary (any subcomponent)
(Displays as Update Statistics: COMP=STAT)
Module: Any DBSS/DSC (Displays 'DBSSANY7')
Trace level: 1
Type: Report
Description: Provides trace output for a temporary trace point inserted into any DBSS or DSC module.
Displays up to two passed local variables:
- The first is called VARNRF1 and is assumed 4 bytes, displayed hex.
- The second is called VARNRF2 and is assumed to be 64 bytes, displayed hexdump.
- To suppress display of any of the above, pass 0 data address in call parameter array.

All output is displayed level 1 (only requiring trace to be active for the current agent).

Trace point: 9907 (hex 26B3)
Subcomponent: DBSS Service Temporary (any subcomponent)
(Displays as Update Statistics: COMP=STAT)
Module: Any DBSS/DSC (Displays 'DBSSANY8')
Trace level: 1
Type: Report
Description: Provides trace output for a temporary trace point inserted into any DBSS or DSC module.
Displays up to two passed local variables:
- The first is called VARNRG1 and is assumed 4 bytes, displayed hex.
- The second is called VARNRG2 and its length is passed as a parameter (a fullword) immediately following the variable itself. The variable is displayed hexdump.
- To suppress display of any of the above, pass 0 data address in call parameter array.

All output is displayed level 1 (only requiring trace to be active for the current agent).

Trace point: 9908 (hex 26B4)
Subcomponent: DBSS Service Temporary (any subcomponent)
(Displays as Update Statistics: COMP=STAT)
Module: Any DBSS/DSC (Displays 'DBSSANY9')
Trace level: 1
Type: Report
Description: Provides trace output for a temporary trace point inserted into any DBSS or DSC module.
Displays one passed local variable:
- Called VARNRH1 and its length is passed as a parameter (a fullword) immediately following the variable itself. The variable is displayed hexdump.

All output is displayed level 1 (only requiring trace to be active for the current agent).

Trace point: 9909 (hex 26B5)
Subcomponent: DBSS Service Temporary (any subcomponent)
(Displays as Update Statistics: COMP=STAT)
Module: Any DBSS/DSC (Displays 'DBSSANYA')
Trace level: 1
Type: Report
Description: Provides trace output for a temporary trace point inserted into any DBSS or DSC module.
Displays one passed local variable:
- Called VARNRI1 and its length is passed as a parameter (a fullword) immediately following the variable itself. The variable is displayed hexdump.

Displays all of YTABLE1, YRSSCVT and Agent TFMAP in hexdump.

All output is displayed level 1 (only requiring trace to be active for the current agent).

ARIXTRO (RDS Executive Trace Point Descriptor Module)

```
Trace point:  4000
Subcomponent: Executive
Module:       ARIXERD
Trace level:  1
Type:         Module Entry
Description:  Entry to ARIXERD

Trace point:  4001
Subcomponent: Executive
Module:       ARIXERD
Trace level:  1
Type:         Report
Description:  ARIXERD dispatched by ARICMUD

Trace point:  4002
Subcomponent: Executive
Module:       ARIXERD
Trace level:  2
Type:         Report
Description:  Contents of RDIIN

Trace point:  4003
Subcomponent: Executive
Module:       ARIXERD
Trace level:  2
Type:         Report
Description:  Contents of SQLCA upon return to user

Trace point:  4004
Subcomponent: Executive
Module:       ARIXERD
Trace level:  1
Type:         Module exit
Description:  Exit from ARIXERD

Trace point:  4005
Subcomponent: Executive
Module:       ARIXEDS
Trace level:  1
Type:         Module entry
Description:  Entry to ARIXEDS

Trace point:  4006
Subcomponent: Executive
Module:       ARIXEDS
Trace level:  2
Type:         Report
Description:  Display input NAMELIST passed to user

Trace point:  4007
Subcomponent: Executive
Module:       ARIXEDS
Trace level:  2
Type:         Report
Description:  Display output NAMELIST passed to user

Trace point:  4008
Subcomponent: Executive
Module:       ARIXEDS
Trace level:  2
Type:         Report
Description:  SLT Entry passed back to user

Trace point:  4009
Subcomponent: Executive
Module:       ARIXEDS
Trace level:  2
Type:         Report
Description:  RDIIN passed back to user

Trace point:  4010
Subcomponent: Executive
Module:       ARIXEDS
Trace level:  1
Type:         Module exit
Description:  Exit from ARIXEDS

Trace point:  4011
Subcomponent: Executive
Module:       ARIXEPP
Trace level:  1
Type:         Module entry
Description:  Entry to ARIXEPP

Trace point:  4012
Subcomponent: Executive
Module:       ARIXEPP
Trace level:  1
Type:         Module exit
Description:  Exit from ARIXEPP
```

```
Trace point:    4013
Subcomponent:   Executive
Module:         ARIXEDP
Trace level:    1
Type:           Module entry
Description:    Entry to ARIXEDP

Trace point:    4014
Subcomponent:   Executive
Module:         ARIXEDP
Trace level:    1
Type:           Report
Description:    SQL statement ARIXEDP will process

Trace point:    4015
Subcomponent:   Executive
Module:         ARIXEDP
Trace level:    1
Type:           Module exit
Description:    Exit from ARIXEDP

Trace point:    4016
Subcomponent:   Executive
Module:         ARIXERP
Trace level:    1
Type:           Module entry
Description:    Entry to  ARIXERP

Trace point:    4017
Subcomponent:   Executive
Module:         ARIXERP
Trace level:    2
Type:           Report
Description:    Fetching entry from SYSACCESS

Trace point:    4018
Subcomponent:   Executive
Module:         ARIXERP
Trace level:    2
Type:           Report
Description:    Return from fetch entry from SYSACCESS

Trace point:    4019
Subcomponent:   Executive
Module:         ARIXERP
Trace level:    2
Type:           Report
Description:    Unlocking SYSACCESS catalog

Trace point:    4020
Subcomponent:   Executive
Module:         ARIXERP
Trace level:    1
Type:           Module exit
Description:    Exit from ARIXERP

Trace point:    4021
Subcomponent:   Executive
Module:         ARIXESX
Trace level:    1
Type:           Report
Description:    Entry to entry point ARIXESX1

Trace point:    4022
Subcomponent:   Executive
Module:         ARIXESX
Trace level:    1
Type:           Report
Description:    Exit from entry at ARIXESX1

Trace point:    4023
Subcomponent:   Executive
Module:         ARIXESX
Trace level:    1
Type:           Report
Description:    Entry to entry point ARIXESX3

Trace point:    4024
Subcomponent:   Executive
Module:         ARIXESX
Trace level:    1
Type:           Report
Description:    Exit from entry at ARIXESX3

Trace point:    4025
Subcomponent:   Executive
Module:         ARIXESX
Trace level:    1
Type:           Report
Description:    Entry to entry point ARIXESX2

Trace point:    4026
Subcomponent:   Executive
Module:         ARIXESX
Trace level:    1
Type:           Report
Description:    Exit fron entry at ARIXESX2
```

Trace point: 4027
Subcomponent: Executive
Module: ARIXESX
Trace level: 2
Type: Report
Description: Fetching RID from SYSACCESS catalog

Trace point: 4028
Subcomponent: Executive
Module: ARIXESX
Trace level: 2
Type: Report
Description: Return from fetching RID from SYSACCESS catalog

Trace point: 4029
Subcomponent: Executive
Module: ARIXESX
Trace level: 2
Type: Report
Description: Generating new access module

Trace point: 4030
Subcomponent: Executive
Module: ARIXESX
Trace level: 2
Type: Report
Description: Return from generating new access module

Trace point: 4031
Subcomponent: Executive
Module: ARIXESX
Trace level: 2
Type: Report
Description: Generating unary link for new access module

Trace point: 4032
Subcomponent: Executive
Module: ARIXESX
Trace level: 2
Type: Report
Description: Return from generating unary link

Trace point: 4033
Subcomponent: Executive
Module: ARIXESX
Trace level: 2
Type: Report
Description: Inserting length row

Trace point: 4034
Subcomponent: Executive
Module: ARIXESX
Trace level: 2
Type: Report
Description: Return from inserting length row

Trace point: 4035
Subcomponent: Executive
Module: ARIXESX
Trace level: 2
Type: Report
Description: Inserting row in SYSACCESS catalog

Trace point: 4036
Subcomponent: Executive
Module: ARIXESX
Trace level: 2
Type: Report
Description: Return from insert row in SYSACCESS catalog

Trace point: 4037
Subcomponent: Executive
Module: ARIXESX
Trace level: 2
Type: Report
Description: Update row in SYSACCESS catalog

Trace point: 4038
Subcomponent: Executive
Module: ARIXESX
Trace level: 2
Type: Report
Description: Return from updating SYSACCESS catalog

Trace point: 4039
Subcomponent: Executive
Module: ARIXESX
Trace level: 2
Type: Report
Description: Inserting access module into the data base

Trace point: 4040
Subcomponent: Executive
Module: ARIXEDS
Trace level: 2
Type: Report
Description: Return from insert of access module into data base

Trace point: 4041
Subcomponent: Executive
Module: ARIXESX
Trace level: 2
Type: Report
Description: Updating length row in SYSACCESS catalog

Trace point: 4042
Subcomponent: Executive
Module: ARIXESX
Trace level: 2
Type: Report
Description: Return from updating length row

Trace point: 4043
Subcomponent: Executive
Module: ARIXELX
Trace level: 1
Type: Module entry
Description: Entry to ARIXELX

Trace point: 4044
Subcomponent: Executive
Module: ARIXELX
Trace level: 1
Type: Module exit
Description: Exit from ARIXELX

Trace point: 4045
Subcomponent: Executive
Module: ARIXELX
Trace level: 1
Type: Report
Description: Entry to entry point ARIXELX1

Trace point: 4046
Subcomponent: Executive
Module: ARIXELX
Trace level: 1
Type: Module exit
Description: Exit from ARIXELX1

Trace point: 4047
Subcomponent: Executive
Module: ARIXELX
Trace level: 2
Type: Report
Description: ARIXELX issuing BEGIN WORK

Trace point: 4048
Subcomponent: Executive
Module: ARIXELX
Trace level: 2
Type: Report
Description: Fetching row from SYSACCESS catalog

Trace point: 4049
Subcomponent: Executive
Module: ARIXELX
Trace level: 2
Type: Report
Description: Return from fetch row from SYSACCESS catalog

Trace point: 4050
Subcomponent: Executive
Module: ARIXELX
Trace level: 2
Type: Report
Description: ARIXELX issuing commit work

Trace point: 4051
Subcomponent: Executive
Module: ARIXELX
Trace level: 2
Type: Report
Description: Moving part of access module from data base to
real storage

Trace point: 4052
Subcomponent: Executive
Module: ARIXELX
Trace level: 2
Type: Report
Description: Return from fetch of part of access module

Trace point: 4053
Subcomponent: Executive
Module: ARIXELX
Trace level: 2
Type: Report
Description: Unlock SYSACCESS catalog

Trace point: 4054
Subcomponent: Executive
Module: ARIXELX
Trace level: 2
Type: Report
Description: Return from unlocking SYSACCESS catalog

```
Trace point:  4055
Subcomponent: Executive
Module:       ARIXELX
Trace level:  2
Type:         Report
Description:  Open the access module

Trace point:  4056
Subcomponent: Executive
Module:       ARIXELX
Trace level:  2
Type:         Report
Description:  Return from opening the access module

Trace point:  4057
Subcomponent: Executive
Module:       ARIXELX
Trace level:  2
Type:         Report
Description:  Close scan

Trace point:  4058
Subcomponent: Executive
Module:       ARIXELX
Trace level:  2
Type:         Report
Description:  Open scan

Trace point:  4059
Subcomponent: Executive
Module:       ARIXELX
Trace level:  2
Type:         Report
Description:  Fetch next

Trace point:  4060
Subcomponent: Executive
Module:       ARIXELX
Trace level:  2
Type:         Report
Description:  Delete where current of cursor

Trace point:  4061
Subcomponent: Executive
Module:       ARIXEAB
Trace level:  1
Type:         Module entry
Description:  Entry to ARIXEAB

Trace point:  4062
Subcomponent: Executive
Module:       ARIXEAB
Trace level:  1
Type:         Module exit
Description:  Exit from ARIXEAB

Trace point:  4063
Subcomponent: Executive
Module:       ARIXEFB
Trace level:  1
Type:         Module entry
Description:  Entry into ARIXEFB

Trace point:  4064
Subcomponent: Executive
Module:       ARIXEFB
Trace level:  1
Type:         Module exit
Description:  Exit from ARIXEFB

Trace point:  4065
Subcomponent: Executive
Module:       ARIXECW
Trace level:  1
Type:         Report
Description:  Entry to ARIXECW

Trace point:  4066
Subcomponent: Executive
Module:       ARIXECW
Trace level:  1
Type:         Module exit
Description:  Exit from ARIXECW

Trace point:  4067
Subcomponent: Executive
Module:       ARIXEBR
Trace level:  1
Type:         Module entry
Description:  Entry into ARIXEBR

Trace point:  4068
Subcomponent: Executive
Module:       ARIXEBR
Trace level:  1
Type:         Module exit
Description:  Exit from ARIXEBR
```

Trace point: 4069
Subcomponent: Executive
Module: ARIXEER
Trace level: 1
Type: Module entry
Description: Entry to ARIXEER

Trace point: 4070
Subcomponent: Executive
Module: ARIXEER
Trace level: 1
Type: Module exit
Description: Exit from ARIXEER

Trace point: 4071
Subcomponent: Executive
Module: ARIXECK
Trace level: 1
Type: Module entry
Description: Entry to ARIXECK

Trace point: 4072
Subcomponent: Executive
Module: ARIXECK
Trace level: 1
Type: Module exit
Description: Exit from ARIXECK

Trace point: 4073
Subcomponent: Executive
Module: ARIXEDC
Trace level: 1
Type: Module entry
Description: Entry to ARIXEDC

Trace point: 4074
Subcomponent: Executive
Module: ARIXEDC
Trace level: 1
Type: Module exit
Description: Exit from ARIXEDC

Trace point: 4075
Subcomponent: Executive
Module: ARIXEDB
Trace level: 1
Type: Module entry
Description: Entry to ARIXEDB

Trace point: 4076
Subcomponent: Executive
Module: ARIXEDB
Trace level: 1
Type: Module exit
Description: Exit from ARIXEDB

Trace point: 4077
Subcomponent: Executive
Module: ARIXEDP1
Trace level: 1
Type: Module entry
Description: Entry to ARIXEDP1

Trace point: 4078
Subcomponent: Executive
Module: ARIXESP
Trace level: 1
Type: Module entry
Description: Entry to ARIXESP

Trace point: 4079
Subcomponent: Executive
Module: ARIXESP
Trace level: 1
Type: Module exit
Description: Exit from ARIXESP

Trace point: 4080
Subcomponent: Executive
Module: ARIXEBR1
Trace level: 1
Type: Module entry
Description: Entry to ARIXEBR1

Trace point: 4081
Subcomponent: Executive
Module: ARIXEBR1
Trace level: 1
Type: Module exit
Description: Exit from ARIXEBR1

Trace point: 4082
Subcomponent: Executive
Module: ARIXEDC
Trace level: 2
Type: Report
Description: Number of entries in returned SQLDA

Trace point: 4083
Subcomponent: Executive
Module: ARIXEDC
Trace level: 2
Type: Report
Description: SQLDA returned to user on DESCRIBE

Trace point: 4084
Subcomponent: Executive
Module: ARIXERO
Trace level: 1
Type: Report
Description: Get catalog table

Trace point: 4085
Subcomponent: Executive
Module: ARIXECL
Trace level: 1
Type: Module entry
Description: Entry to ARIXECL

Trace point: 4086
Subcomponent: Executive
Module: ARIXECL
Trace level: 1
Type: Module exit
Description: Exit from ARIXECL

Trace point: 4087
Subcomponent: Executive
Module: ARIXEPP
Trace level: 2
Type: Report
Description: SQL statement passed to ARIXPA1

Trace point: 4088
Subcomponent: Executive
Module: ARIXERD
Trace level: 2
Type: Report
Description: Lock SYSACCESS TID

ARIXTR1 (Parser and Optimizer Trace Point Descriptor Module)

Trace point: 4400
Subcomponent: Parser
Module: ARIXPA1 (top Parser module)
Trace level: 1
Type: Module entry
Description: At trace level 1, the first 480 characters of the input SQL statement are displayed. DBCS constants will be presented incorrectly when the shift-out (so) and matching shift-in (si) characters are not on the same line, that is when the constant overflows a line.

Trace point: 4401
Subcomponent: Parser
Module: ARIXPA1
Trace level: 2
Type: Module exit
Description: Normal exit from the Parser (ARIXPA1). At trace level 2, the Parse Tree is put to the trace tape for off-line formatting.

Trace point: 4402
Subcomponent: Parser
Module: ARIXPA2 or ARIXPA3
Trace level: 1
Type: Module entry
Description: Entry into one of the Parser modules containing semantic routines (ARIXPA2 or ARIXPA3). Semantic routines are associated with BNF lines in Parser generation (syntax definition). The semantic routine number is also displayed (PSNUMBER) if trace is at level 2. These numbers are indexes into the named entry points (to ARIXPA2/ARIXPA3) contained in (generated by) ARIXPA0.

Trace point: 4403
Subcomponent: Parser
Module: ARIXPA2
Trace level: 1
Type: Module exit
Description: Exit from one of the Parser semantic routine modules (ARIXPA2). The SQLERRD1 is also displayed as the return code.

Trace point: 4405
Subcomponent: Parser
Module: ARIXPA3
Trace level: 1
Type: Module exit
Description: Exit from one of the Parser semantic routine modules (ARIXPA3). The SQLERRD1 is also displayed as the return code.

Trace point: 4406
Subcomponent: Parser
Module: ARIXPA1
Trace level: 1
Type: Module exit
Description: Exit from the Parser (ARIXPA1). SQLERRD1 is displayed as the return code.

End of RDS Parser Trace Points. Start Working Storage (analysis) (range 4410 through 4419)

Trace point: 4410
Subcomponent: Working storage (analysis)
Module: ARICWSG
Trace level: 2
Type: Report (error case in ARICWSG)
Description: Displays working storage analysis element.

Trace point: 4411
Subcomponent: Working storage (anaylsis)
Module: ARICWSG
Trace level: 2
Type: Report (successful case in ARICWSG)
Description: Displays working storage analysis element

Trace point: 4415
Subcomponent: Working storage (anaylsis)
Module: ARICWSF
Trace level: 2
Type: Report (error case in ARICWSF)
Description: Displays working storage analysis element

Trace point: 4416
Subcomponent: Working storage (anaylsis)
Module: ARICWSF
Trace level: 2
Type: Report (successful case in ARICWSF)
Description: Displays working storage analysis element

End working storage (anaylsis)

Trace point: 4600
Subcomponent: Optimizer
Module: ARIXOOP
Trace level: 1
Type: Module entry
Description: This is entry trace point for the entry module of the Optimizer.

Trace point: 4605
Subcomponent: Optimizer
Module: ARIXOQ1
Trace level: 1
Type: Module entry
Description: This is entry trace point for the module ARIXOQ1 which is the start of Pass 1 processing in the Optimizer.

Trace point: 4610
Subcomponent: Optimizer
Module: ARIXOQ1
Trace level: 1
Type: Module exit
Description: This is exit trace point for the module ARIXOQ1 which marks the end of Pass 1 processing in the Optimizer.

Trace point: 4615
Subcomponent: Optimizer
Module: ARIXOVC
Trace level: 1
Type: Module entry
Description: This is entry trace point for the module ARIXOVC, view composition.

Trace point: 4620
Subcomponent: Optimizer
Module: ARIXOVC
Trace level: 2
Type: Report
Description: This is trace point for displaying the Parse Tree which has been modified as a result of view composition.

Trace point: 4625
Subcomponent: Optimizer
Module: ARIXOVC
Trace level: 2
Type: Report
Description: This is trace point for displaying the global tables used throughout the Optimizer. At this point in Pass1, catalog lookup and view composition (if necessary) have been completed. Tables displayed are: Table array, Column Array, Index Array, and Query Array.

Trace point: 4630
Subcomponent: Optimizer
Module: ARIXOVC
Trace level: 1
Type: Module exit
Description: This is exit trace point for the module ARIXOVC, which marks the end of view composition processing in the Optimizer.

Trace point: 4635
Subcomponent: Optimizer
Module: ARIXOOP
Trace level: 1
Type: Module exit
Description: This is exit trace point for the module ARIXOOP, which marks the completion of the optimization function of RDS.

Trace point: 4700
Subcomponent: Optimizer
Module: ARIXOB2
Trace level: 1
Type: Module entry
Description: This is entry trace point for the module ARIXOB2, which is the start of Pass 2 in the Optimizer.

Trace point: 4705
Subcomponent: Optimizer
Module: ARIXOB2
Trace level: 2
Type: Report
Description: This is trace point for displaying the global table used to store predicate information. All necessary data has been stored at completion of Pass 2.

Trace point: 4710
Subcomponent: Optimizer
Module: ARIXOB2
Trace level: 1
Type: Module exit
Description: This is exit trace point for the module ARIXOB2, which marks the end of Pass 2 processing in the Optimizer.

Trace point: 4800
Subcomponent: Optimizer (path selection)
Module: ARIXOGP
Trace level: 1
Type: Module entry
Description: Entry to module ARIXOGP. Parameters are OPPTR and SPACEPTR, which should be traced already.

Trace point: 4810
Subcomponent: Optimizer (path selection)
Module: ARIXOGP
Trace level: 2
Type: Report
Description: Miniplans in access order at end of path selection. ARIXOGP will loop on this trace point to show all miniplans in the chosen path.

Trace point: 4812
Subcomponent: Optimizer (path selection)
Module: ARIXOGP
Trace level: 2
Type: Report
Description: Area allocated for COSTS and CHOICES. Data areas are dumped in hex.

Trace point: 4819
Subcomponent: Optimizer (path selection)
Module: ARIXOGP
Trace level: 1
Type: Module exit
Description: Exit from module ARIXOGP

Trace point: 4820
Subcomponent: Optimizer (path selection)
Module: ARIXOTS
Trace level: 1
Type: Module entry
Description: Entry to module ARIXOTS. QNO=INDEX in QARRAY of this query.

Trace point: 4829
Subcomponent: Optimizer (path selection)
Module: ARIXOTS
Trace level: 1
Type: Module exit
Description: Exit from module ARIXOTS

Trace point: 4830
Subcomponent: Optimizer (path selection)
Module: ARIXOCS
Trace level: 1
Type: Module entry
Description: Entry to module ARIXOCS

Trace point: 4839
Subcomponent: Optimizer (path selection)
Module: ARIXOCS
Trace level: 1
Type: Module exit
Description: Exit from module ARIXOCS

Trace point: 4840
Subcomponent: Optimizer (path selection)
Module: ARIXOGC
Trace level: 1
Type: Module entry
Description: Entry to module ARIXOGC
TABAVAIL - Bits indicate tables in query so far.
NEWTABLE - Table to be added to TABAVAIL.

Trace point: 4859
Subcomponent: Optimizer (path selection)
Module: ARIXOGC
Trace level: 1
Type: Module exit
Description: Exit from module ARIXOGC.
CARD - Number of items in query result
OPTPLANV - PLANVEC address
OPT#PLNS - Number of plans in PLANVEC
PLANVEC - Plans for each interesting order and cheapest plan

Trace point: 4860
Subcomponent: Optimizer (path selection)
Module: ARIXODF
Trace level: 1
Type: Module entry
Description: Entry to module ARIXODF

Trace point: 4879
Subcomponent: Optimizer (path selection)
Module: ARIXODF
Trace level: 1
Type: Module exit
Description: Exit from module ARIXODF

Trace point: 4880
Subcomponent: Optimizer (path selection)
Module: ARIXOSS
Trace level: 1
Type: Module entry
Description: Entry to module ARIXOSS

Trace point: 4889
Subcomponent: Optimizer (path selection)
Module: ARIXOSS
Trace level: 1
Type: Module exit
Description: Exit from module ARIXOSS (TPRET not used as ARIXOSS detects no errors)

Trace point: 4900
Subcomponent: Optimizer
Module: ARIXOGA
Trace level: 1
Type: Module entry
Description: This is entry trace point for the module ARIXOGA, which is the start of ASL (Access Specification Language) generation.

Trace point: 4905
Subcomponent: Optimizer
Module: ARIXOCT
Trace level: 1
Type: Module entry
Description: This is entry trace point for the module ARIXOCT, which is the module that allocates and initializes control blocks for DBSS calls. Space is alloted for it in Block 1.

Trace point: 4910
Subcomponent: Optimizer
Module: ARIXOCT
Trace level: 1
Type: Module exit
Description: This is exit trace point for the module ARIXOCT, which marks the end control block allocation.

Trace point: 4915
Subcomponent: Optimizer
Module: ARIXOFR
Trace level: 1
Type: Module entry
Description: This is entry trace point for the module ARIXOFR, which allocates the OPEN, NEXT, and MOD structures in the Space Block and places their location in the control block.

Trace point: 4920
Subcomponent: Optimizer
Module: ARIXOFR
Trace level: 1
Type: Module exit
Description: This is exit trace point for the module ARIXOFR.

Trace point: 4925
Subcomponent: Optimizer
Module: ARIXOGA
Trace level: 1
Type: Module exit
Description: This is exit trace point for the module ARIXOGA, which marks end of ASL generation.

Trace point: 4930
Subcomponent: Optimizer
Module: ARIXOOP
Trace level: 2
Type: Report
Description: This is trace point for dumping the Parse Tree, the Space Blocks and their respective Relocation Directories at completion of optimization. The information will later be input into an offline formatter.

ARIXTR5 (Code Generator Trace Point Descriptor Module)

Note: ARIXTR5 also contains Service Temporary trace points (9950 - 9956). See page 487.

Trace point: 5200
Subcomponent: Code Generator
Module: ARIXC37 (top module for Code generator)
Trace level: 1
Type: Module entry
Description: Entry to module ARIXC37

Trace point: 5201
Subcomponent: Code Generator
Module: ARIXC37 (top module for Code generator)
Trace level: 1
Type: Report
Description: Where to look for operands

Trace point: 5202
Subcomponent: Code Generator
Module: ARIXC37 (top module for Code generator)
Trace level: 2
Type: Report
Description: Syntactic stack working index Number of operands expected for a operator on the syntactic stack.

Trace point: 5203
Subcomponent: Code Generator
Module: ARIXC37 (top module for Code generator)
Trace level: 2
Type: Report
Description: Index to add entry to sub-query table Index to process a sub-query entry.

Trace point: 5204
Subcomponent: Code Generator
Module: ARIXC37 (top module for Code generator)
Trace level: 2
Type: Report
Description: Index to sub-query table

Trace point: 5205
Subcomponent: Code Generator
Module: ARIXC37 (top module for Code generator)
Trace level: 1
Type: Module exit
Description: Exit module ARIXC37

Trace point: 5206
Subcomponent: Code Generator
Module: ARIXC38 (create access module code from optimized data)
Trace level: 1
Type: Module entry
Description: Entry module ARIXC38

Trace point: 5207
Subcomponent: Code Generator
Module: ARIXC38 (create access module code from optimized data)
Trace level: 2
Type: Report
Description: Move prolog fragment for insert

Trace point: 5208
Subcomponent: Code Generator
Module: ARIXC38 (create access module code from optimized data)
Trace level: 2
Type: Report
Description: Processing a DELETE WHERE CURRENT OF ...

Trace point: 5209
Subcomponent: Code Generator
Module: ARIXC38 (create access module code from optimized data)
Trace level: 2
Type: Report
Description: Processing an UPDATE WHERE CURRENT OF ....

Trace point: 5210
Subcomponent: Code Generator
Module: ARIXC38 (create access module code from optimized data)
Trace level: 2
Type: Report
Description: Move prolog fragment for template 1

Trace point: 5211
Subcomponent: Code Generator
Module: ARIXC38 (create access module code from optimized data)
Trace level: 2
Type: Report
Description: Set function query without cursor

Trace point: 5212
Subcomponent: Code Generator
Module: ARIXC38 (create access module code from optimized data)
Trace level: 2
Type: Report
Description: Set function query with cursor

Trace point: 5213
Subcomponent: Code Generator
Module: ARIXC38 (create access module code from optimized data)
Trace level: 2
Type: Report
Description: Select with a cursor

Trace point: 5214
Subcomponent: Code Generator
Module: ARIXC38 (create access module code from optimized data)
Trace level: 2
Type: Report
Description: Second invocation for BQUERY node

Trace point: 5215
Subcomponent: Code Generator
Module: ARIXC38 (create access module code from optimized data)
Trace level: 2
Type: Report
Description: Last invocation for a insert

Trace point: 5216
Subcomponent: Code Generator
Module: ARIXC38 (create access module code from optimized data)
Trace level: 2
Type: Report
Description: Evaluation of boolean expression

Trace point: 5217
Subcomponent: Code Generator
Module: ARIXC38 (create access module code from optimized data)
Trace level: 2
Type: Report
Description: Evaluation of boolean expression

Trace point: 5218
Subcomponent: Code Generator
Module: ARIXC38 (create access module code from optimized data)
Trace level: 2
Type: Report
Description: Third invocation for a delete

Trace point: 5219
Subcomponent: Code Generator
Module: ARIXC38 (create access module code from optimized data)
Trace level: 2
Type: Report
Description: Third invocation for a update

Trace point: 5220
Subcomponent: Code Generator
Module: ARIXC38 (create access module code from optimized data)
Trace level: 2
Type: Report
Description: Third invocation for insert

Trace point: 5221
Subcomponent: Code Generator
Module: ARIXC38 (create access module code from optimized data)
Trace level: 2
Type: Report
Description: Third invocation for set function with cursor

Trace point: 5222
Subcomponent: Code Generator
Module: ARIXC38 (create access module code from optimized data)
Trace level: 2
Type: Report
Description: Third invocation for set function without cursor

```
Trace point:  5223
Subcomponent: Code Generator
Module:       ARIXC38 (create access module code from
              optimized data)
Trace level:  2
Type:         Report
Description:  Processing an UPDATE WHERE CURRENT OF ....

Trace point:  5224
Subcomponent: Code Generator
Module:       ARIXC38 (create access module code from
              optimized data)
Trace level:  2
Type:         Report
Description:  Fourth invocation for set function with cursor

Trace point:  5225
Subcomponent: Code Generator
Module:       ARIXC38 (create access module code from
              optimized data)
Trace level:  2
Type:         Report
Description:  Fourth invocation for set function without
              cursor

Trace point:  5226
Subcomponent: Code Generator
Module:       ARIXC38 (create access module code from
              optimized data)
Trace level:  2
Type:         Report
Description:  Semantic routine for is null node

Trace point:  5227
Subcomponent: Code Generator
Module:       ARIXC38 (create access module code from
              optimized data)
Trace level:  2
Type:         Report
Description:  Process SETNODE for update

Trace point:  5228
Subcomponent: Code Generator
Module:       ARIXC38 (create access module code from
              optimized data)
Trace level:  2
Type:         Report
Description:  Process minus operator

Trace point:  5229
Subcomponent: Code Generator
Module:       ARIXC38 (create access module code from
              optimized data)
Trace level:  2
Type:         Report
Description:  Process arithmetic operator

Trace point:  5230
Subcomponent: Code Generator
Module:       ARIXC38 (create access module code from
              optimized data)
Trace level:  2
Type:         Report
Description:  Process SETFN1 node

Trace point:  5231
Subcomponent: Code Generator
Module:       ARIXC38 (create access module code from
              optimized data)
Trace level:  2
Type:         Report
Description:  Process template 2 for select

Trace point:  5232
Subcomponent: Code Generator
Module:       ARIXC38 (create access module code from
              optimized data)
Trace level:  1
Type:         Module exit
Description:  Exit module ARIXC38

Trace point:  5233
Subcomponent: Code Generator
Module:       ARIXC02 (puts out code in block 2 to perform
              arithmetic operations)
Trace level:  1
Type:         Module entry
Description:  Entry module ARIXC02

Trace point:  5234
Subcomponent: Code Generator
Module:       ARIXC02 (puts out code in block 2 to perform
              arithmetic operations)
Trace level:  1
Type:         Module exit
Description:  Exit module ARIXC02
```

Trace point: 5235
Subcomponent: Code Generator
Module: ARIXC03 (put out code in block 2 to bind input variables)
Trace level: 1
Type: Module entry
Description: Entry module ARIXC03

Trace point: 5236
Subcomponent: Code Generator
Module: ARIXC03 (put out code in block 2 to bind input variables)
Trace level: 1
Type: Module exit
Description: Exit module ARIXC03

Trace point: 5237
Subcomponent: Code Generator
Module: ARIXC05 (puts out code in block 2 to perform compares)
Trace level: 1
Type: Module entry
Description: Entry module ARIXC05

Trace point: 5238
Subcomponent: Code Generator
Module: ARIXC05 (puts out code inblock 2 to perform compares
Trace level: 1
Type: Module exit
Description: Exit module ARIXC05

Trace point: 5239
Subcomponent: Code Generator
Module: ARIXC13 (code generator initalization routine)
Trace level: 1
Type: Module entry
Description: Entry module ARIXC13

Trace point: 5240
Subcomponent: Code Generator
Module: ARIXC13 (code generator initalization routine)
Trace level: 1
Type: Report
Description: Get address of block 0

Trace point: 5241
Subcomponent: Code Generator
Module: ARIXC13 (code generator initalization routine)
Trace level: 1
Type: Report
Description: Get address of space blocks

Trace point: 5242
Subcomponent: Code Generator
Module: ARIXC13 (code generator initalization routine)
Trace level: 1
Type: Module exit
Description: Exit module ARIXC13

Trace point: 5243
Subcomponent: Code Generator
Module: ARIXC14 (cleans up loose ends at end of Code generation)
Trace level: 1
Type: Module entry
Description: Entry module ARIXC14

Trace point: 5244
Subcomponent: Code Generator
Module: ARIXC14 (cleans up loose ends at end of Code generation)
Trace level: 1
Type: Module exit
Description: Exit module ARIXC14

Trace point: 5245
Subcomponent: Code Generator
Module: ARIXC16 (to implement is/is not null predicate)
Trace level: 1
Type: Module entry
Description: Entry module ARIXC16

Trace point: 5246
Subcomponent: Code Generator
Module: ARIXC16 (to implement is/is not null predicate)
Trace level: 1
Type: Module exit
Description: Exit module ARIXC16

Trace point: 5247
Subcomponent: Code Generator
Module: ARIXC18 (moves a fragment of machine code into code block)
Trace level: 1
Type: Module entry
Description: Entry module ARIXC18

```
Trace point:  5248
Subcomponent: Code Generator
Module:       ARIXC18 (moves a fragment of machine code into
              code block)
Trace level:  1
Type:         Module exit
Description:  Exit module ARIXC18

Trace point:  5249
Subcomponent: Code Generator
Module:       ARIXC19 (allocates a spaceblk to hold the
              compiled code for compiler sub-routine)
Trace level:  1
Type:         Module entry
Description:  Entry module ARIXC19

Trace point:  5250
Subcomponent: Code Generator
Module:       ARIXC19 (allocates a SPACEBLK to hold the
              compiled code for the compiler generated
              sub-routine.)
Trace level:  1
Type:         Module exit
Description:  Exit module ARIXC19

Trace point:  5251
Subcomponent: Code Generator
Module:       ARIXC21 (puts out code to move data items to
              user area)
Trace level:  1
Type:         Module entry
Description:  Entry module ARIXC21

Trace point:  5252
Subcomponent: Code Generator
Module:       ARIXC21 (puts out code to move data items to
              user area)
Trace level:  1
Type:         Module exit
Description:  Exit module ARIXC21

Trace point:  5253
Subcomponent: Code Generator
Module:       ARIXC22 (puts out code to return sub-query to
              a slot in block 1 where they will be used as
              input variables for a outer query)
Trace level:  1
Type:         Module entry
Description:  Entry module ARIXC22

Trace point:  5254
Subcomponent: Code Generator
Module:       ARIXC22 (puts out code to return sub-query to
              a slot in block 1 where they will be used as
              input variables for a outer query)
Trace level:  1
Type:         Module exit
Description:  Exit module ARIXC22

Trace point:  5255
Subcomponent: Code Generator
Module:       ARIXC23 (process the bind table)
Trace level:  1
Type:         Module entry
Description:  Entry module ARIXC23

Trace point:  5256
Subcomponent: Code Generator
Module:       ARIXC23 (process the bind table)
Trace level:  1
Type:         Module exit
Description:  Exit module ARIXC23

Trace point:  5257
Subcomponent: Code Generator
Module:       ARIXC24 (puts out code to call sub-routines
              for sub-queries)
Trace level:  1
Type:         Module entry
Description:  Entry module ARIXC24

Trace point:  5258
Subcomponent: Code Generator
Module:       ARIXC24 (puts out code to call sub-routines
              for sub-queries)
Trace level:  1
Type:         Module exit
Description:  Exit module ARIXC24

Trace point:  5259
Subcomponent: Code Generator
Module:       ARIXC27 (intialize the sub-query output slots
              to null)
Trace level:  1
Type:         Module entry
Description:  Entry module ARIXC27
```

Trace point: 5260
Subcomponent: Code Generator
Module: ARIXC27 (intialize the sub-query output slots to null)
Trace level: 1
Type: Module exit
Description: Exit module ARIXC27

Trace point: 5261
Subcomponent: Code Generator
Module: ARIXC28 (puts out code to initialize for set function)
Trace level: 1
Type: Module entry
Description: Entry module ARIXC28

Trace point: 5262
Subcomponent: Code Generator
Module: ARIXC28 (puts out code to initialize for set function)
Trace level: 1
Type: Module exit
Description: Exit module ARIXC28

Trace point: 5263
Subcomponent: Code Generator
Module: ARIXC29 (sets up tables and code block to place machine code into a new block for a sub-routine)
Trace level: 1
Type: Module entry
Description: Entry module ARIXC29

Trace point: 5264
Subcomponent: Code Generator
Module: ARIXC29 (sets up tables and code block to place machine code into a new block for a sub-routine)
Trace level: 1
Type: Module exit
Description: Exit module ARIXC29

Trace point: 5265
Subcomponent: Code Generator
Module: ARIXC30 (puts out code to test if end of group was encountered for the RHS of a type 2 join)
Trace level: 1
Type: Module entry
Description: Entry module ARIXC30

Trace point: 5266
Subcomponent: Code Generator
Module: ARIXC30 (puts out code to test if end of group was encountered for the RHS of a type 2 join)
Trace level: 1
Type: Module exit
Description: Exit module ARIXC30

Trace point: 5267
Subcomponent: Code Generator
Module: ARIXC31 (test end of group for method 2 join)
Trace level: 1
Type: Module entry
Description: Entry module ARIXC31

Trace point: 5268
Subcomponent: Code Generator
Module: ARIXC31 (test end of group for method 2 join)
Trace level: 1
Type: Module exit
Description: Exit module ARIXC31

Trace point: 5269
Subcomponent: Code Generator
Module: ARIXC32 (test new group for method 2 join)
Trace level: 1
Type: Module entry
Description: Entry module ARIXC32

Trace point: 5270
Subcomponent: Code Generator
Module: ARIXC32 (test new group for method 2 join)
Trace level: 1
Type: Module exit
Description: Exit module ARIXC32

Trace point: 5271
Subcomponent: Code Generator
Module: ARIXC35 (to generate code to perform part of the update)
Trace level: 1
Type: Module entry
Description: Entry module ARIXC35

Trace point: 5272
Subcomponent: Code Generator
Module: ARIXC35 (to generate code to perform part of the update)
Trace level: 1
Type: Module exit
Description: Exit module ARIXC35

```
Trace point:    5273
Subcomponent:   Code Generator
Module:         ARIXC15 (to help implememt where <expr-1> <=
                any select expr-2>)
Trace level:    1
Type:           Module entry
Description:    Entry module ARIXC15

Trace point:    5274
Subcomponent:   Code Generator
Module:         ARIXC15 (to help implememt where <expr-1> <=
                any select expr-2>)
Trace level:    1
Type:           Module exit
Description:    Exit module ARIXC15

Trace point:    5275
Subcomponent:   Code Generator
Module:         ARIXC40 (moves the RR instruction into the
                next available slot in the CODEBLK)
Trace level:    1
Type:           Report
Description:    Moving an RR type instruction into the CODEBLK

Trace point:    5276
Subcomponent:   Code Generator
Module:         ARIXC41 (moves the RX instruction into the
                next available slot in the CODEBLK)
Trace level:    1
Type:           Report
Description:    Moving an RX type instruction into the CODEBLK

Trace point:    5277
Subcomponent:   Code Generator
Module:         ARIXC42 (moves the SI instruction into the
                next available slot in the CODEBLK)
Trace level:    1
Type:           Report
Description:    Moving an SI type instruction into the CODEBLK

Trace point:    5278
Subcomponent:   Code Generator
Module:         ARIXC43 (moves the SS instruction into the
                next available slot in the CODEBLK)
Trace level:    1
Type:           Report
Description:    Moving an SS type instruction into the CODEBLK
```

ARIXTR2 (Interpreter Trace Point Descriptor Module (part 1))

```
Trace point:    5800
Subcomponent:   Interpreter
Module:         ARIXI14 (top module for Interpreter component)
Trace level:    1
Type:           Module entry
Description:    Entry to module ARIXI14

Trace point:    5801
Subcomponent:   Interpreter
Module:         ARIXI14 (top module for Interpreter component)
Trace level:    2
Type:           Report
Description:    Display ROOTTYPE

Trace point:    5805
Subcomponent:   Interpreter
Module:         ARIXI14 (top module for Interpreter component)
Trace level:    1
Type:           Module exit
Description:    Exit from module ARIXI14

Trace point:    5806
Subcomponent:   Interpreter
Module:         ARIXI01 (acquire DBSPACE)
Trace level:    1
Type:           Module entry
Description:    Entry to module ARIXI01

Trace point:    5807
Subcomponent:   Interpreter
Module:         ARIXI01 (acquire DBSPACE)
Trace level:    2
Type:           Report
Description:    Input parameters to acquire DBSPACE

Trace point:    5808
Subcomponent:   Interpreter
Module:         ARIXI01 (acquire DBSPACE)
Trace level:    2
Type:           Report
Description:    Fetch from SYSDBSPACES to look if DBSPACE with
                given name already defined

Trace point:    5809
Subcomponent:   Interpreter
Module:         ARIXI01 (acquire DBSPACE)
Trace level:    2
Type:           Report
Description:    Open scan on SYSDBSPACES to look for available
                DBSPACE
```

```
Trace point:  5810
Subcomponent: Interpreter
Module:       ARIXI01 (acquire DBSPACE)
Trace level:  2
Type:         Report
Description:  Next on SYSDBSPACES to look for available
              DBSPACE

Trace point:  5811
Subcomponent: Interpreter
Module:       ARIXI01 (acquire DBSPACE)
Trace level:  2
Type:         Report
Description:  Inserting a DBSPACE control table for acquired
              DBSPACE

Trace point:  5812
Subcomponent: Interpreter
Module:       ARIXI01 (acquire DBSPACE)
Trace level:  2
Type:         Report
Description:  Update SYSDBSPACES with information for
              acquired DBSPACE

Trace point:  5813
Subcomponent: Interpreter
Module:       ARIXI01 (acquire DBSPACE)
Trace level:  2
Type:         Report
Description:  Closing scan on SYSDBSPACES

Trace point:  5819
Subcomponent: Interpreter
Module:       ARIXI01 (acquire DBSPACE)
Trace level:  1
Type:         Module exit
Description:  Exit from module ARIXI01

Trace point:  5820
Subcomponent: Interpreter
Module:       ARIXI02 (begin work)
Trace level:  1
Type:         Module entry
Description:  Entry to module ARIXI02

Trace point:  5821
Subcomponent: Interpreter
Module:       ARIXI02 (begin work)
Trace level:  2
Type:         Report
Description:  Calling DBSS to begin work

Trace point:  5824
Subcomponent: Interpreter
Module:       ARIXI02 (begin work)
Trace level:  1
Type:         Module exit
Description:  Exit from module ARIXI02

Trace point:  5825
Subcomponent: Interpreter
Module:       ARIXI03 (change DBSPACE)
Trace level:  1
Type:         Module entry
Description:  Entry to module ARIXI03

Trace point:  5826
Subcomponent: Interpreter
Module:       ARIXI03 (change DBSPACE)
Trace level:  2
Type:         Report
Description:  Input parameters to change DBSPACE

Trace point:  5827
Subcomponent: Interpreter
Module:       ARIXI03 (change DBSPACE)
Trace level:  2
Type:         Report
Description:  Fetch for private DBSPACE from SYSDBSPACES

Trace point:  5828
Subcomponent: Interpreter
Module:       ARIXI03 (change DBSPACE)
Trace level:  2
Type:         Report
Description:  Change for public DBSPACE from SYSDBSPACES

Trace point:  5829
Subcomponent: Interpreter
Module:       ARIXI03 (change DBSPACE)
Trace level:  2
Type:         Report
Description:  Information in SYSDBSPACES before change

Trace point:  5830
Subcomponent: Interpreter
Module:       ARIXI03 (change DBSPACE)
Trace level:  2
Type:         Report
Description:  Fetching a DBSPACE control table
```

```
Trace point:  5831
Subcomponent: Interpreter
Module:       ARIXI03 (change DBSPACE)
Trace level:  2
Type:         Report
Description:  Updating DBSPACE control table

Trace point:  5832
Subcomponent: Interpreter
Module:       ARIXI03 (change DBSPACE)
Trace level:  2
Type:         Report
Description:  Update SYSDBSPACES

Trace point:  5833
Subcomponent: Interpreter
Module:       ARIXI03 (change DBSPACE)
Trace level:  2
Type:         Report
Description:  Unlock SYSDBSPACES

Trace point:  5837
Subcomponent: Interpreter
Module:       ARIXI03 (change DBSPACE)
Trace level:  2
Type:         Report
Description:  Return from module ARIXI03

Trace point:  5838
Subcomponent: Interpreter
Module:       ARIXI04 (insert comment)
Trace level:  1
Type:         Module entry
Description:  Entry to module ARIXI04

Trace point:  5839
Subcomponent: Interpreter
Module:       ARIXI04 (insert comment)
Trace level:  2
Type:         Report
Description:  Input parameters to ARIXI04

Trace point:  5840
Subcomponent: Interpreter
Module:       ARIXI04 (insert comment)
Trace level:  2
Type:         Report
Description:  Description of column for which comment is
              inserted

Trace point:  5841
Subcomponent: Interpreter
Module:       ARIXI04 (insert comment)
Trace level:  2
Type:         Report
Description:  Comment on table

Trace point:  5842
Subcomponent: Interpreter
Module:       ARIXI04 (insert comment)
Trace level:  2
Type:         Report
Description:  Fetch from SYSCOLUMNS or SYSCATALOG

Trace point:  5843
Subcomponent: Interpreter
Module:       ARIXI04 (insert comment)
Trace level:  2
Type:         Report
Description:  Update SYSCOLUMNS or SYSCATALOG

Trace point:  5846
Subcomponent: Interpreter
Module:       ARIXI04 (insert comment)
Trace level:  1
Type:         Module exit
Description:  Exit from module ARIXI04

Trace point:  5847
Subcomponent: Interpreter
Module:       ARIXI06 (create index)
Trace level:  1
Type:         Module entry
Description:  Entry to module ARIXI06

Trace point:  5848
Subcomponent: Interpreter
Module:       ARIXI06 (create index)
Trace level:  2
Type:         Report
Description:  Input parameters to module ARIXI06

Trace point:  5849
Subcomponent: Interpreter
Module:       ARIXI06 (create index)
Trace level:  2
Type:         Report
Description:  Input parameters to module ARIXI06
```

Trace point: 5850
Subcomponent: Interpreter
Module: ARIXI06 (create index)
Trace level: 2
Type: Report
Description: Number of columns in index

Trace point: 5851
Subcomponent: Interpreter
Module: ARIXI06 (create index)
Trace level: 2
Type: Report
Description: Fetch from SYSINDEXES to check if index with given name already defined

Trace point: 5852
Subcomponent: Interpreter
Module: ARIXI06 (create index)
Trace level: 2
Type: Report
Description: Fetch from SYSCATALOG to get DBSPACENO and TABID

Trace point: 5853
Subcomponent: Interpreter
Module: ARIXI06 (create index)
Trace level: 2
Type: Report
Description: Information fetched from SYSCATALOG

Trace point: 5854
Subcomponent: Interpreter
Module: ARIXI06 (create index)
Trace level: 2
Type: Report
Description: Information about column

Trace point: 5855
Subcomponent: Interpreter
Module: ARIXI06 (create index)
Trace level: 2
Type: Report
Description: Keysize information

Trace point: 5856
Subcomponent: Interpreter
Module: ARIXI06 (create index)
Trace level: 2
Type: Report
Description: CINSERT call to DBSS

Trace point: 5857
Subcomponent: Interpreter
Module: ARIXI06 (create index)
Trace level: 2
Type: Report
Description: Information about inserted index

Trace point: 5858
Subcomponent: Interpreter
Module: ARIXI06 (create index)
Trace level: 2
Type: Report
Description: Open INDEXLINK scan call to DBSS

Trace point: 5859
Subcomponent: Interpreter
Module: ARIXI06 (create index)
Trace level: 2
Type: Report
Description: Next on INDEXLINK scan call to DBSS

Trace point: 5860
Subcomponent: Interpreter
Module: ARIXI06 (create index)
Trace level: 2
Type: Report
Description: Next on INDEXLINK scan call to DBSS

Trace point: 5861
Subcomponent: Interpreter
Module: ARIXI06 (create index)
Trace level: 2
Type: Report
Description: Close index link scan call to DBSS

Trace point: 5862
Subcomponent: Interpreter
Module: ARIXI06 (create index)
Trace level: 2
Type: Report
Description: Inserting into SYSINDEXES for new index

Trace point: 5863
Subcomponent: Interpreter
Module: ARIXI06 (create index)
Trace level: 2
Type: Report
Description: Update SYSCATALOG call to DBSS

```
Trace point:  5864
Subcomponent: Interpreter
Module:       ARIXI06 (create index)
Trace level:  2
Type:         Report
Description:  Lock DBSPACE call to DBSS

Trace point:  5865
Subcomponent: Interpreter
Module:       ARIXI06 (create index)
Trace level:  2
Type:         Report
Description:  Start unload call to DBSS

Trace point:  5866
Subcomponent: Interpreter
Module:       ARIXI06 (create index)
Trace level:  2
Type:         Report
Description:  End unload call to DBSS

Trace point:  5867
Subcomponent: Interpreter
Module:       ARIXI06 (create index)
Trace level:  2
Type:         Report
Description:  Call ARIXI20 for update statistics

Trace point:  5868
Subcomponent: Interpreter
Module:       ARIXI06 (create index)
Trace level:  2
Type:         Report
Description:  Unlock DBSPACE call to DBSS

Trace point:  5869
Subcomponent: Interpreter
Module:       ARIXI06 (create index)
Trace level:  2
Type:         Report
Description:  Unlock SYSCATALOG call to DBSS

Trace point:  5875
Subcomponent: Interpreter
Module:       ARIXI06 (create index)
Trace level:  1
Type:         Module exit
Description:  Exit from module ARIXI06
```

```
Trace point:  5876
Subcomponent: Interpreter
Module:       ARIXI07 (create table)
Trace level:  1
Type:         Module entry
Description:  Entry to module ARIXI07

Trace point:  5877
Subcomponent: Interpreter
Module:       ARIXI07 (create table)
Trace level:  2
Type:         Report
Description:  Input parameters to module ARIXI07

Trace point:  5878
Subcomponent: Interpreter
Module:       ARIXI07 (create table)
Trace level:  2
Type:         Report
Description:  Information about columns

Trace point:  5879
Subcomponent: Interpreter
Module:       ARIXI07 (create table)
Trace level:  2
Type:         Report
Description:  Input information for table

Trace point:  5880
Subcomponent: Interpreter
Module:       ARIXI07 (create table)
Trace level:  2
Type:         Report
Description:  Fetch from SYSCATALOG to check if table name
              already exist for given creator

Trace point:  5881
Subcomponent: Interpreter
Module:       ARIXI07 (create table)
Trace level:  2
Type:         Report
Description:  Fetch from SYSSYNONYMS to check if table name
              used as synonym for given creator

Trace point:  5882
Subcomponent: Interpreter
Module:       ARIXI07 (create table)
Trace level:  2
Type:         Report
Description:  Creating table in user specified DBSPACE
```

```
Trace point:  5883
Subcomponent: Interpreter
Module:       ARIXI07 (create table)
Trace level:  2
Type:         Report
Description:  Fetch from SYSDBSPACES call to DBSS to look
              for specified DBSPACE

Trace point:  5884
Subcomponent: Interpreter
Module:       ARIXI07 (create table)
Trace level:  2
Type:         Report
Description:  Fetch DBSPACEs call to DBSS for public DBSPACE

Trace point:  5885
Subcomponent: Interpreter
Module:       ARIXI07 (create table)
Trace level:  2
Type:         Report
Description:  Creating table in not user specified DBSPACE
              fetch from DBSPACEs call to DBSS for private

Trace point:  5886
Subcomponent: Interpreter
Module:       ARIXI07 (create table)
Trace level:  2
Type:         Report
Description:  Name of found DBSPACE

Trace point:  5887
Subcomponent: Interpreter
Module:       ARIXI07 (create table)
Trace level:  2
Type:         Report
Description:  Authorization check if DBSPACE owner is not
              equal to AUTHID or PUBLIC.

Trace point:  5888
Subcomponent: Interpreter
Module:       ARIXI07 (create table)
Trace level:  2
Type:         Report
Description:  Column name

Trace point:  5889
Subcomponent: Interpreter
Module:       ARIXI07 (create table)
Trace level:  2
Type:         Report
Description:  Inserting master control row

Trace point:  5890
Subcomponent: Interpreter
Module:       ARIXI07 (create table)
Trace level:  2
Type:         Report
Description:  Master control row inserted

Trace point:  5891
Subcomponent: Interpreter
Module:       ARIXI07 (create table)
Trace level:  2
Type:         Report
Description:  Inserting companion master control row for
              long field

Trace point:  5892
Subcomponent: Interpreter
Module:       ARIXI07 (create table)
Trace level:  2
Type:         Report
Description:  Companion master control row inserted

Trace point:  5893
Subcomponent: Interpreter
Module:       ARIXI07 (create table)
Trace level:  2
Type:         Report
Description:  Creating unary link

Trace point:  5894
Subcomponent: Interpreter
Module:       ARIXI07 (create table)
Trace level:  2
Type:         Report
Description:  Unary link created

Trace point:  5895
Subcomponent: Interpreter
Module:       ARIXI07 (create table)
Trace level:  2
Type:         Report
Description:  Updating DBSPACE number in SYSDBSPACES

Trace point:  5896
Subcomponent: Interpreter
Module:       ARIXI07 (create table)
Trace level:  2
Type:         Report
Description:  Inserting row into SYSCATALOG for new table
```

Trace point: 5897
Subcomponent: Interpreter
Module: ARIXI07 (create table)
Trace level: 2
Type: Report
Description: Inserting information about column in SYSCOLUMNS

Trace point: 5898
Subcomponent: Interpreter
Module: ARIXI07 (create table)
Trace level: 2
Type: Report
Description: Grant authorization on created table to its creator

Trace point: 5899
Subcomponent: Interpreter
Module: ARIXI07 (create table)
Trace level: 2
Type: Report
Description: Unlock SYSDBSPACES

Trace point: 5905
Subcomponent: Interpreter
Module: ARIXI07 (create table)
Trace level: 1
Type: Module exit
Description: Exit from module ARIXI07

Trace point: 5906
Subcomponent: Interpreter
Module: ARIXI08 (create synonym)
Trace level: 1
Type: Module entry
Description: Entry to module ARIXI08

Trace point: 5907
Subcomponent: Interpreter
Module: ARIXI08 (create synonym)
Trace level: 2
Type: Report
Description: Input information for module ARIXI08

Trace point: 5908
Subcomponent: Interpreter
Module: ARIXI08 (create synonym)
Trace level: 2
Type: Report
Description: Input information for module ARIXI08

Trace point: 5910
Subcomponent: Interpreter
Module: ARIXI08 (create synonym)
Trace level: 2
Type: Report
Description: Fetch from SYSCATALOG to check if table with given name exists

Trace point: 5911
Subcomponent: Interpreter
Module: ARIXI08 (create synonym)
Trace level: 2
Type: Report
Description: Insert into SYSSYNONYMS entry for created synonym

Trace point: 5915
Subcomponent: Interpreter
Module: ARIXI08 (create synonym)
Trace level: 1
Type: Module exit
Description: Exit from module ARIXI08

Trace point: 5916
Subcomponent: Interpreter
Module: ARIXI09 (create view)
Trace level: 1
Type: Module entry
Description: Entry to module ARIXI09

Trace point: 5917
Subcomponent: Interpreter
Module: ARIXI09 (create view)
Trace level: 2
Type: Report
Description: Input parameters for module ARIXI09

Trace point: 5918
Subcomponent: Interpreter
Module: ARIXI09 (create view)
Trace level: 2
Type: Report
Description: Input information about view

Trace point: 5919
Subcomponent: Interpreter
Module: ARIXI09 (create view)
Trace level: 2
Type: Report
Description: Input information about view columns

Trace point: 5920
Subcomponent: Interpreter
Module: ARIXI09 (create view)
Trace level: 2
Type: Report
Description: Fetch from SYSCATALOG to check if view name already used as table or view name

Trace point: 5921
Subcomponent: Interpreter
Module: ARIXI09 (create view)
Trace level: 2
Type: Report
Description: Fetch from SYSSYNONYMS to check if view name already used as synonym name

Trace point: 5922
Subcomponent: Interpreter
Module: ARIXI09 (create view)
Trace level: 2
Type: Report
Description: Checking select authority

Trace point: 5923
Subcomponent: Interpreter
Module: ARIXI09 (create view)
Trace level: 2
Type: Report
Description: Checking delete authority

Trace point: 5924
Subcomponent: Interpreter
Module: ARIXI09 (create view)
Trace level: 2
Type: Report
Description: Checking insert authority

Trace point: 5925
Subcomponent: Interpreter
Module: ARIXI09 (create view)
Trace level: 2
Type: Report
Description: Granting at least one of select, insert, delete for view

Trace point: 5926
Subcomponent: Interpreter
Module: ARIXI09 (create view)
Trace level: 2
Type: Report
Description: Check update authority for view column

Trace point: 5927
Subcomponent: Interpreter
Module: ARIXI09 (create view)
Trace level: 2
Type: Report
Description: Grant update authority on view column

Trace point: 5928
Subcomponent: Interpreter
Module: ARIXI09 (create view)
Trace level: 2
Type: Report
Description: Grant total update on view columns

Trace point: 5929
Subcomponent: Interpreter
Module: ARIXI09 (create view)
Trace level: 2
Type: Report
Description: Grant select authority

Trace point: 5930
Subcomponent: Interpreter
Module: ARIXI09 (create view)
Trace level: 2
Type: Report
Description: Insert into SYSVIEWS

Trace point: 5931
Subcomponent: Interpreter
Module: ARIXI09 (create view)
Trace level: 2
Type: Report
Description: Insert into SYSCATALOG

Trace point: 5932
Subcomponent: Interpreter
Module: ARIXI09 (create view)
Trace level: 2
Type: Report
Description: Insert into SYSCOLUMNS

Trace point: 5933
Subcomponent: Interpreter
Module: ARIXI09 (create view)
Trace level: 2
Type: Report
Description: Insert into SYSUSAGE

Trace point: 5934
Subcomponent: Interpreter
Module: ARIXI09 (create view)
Trace level: 2
Type: Report
Description: Initialize AUX storing

Trace point: 5935
Subcomponent: Interpreter
Module: ARIXI09 (create view)
Trace level: 2
Type: Report
Description: Add section to AUX

Trace point: 5936
Subcomponent: Interpreter
Module: ARIXI09 (create view)
Trace level: 2
Type: Report
Description: Finish storing AUX

Trace point: 5937
Subcomponent: Interpreter
Module: ARIXI09 (create view)
Trace level: 2
Type: Report
Description: Unlock SYSCATALOG row

Trace point: 5944
Subcomponent: Interpreter
Module: ARIXI09 (create view)
Trace level: 1
Type: Module exit
Description: Exit from module ARIXI09

ARIXTR3 (Interpreter Trace Point Descriptor Module (part 2))

Trace point: 5945
Subcomponent: Interpreter
Module: ARIXI10 (drop)
Trace level: 1
Type: Module entry
Description: Entry to module ARIXI10

Trace point: 5946
Subcomponent: Interpreter
Module: ARIXI10 (drop)
Trace level: 2
Type: Report
Description: Input parameters to module ARIXI10

Trace point: 5947
Subcomponent: Interpreter
Module: ARIXI10 (drop)
Trace level: 1
Type: Report
Description: Entry to just drop AUX. Entry point ARIXI101 - Display entry parameters

Trace point: 5948
Subcomponent: Interpreter
Module: ARIXI10 (drop)
Trace level: 1
Type: Report
Description: Entry to drop AUX or view. Entry point ARIXI102. Display entry parameters

Trace point: 5954
Subcomponent: Interpreter
Module: ARIXI22 (drop index)
Trace level: 1
Type: Module entry
Description: Entry to module ARIXI22. Display input parameters.

Trace point: 5955
Subcomponent: Interpreter
Module: ARIXI22 (drop index)
Trace level: 2
Type: Report
Description: Fetch from SYSINDEXES

Trace point: 5956
Subcomponent: Interpreter
Module: ARIXI22 (drop index)
Trace level: 2
Type: Report
Description: Information from SYSINDEXES

Trace point: 5957
Subcomponent: Interpreter
Module: ARIXI22 (drop index)
Trace level: 2
Type: Report
Description: Delete from SYSINDEXES

Trace point: 5958
Subcomponent: Interpreter
Module: ARIXI22 (drop)
Trace level: 2
Type: Report
Description: Fetch from syscatalog

Trace point: 5959
Subcomponent: Interpreter
Module: ARIXI22 (drop index)
Trace level: 2
Type: Report
Description: Display information from SYSCATALOG

Trace point: 5960
Subcomponent: Interpreter
Module: ARIXI22 (drop index)
Trace level: 2
Type: Report
Description: Update CLUSTERTYPE and CLUSTERROW fields in SYSCATALOG

Trace point: 5961
Subcomponent: Interpreter
Module: ARIXI22 (drop)
Trace level: 1
Type: Module exit
Description: Exit from ARIXI22

Trace point: 5962
Subcomponent: Interpreter
Module: ARIXI23 (drop table)
Trace level: 1
Type: Module entry
Description: Entry to module ARIXI23

Trace point: 5963
Subcomponent: Interpreter
Module: ARIXI23 (drop table)
Trace level: 2
Type: Report
Description: Fetch from SYSCATALOG

Trace point: 5964
Subcomponent: Interpreter
Module: ARIXI23 (drop table)
Trace level: 2
Type: Report
Description: Display information from SYSCATALOG

Trace point: 5965
Subcomponent: Interpreter
Module: ARIXI23 (drop table)
Trace level: 2
Type: Report
Description: Fetch from SYSDBSPACES

Trace point: 5966
Subcomponent: Interpreter
Module: ARIXI23 (drop table)
Trace level: 2
Type: Report
Description: Display information from SYSDBSPACES and update SYSDBSPACES

Trace point: 5967
Subcomponent: Interpreter
Module: ARIXI23 (drop table)
Trace level: 1
Type: Module exit
Description: Exit from module ARIXI23

Trace point: 5968
Subcomponent: Interpreter
Module: ARIXI24 (drop view)
Trace level: 1
Type: Module entry
Description: Entry to module ARIXI24

Trace point: 5969
Subcomponent: Interpreter
Module: ARIXI24 (drop view)
Trace level: 2
Type: Report
Description: Fetch from SYSCATALOG

Trace point: 5970
Subcomponent: Interpreter
Module: ARIXI24 (drop view)
Trace level: 2
Type: Report
Description: Open scan on SYSVIEWS

Trace point: 5971
Subcomponent: Interpreter
Module: ARIXI24 (drop view)
Trace level: 2
Type: Report
Description: Close scan on SYSVIEWS

Trace point: 5972
Subcomponent: Interpreter
Module: ARIXI24 (drop view)
Trace level: 2
Type: Report
Description: Delete from SYSVIEWS

```
Trace point:  5973
Subcomponent: Interpreter
Module:       ARIXI24 (drop view)
Trace level:  2
Type:         Report
Description:  Next from SYSVIEWS

Trace point:  5974
Subcomponent: Interpreter
Module:       ARIXI24 (drop view)
Trace level:  2
Type:         Report
Description:  Fetch from SYSACCESS

Trace point:  5975
Subcomponent: Interpreter
Module:       ARIXI24 (drop view)
Trace level:  2
Type:         Report
Description:  Display information from SYSACCESS. Delete
              from SYSACCESS

Trace point:  5976
Subcomponent: Interpreter
Module:       ARIXI24 (drop view)
Trace level:  1
Type:         Module exit
Description:  Exit from procedure ARIXI24

Trace point:  5977
Subcomponent: Interpreter
Module:       ARIXI25 (drop DBSPACE)
Trace level:  1
Type:         Module entry
Description:  Entry to module ARIXI25

Trace point:  5978
Subcomponent: Interpreter
Module:       ARIXI25 (drop DBSPACE)
Trace level:  2
Type:         Report
Description:  Fetch from SYSDBSPACES

Trace point:  5979
Subcomponent: Interpreter
Module:       ARIXI25 (drop DBSPACE)
Trace level:  2
Type:         Report
Description:  Fetch for public.  Owner from SYSDBSPACES

Trace point:  5980
Subcomponent: Interpreter
Module:       ARIXI25 (drop DBSPACE)
Trace level:  2
Type:         Report
Description:  Found public DBSPACE.  Check DBA authority of
              PREPper.

Trace point:  5981
Subcomponent: Interpreter
Module:       ARIXI25 (drop DBSPACE)
Trace level:  2
Type:         Report
Description:  Display information from SYSDBSPACES. Open
              relational scan for DBSPACE that will be
              dropped.

Trace point:  5982
Subcomponent: Interpreter
Module:       ARIXI25 (drop DBSPACE)
Trace level:  2
Type:         Report
Description:  Information about table to be dropped in
              dropped DBSPACE

Trace point:  5983
Subcomponent: Interpreter
Module:       ARIXI25 (drop DBSPACE)
Trace level:  2
Type:         Report
Description:  Next from SYSCATALOG

Trace point:  5984
Subcomponent: Interpreter
Module:       ARIXI25 (drop DBSPACE)
Trace level:  2
Type:         Report
Description:  Close scan on SYSCATALOG

Trace point:  5985
Subcomponent: Interpreter
Module:       ARIXI25 (drop DBSPACE)
Trace level:  2
Type:         Report
Description:  Update SYSDBSPACES

Trace point:  5987
Subcomponent: Interpreter
Module:       ARIXI25 (drop DBSPACE)
Trace level:  1
Type:         Module exit
Description:  Exit from module ARIXI25
```

```
Trace point:  5988
Subcomponent: Interpreter
Module:       ARIXI26 (drop synonym)
Trace level:  1
Type:         Module entry
Description:  Entry to module ARIXI26

Trace point:  5989
Subcomponent: Interpreter
Module:       ARIXI26 (drop synonym)
Trace level:  2
Type:         Report
Description:  Delete from SYSSYNONYMS

Trace point:  5990
Subcomponent: Interpreter
Module:       ARIXI26 (drop synonym)
Trace level:  1
Type:         Module exit
Description:  Exit from module ARIXI26

Trace point:  5991
Subcomponent: Interpreter
Module:       ARIXI27
Trace level:  1
Type:         Module entry
Description:  Entry to module ARIXI27

Trace point:  5992
Subcomponent: Interpreter
Module:       ARIXI27
Trace level:  2
Type:         Report
Description:  Doing CDELETE

Trace point:  5993
Subcomponent: Interpreter
Module:       ARIXI27
Trace level:  2
Type:         Report
Description:  Insert into SYSDROP

Trace point:  5994
Subcomponent: Interpreter
Module:       ARIXI27
Trace level:  1
Type:         Module exit
Description:  Exit from module ARIXI27

Trace point:  5995
Subcomponent: Interpreter
Module:       ARIXI28
Trace level:  1
Type:         Module entry
Description:  Entry to module ARIXI28. (delete rows from
              child catalog)

Trace point:  5996
Subcomponent: Interpreter
Module:       ARIXI28
Trace level:  2
Type:         Report
Description:  Open link scan from SYSCATALOG.

Trace point:  5997
Subcomponent: Interpreter
Module:       ARIXI28
Trace level:  2
Type:         Module
Description:  Next from child catalog

Trace point:  5998
Subcomponent: Interpreter
Module:       ARIXI28
Trace level:  2
Type:         Report
Description:  Close scan on child catalog

Trace point:  5999
Subcomponent: Interpreter
Module:       ARIXI28
Trace level:  2
Type:         Report
Description:  Delete from child catalog

Trace point:  6000
Subcomponent: Interpreter
Module:       ARIXI28
Trace level:  1
Type:         Module exit
Description:  Exit from module ARIXI28

Trace point:  6001
Subcomponent: Interpreter
Module:       ARIXI29
Trace level:  1
Type:         Module entry
Description:  Entry to module ARIXI29 (drop dependencies)
```

```
Trace point:  6002
Subcomponent: Interpreter
Module:       ARIXI29
Trace level:  2
Type:         Report
Description:  Open scan on SYSUSAGE

Trace point:  6003
Subcomponent: Interpreter
Module:       ARIXI29
Trace level:  2
Type:         Report
Description:  Close scan on SYSUSAGE

Trace point:  6004
Subcomponent: Interpreter
Module:       ARIXI29
Trace level:  2
Type:         Report
Description:  Delete from SYSUSAGE

Trace point:  6005
Subcomponent: Interpreter
Module:       ARIXI29
Trace level:  2
Type:         Report
Description:  Display information about dependent object.

Trace point:  6006
Subcomponent: Interpreter
Module:       ARIXI29
Trace level:  2
Type:         Report
Description:  Next from SYSUSAGE

Trace point:  6007
Subcomponent: Interpreter
Module:       ARIXI29
Trace level:  1
Type:         Module exit
Description:  Exit from module ARIXI29

Trace point:  6008
Subcomponent: Interpreter
Module:       ARIXI30 (drop AUX)
Trace level:  1
Type:         Module entry
Description:  Entering module ARIXI30

Trace point:  6009
Subcomponent: Interpreter
Module:       ARIXI30 (drop AUX)
Trace level:  2
Type:         Report
Description:  Fetch from SYSACCESS

Trace point:  6010
Subcomponent: Interpreter
Module:       ARIXI30 (drop AUX)
Trace level:  2
Type:         Report
Description:  Invalidate AUX by updating valid field in
              SYSACCESS

Trace point:  6012
Subcomponent: Interpreter
Module:       ARIXI30 (drop AUX)
Trace level:  2
Type:         Report
Description:  Delete from SYSACCESS

Trace point:  6012
Subcomponent: Interpreter
Module:       ARIXI30 (drop AUX)
Trace level:  1
Type:         Module exit
Description:  Exit from module ARIXI30

Trace point:  6013
Subcomponent: Interpreter
Module:       ARIXI31
Trace level:  1
Type:         Module entry
Description:  Entry to module ARIXI31

Trace point:  6014
Subcomponent: Interpreter
Module:       ARIXI31
Trace level:  2
Type:         Report
Description:  Open scan on SYSUSAGE

Trace point:  6015
Subcomponent: Interpreter
Module:       ARIXI31
Trace level:  2
Type:         Report
Description:  Close scan on SYSUSAGE
```

```
Trace point:  6016
Subcomponent: Interpreter
Module:       ARIXI31
Trace level:  2
Type:         Report
Description:  Delete from SYSUSAGE

Trace point:  6017
Subcomponent: Interpreter
Module:       ARIXI31
Trace level:  2
Type:         Report
Description:  Next from SYSUSAGE

Trace point:  6018
Subcomponent: Interpreter
Module:       ARIXI31
Trace level:  1
Type:         Module exit
Description:  Exit from module ARIXI31

Trace point:  6019
Subcomponent: Interpreter
Module:       ARIXI32
Trace level:  1
Type:         Module entry
Description:  Entry to module ARIXI32

Trace point:  6020
Subcomponent: Interpreter
Module:       ARIXI32
Trace level:  2
Type:         Report
Description:  Delete from SYSCATALOG

Trace point:  6021
Subcomponent: Interpreter
Module:       ARIXI32
Trace level:  1
Type:         Module exit
Description:  Exit from module ARIXI32

Trace point:  6025
Subcomponent: Interpreter
Module:       ARIXI10 (drop)
Trace level:  1
Type:         Module exit
Description:  Exit from module ARIXI10

Trace point:  6027
Subcomponent: Interpreter
Module:       ARIXI10 (drop)
Trace level:  1
Type:         Module exit
Description:  Exit from module ARIXI10 (ARIXI101)

Trace point:  6029
Subcomponent: Interpreter
Module:       ARIXI10 (drop)
Trace level:  1
Type:         Module exit
Description:  Exit from module ARIXI10 (ARIXI102)

Trace point:  6030
Subcomponent: Interpreter
Module:       ARIXI12 (commit work)
Trace level:  1
Type:         Module entry
Description:  Entry to module ARIXI12

Trace point:  6031
Subcomponent: Interpreter
Module:       ARIXI12 (commit work)
Trace level:  2
Type:         Report
Description:  Issue COMMIT WORK call to DBSS

Trace point:  6033
Subcomponent: Interpreter
Module:       ARIXI12 (commit work)
Trace level:  1
Type:         Module exit
Description:  Exit from module ARIXI12

Trace point:  6034
Subcomponent: Interpreter
Module:       ARIXI13 (alter table)
Trace level:  1
Type:         Module entry
Description:  Entry to module ARIXI13

Trace point:  6035
Subcomponent: Interpreter
Module:       ARIXI13 (alter table)
Trace level:  2
Type:         Report
Description:  Display input parameters
```

```
Trace point:  6036
Subcomponent: Interpreter
Module:       ARIXI13 (alter table)
Trace level:  2
Type:         Report
Description:  Information about new column

Trace point:  6037
Subcomponent: Interpreter
Module:       ARIXI13 (alter table)
Trace level:  2
Type:         Report
Description:  Fetch from SYSCATALOG to find table for alter

Trace point:  6038
Subcomponent: Interpreter
Module:       ARIXI13 (alter table)
Trace level:  2
Type:         Report
Description:  Display information from SYSCATALOG and do a
              CFETCH for a table.

Trace point:  6039
Subcomponent: Interpreter
Module:       ARIXI13 (alter table)
Trace level:  2
Type:         Report
Description:  Do CUPDATE for master control row

Trace point:  6040
Subcomponent: Interpreter
Module:       ARIXI13 (alter table)
Trace level:  2
Type:         Report
Description:  CINSERT companion master control row

Trace point:  6041
Subcomponent: Interpreter
Module:       ARIXI13 (alter table)
Trace level:  2
Type:         Report
Description:  Creating unary link

Trace point:  6042
Subcomponent: Interpreter
Module:       ARIXI13 (alter table)
Trace level:  2
Type:         Report
Description:  Update SYSCATALOG

Trace point:  6043
Subcomponent: Interpreter
Module:       ARIXI13 (alter table)
Trace level:  2
Type:         Report
Description:  Open column link scan on SYSCATALOG

Trace point:  6044
Subcomponent: Interpreter
Module:       ARIXI13 (alter table)
Trace level:  2
Type:         Report
Description:  Next on column link scan

Trace point:  6045
Subcomponent: Interpreter
Module:       ARIXI13 (alter table)
Trace level:  2
Type:         Module
Description:  Insert into SYSCOLUMNS

Trace point:  6046
Subcomponent: Interpreter
Module:       ARIXI13 (alter table)
Trace level:  2
Type:         Report
Description:  Close column link scan

Trace point:  6047
Subcomponent: Interpreter
Module:       ARIXI13 (alter table)
Trace level:  2
Type:         Report
Description:  Grant update on new column

Trace point:  6048
Subcomponent: Interpreter
Module:       ARIXI13 (alter table)
Trace level:  2
Type:         Report
Description:  Unlock SYSCATALOG row

Trace point:  6052
Subcomponent: Interpreter
Module:       ARIXI13 (alter table)
Trace level:  1
Type:         Module exit
Description:  Exit from module ARIXI13
```

```
Trace point:  6053
Subcomponent: Interpreter
Module:       ARIXI15 (lock)
Trace level:  1
Type:         Module entry
Description:  Entry to module ARIXI15

Trace point:  6054
Subcomponent: Interpreter
Module:       ARIXI15 (lock)
Trace level:  2
Type:         Report
Description:  Display input parameters to ARIXI15

Trace point:  6055
Subcomponent: Interpreter
Module:       ARIXI15 (lock)
Trace level:  2
Type:         Report
Description:  Lock DBSPACE. Fetch for OWNER=CREATOR from
              SYSDBSPACES.

Trace point:  6056
Subcomponent: Interpreter
Module:       ARIXI15 (lock)
Trace level:  2
Type:         Report
Description:  Fetch for OWNER = PUBLIC from SYSDBSPACES

Trace point:  6057
Subcomponent: Interpreter
Module:       ARIXI15 (lock)
Trace level:  2
Type:         Report
Description:  Will lock table. Fetch from SYSCATALOG

Trace point:  6058
Subcomponent: Interpreter
Module:       ARIXI15 (lock)
Trace level:  2
Type:         Report
Description:  Lock call to DBSS

Trace point:  6061
Subcomponent: Interpreter
Module:       ARIXI15 (lock)
Trace level:  1
Type:         Module exit
Description:  Exit from module ARIXI15

Trace point:  6062
Subcomponent: Interpreter
Module:       ARIXI16 (rollback work)
Trace level:  1
Type:         Module entry
Description:  Entry to module ARIXI16

Trace point:  6063
Subcomponent: Interpreter
Module:       ARIXI16 (rollback work)
Trace level:  2
Type:         Report
Description:  Doing ROLLBACK WORK call to DBSS

Trace point:  6065
Subcomponent: Interpreter
Module:       ARIXI16 (rollback work)
Trace level:  1
Type:         Module exit
Description:  Exit from module ARIXI16

Trace point:  6066
Subcomponent: Interpreter
Module:       ARIXI18 (clean SYSDROP)
Trace level:  1
Type:         Module entry
Description:  Entry to module ARIXI18

Trace point:  6067
Subcomponent: Interpreter
Module:       ARIXI18 (clean SYSDROP)
Trace level:  2
Type:         Report
Description:  Begin work

Trace point:  6068
Subcomponent: Interpreter
Module:       ARIXI18 (clean SYSDROP)
Trace level:  2
Type:         Report
Description:  Next from SYSDROP

Trace point:  6069
Subcomponent: Interpreter
Module:       ARIXI18 (clean SYSDROP)
Trace level:  2
Type:         Report
Description:  Open scan on SYSDROP
```

```
Trace point:  6070
Subcomponent: Interpreter
Module:       ARIXI18 (clean SYSDROP)
Trace level:  2
Type:         Report
Description:  SYSDROP is clean

Trace point:  6071
Subcomponent: Interpreter
Module:       ARIXI18 (clean SYSDROP)
Trace level:  2
Type:         Report
Description:  Delete from SYSDROP

Trace point:  6072
Subcomponent: Interpreter
Module:       ARIXI18 (clean SYSDROP)
Trace level:  2
Type:         Report
Description:  Update SYSDBSPACES

Trace point:  6073
Subcomponent: Interpreter
Module:       ARIXI18 (clean SYSDROP)
Trace level:  2
Type:         Report
Description:  Do CDELETE

Trace point:  6074
Subcomponent: Interpreter
Module:       ARIXI18 (clean SYSDROP)
Trace level:  2
Type:         Report
Description:  Begin work after forced commit work

Trace point:  6075
Subcomponent: Interpreter
Module:       ARIXI18 (clean SYSDROP)
Trace level:  2
Type:         Report
Description:  Commit work

Trace point:  6078
Subcomponent: Interpreter
Module:       ARIXI18 (clean SYSDROP)
Trace level:  1
Type:         Module exit
Description:  Exit from module ARIXI18
```

```
Trace point:  6079
Subcomponent: Interpreter
Module:       ARIXI19 (top module for undate statistics)
Trace level:  1
Type:         Module entry
Description:  Entry to module ARIXI19

Trace point:  6080
Subcomponent: Interpreter
Module:       ARIXI19 (top module for update statistics)
Trace level:  2
Type:         Report
Description:  Lock DBSPACE

Trace point:  6081
Subcomponent: Interpreter
Module:       ARIXI19 (top module for update statistics)
Trace level:  2
Type:         Report
Description:  Start update statistics

Trace point:  6082
Subcomponent: Interpreter
Module:       ARIXI19 (top module for update statistics)
Trace level:  2
Type:         Report
Description:  End update statistics

Trace point:  6083
Subcomponent: Interpreter
Module:       ARIXI19 (top module for update statistics)
Trace level:  2
Type:         Report
Description:  Fetch from SYSINDEXES

Trace point:  6084
Subcomponent: Interpreter
Module:       ARIXI19 (top module for update statistics)
Trace level:  2
Type:         Report
Description:  Call ARIXI20 to update catalogs

Trace point:  6085
Subcomponent: Interpreter
Module:       ARIXI19 (top module for update statistics)
Trace level:  2
Type:         Report
Description:  Unlock DBSPACE
```

```
Trace point:  6088
Subcomponent: Interpreter
Module:       ARIXI19 (top module for update statistics)
Trace level:  1
Type:         Module exit
Description:  Exit from module ARIXI19

Trace point:  6089
Subcomponent: Interpreter
Module:       ARIXI20 (update statistics)
Trace level:  1
Type:         Module entry
Description:  Entry to module ARIXI20

Trace point:  6090
Subcomponent: Interpreter
Module:       ARIXI20 (update statistics)
Trace level:  2
Type:         Report
Description:  Update SYSCATALOG

Trace point:  6091
Subcomponent: Interpreter
Module:       ARIXI20 (update statistics)
Trace level:  2
Type:         Report
Description:  Update SYSDBSPACES

Trace point:  6092
Subcomponent: Interpreter
Module:       ARIXI20 (update statistics)
Trace level:  2
Type:         Report
Description:  Update SYSINDEXES

Trace point:  6093
Subcomponent: Interpreter
Module:       ARIXI20 (update statistics)
Trace level:  2
Type:         Report
Description:  Open scan on SYSCOLUMNS

Trace point:  6094
Subcomponent: Interpreter
Module:       ARIXI20 (update statistics)
Trace level:  2
Type:         Report
Description:  Next from SYSCOLUMNS

Trace point:  6095
Subcomponent: Interpreter
Module:       ARIXI20 (update statistics)
Trace level:  2
Type:         Report
Description:  Update SYSCOLUMNS

Trace point:  6096
Subcomponent: Interpreter
Module:       ARIXI20 (update statistics)
Trace level:  2
Type:         Report
Description:  Close scan on SYSCOLUMNS

Trace point:  6099
Subcomponent: Interpreter
Module:       ARIXI20 (update statistics)
Trace level:  1
Type:         Module exit
Description:  Exit from ARIXI20
```

ARIXTR4 (RDS Authorization and Security/Audit Trace Point Descriptor Module)

```
Trace point:  6150
Subcomponent: Authorization
Module:       ARIXA01 (SCHEDULE/CONNECT)
Trace level:  1
Type:         Module entry
Description:  Entry to module ARIXA01

Trace point:  6151
Subcomponent: Authorization
Module:       ARIXA01 (SCHEDULE/CONNECT)
Trace level:  2
Type:         Report
Description:  BEGIN WORK call to DBSS

Trace point:  6152
Subcomponent: Authorization
Module:       ARIXA01 (SCHEDULE/CONNECT)
Trace level:  2
Type:         Report
Description:  Fetch from SYSUSERAUTH

Trace point:  6153
Subcomponent: Authorization
Module:       ARIXA01 (SCHEDULE/CONNECT)
Trace level:  2
Type:         Report
Description:  COMMIT WORK call to DBSS
```

```
Trace point:  6154
Subcomponent: Authorization
Module:       ARIXA01 (SCHEDULE/CONNECT)
Trace level:  2
Type:         Report
Description:  COMMIT WORK call to DBSS

Trace point:  6155
Subcomponent: Authorization
Module:       ARIXA01 (SCHEDULE/CONNECT)
Trace level:  2
Type:         Report
Description:  Attempt to SCHEDULE/CONNECT user using invalid
              password

Trace point:  6156
Subcomponent: Authorization
Module:       ARIXA01 (SCHEDULE/CONNECT)
Trace level:  2
Type:         Report
Description:  SCHEDULE call to DBSS

Trace point:  6157
Subcomponent: Authorization
Module:       ARIXA02 (SCHEDULE/CONNECT)
Trace level:  2
Type:         Report
Description:  COMMIT WORK call to DBSS

Trace point:  6159
Subcomponent: Authorization
Module:       ARIXA01 (SCHEDULE/CONNECT)
Trace level:  1
Type:         Module exit
Description:  Exit from module ARIXA01

Trace point:  6160
Subcomponent: Authorization
Module:       ARIXA02 (check specific authority)
Trace level:  1
Type:         Report
Description:  Input to module ARIXA01. Display CHKSTRUCTURE.

Trace point:  6161
Subcomponent: Authorization
Module:       ARIXA02 (check specific authority)
Trace level:  2
Type:         Report
Description:  Display output areas of CHK-STRUCTURE

Trace point:  6162
Subcomponent: Authorization
Module:       ARIXA02 (check specific authority)
Trace level:  2
Type:         Report
Description:  Entry to procedure RUNCHEK (check RUNAUTH)

Trace point:  6163
Subcomponent: Authorization
Module:       ARIXA02 (check specific authority)
Trace level:  2
Type:         Report
Description:  In procedure RUNCHEK. Fetch from SYSPROGAUTH

Trace point:  6164
Subcomponent: Authorization
Module:       ARIXA02 (check specific authority)
Trace level:  2
Type:         Report
Description:  Exit from procedure RUNCHEK

Trace point:  6165
Subcomponent: Authorization
Module:       ARIXA02 (check specific authority)
Trace level:  2
Type:         Report
Description:  Entry to procedure USERCHEK (check user
              authority)

Trace point:  6166
Subcomponent: Authorization
Module:       ARIXA02 (check specific authority)
Trace level:  2
Type:         Report
Description:  In procedure USERCHEK. Fetch from SYSUSERAUTH.

Trace point:  6167
Subcomponent: Authorization
Module:       ARIXA02 (check specific authority)
Trace level:  2
Type:         Report
Description:  Exit from procedure USERCHEK

Trace point:  6168
Subcomponent: Authorization
Module:       ARIXA02 (check specific authority)
Trace level:  2
Type:         Report
Description:  Entry to procedure TABCHECK
```

```
Trace point:  6169
Subcomponent: Authorization
Module:       ARIXA02 (check specific authority)
Trace level:  2
Type:         Report
Description:  In procedure TABCHECK. Open scan on
              SYSTABAUTH.

Trace point:  6170
Subcomponent: Authorization
Module:       ARIXA02 (check specific authority)
Trace level:  2
Type:         Report
Description:  In procedure TABCHECK. Next from SYSTABAUTH

Trace point:  6171
Subcomponent: Authorization
Module:       ARIXA02 (check specific authority)
Trace level:  2
Type:         Report
Description:  In procedure TABCHECK. Next from SYSTABAUTH

Trace point:  6172
Subcomponent: Authorization
Module:       ARIXA02 (check specific authority)
Trace level:  2
Type:         Report
Description:  In procedure TABCHECK. Next from SYSTABAUTH

Trace point:  6173
Subcomponent: Authorization
Module:       ARIXA02 (check specific authority)
Trace level:  2
Type:         Report
Description:  In procedure TABCHECK. Close scan on
              SYSTABAUTH

Trace point:  6174
Subcomponent: Authorization
Module:       ARIXA02 (check specific authority)
Trace level:  2
Type:         Report
Description:  In procedure TABCHECK. Fetch from SYSCOLAUTH

Trace point:  6175
Subcomponent: Authorization
Module:       ARIXA02 (check specific authority)
Trace level:  2
Type:         Report
Description:  In procedure TABCHECK. Close scan on
              SYSTABAUTH.

Trace point:  6176
Subcomponent: Authorization
Module:       ARIXA02 (check specific authority)
Trace level:  2
Type:         Report
Description:  Exit from procedure TABCHECK

Trace point:  6179
Subcomponent: Authorization
Module:       ARIXA02 (check specific authority)
Trace level:  1
Type:         Module exit
Description:  Exit from module ARIXA02

Trace point:  6180
Subcomponent: Authorization
Module:       ARIXA03 (delete columns from SYSCOLAUTH)
Trace level:  1
Type:         Module entry
Description:  Entry to module ARIXI03

Trace point:  6181
Subcomponent: Authorization
Module:       ARIXA03 (delete columns from SYSCOLAUTH)
Trace level:  2
Type:         Report
Description:  Open scan on SYSCOLAUTH

Trace point:  6182
Subcomponent: Authorization
Module:       ARIXA03 (delete columns from SYSCOLAUTH)
Trace level:  2
Type:         Report
Description:  Delete from SYSCOLAUTH

Trace point:  6183
Subcomponent: Authorization
Module:       ARIXA03 (delete columns from SYSCOLAUTH)
Trace level:  2
Type:         Report
Description:  Next from SYSCOLAUTH

Trace point:  6184
Subcomponent: Authorization
Module:       ARIXA03 (delete columns from SYSCOLAUTH)
Trace level:  2
Type:         Report
Description:  Close scan on SYSCOLAUTH
```

Trace point: 6187
Subcomponent: Authorization
Module: ARIXA03 (delete columns from SYSCOLAUTH)
Trace level: 1
Type: Module exit
Description: Exit from module ARIXA03

Trace point: 6188
Subcomponent: Authorization
Module: ARIXA04 (delete rows from authorization catalogs)
Trace level: 1
Type: Module entry
Description: Entry to module ARIXA04

Trace point: 6189
Subcomponent: Authorization
Module: ARIXA04 (delete rows from authorization catalogs)
Trace level: 2
Type: Report
Description: Open scan on SYSPROGAUTH

Trace point: 6190
Subcomponent: Authorization
Module: ARIXA04 (delete rows from authorization catalogs)
Trace level: 2
Type: Report
Description: Delete from SYSPROGAUTH

Trace point: 6191
Subcomponent: Authorization
Module: ARIXA04 (delete rows from authorization catalogs)
Trace level: 2
Type: Report
Description: Next from SYSPROGAUTH

Trace point: 6192
Subcomponent: Authorization
Module: ARIXA04 (delete rows from authorization catalogs)
Trace level: 2
Type: Report
Description: Close scan on SYSPROGAUTH

Trace point: 6193
Subcomponent: Authorization
Module: ARIXA04 (delete rows from authorization catalogs)
Trace level: 2
Type: Report
Description: Open scan on SYSTABAUTH

Trace point: 6194
Subcomponent: Authorization
Module: ARIXA04 (delete rows from authorization catalogs)
Trace level: 2
Type: Report
Description: Next from SYSTABAUTH

Trace point: 6195
Subcomponent: Authorization
Module: ARIXA04 (delete rows from authorization catalogs)
Trace level: 2
Type: Report
Description: Delete from SYSTABAUTH

Trace point: 6196
Subcomponent: Authorization
Module: ARIXA04 (delete rows from authorization catalogs)
Trace level: 2
Type: Report
Description: Next from SYSTABAUTH

Trace point: 6197
Subcomponent: Authorization
Module: ARIXA04 (delete rows from authorization catalogs)
Trace level: 2
Type: Report
Description: Close scan on SYSTABAUTH

Trace point: 6198
Subcomponent: Authorization
Module: ARIXA04 (delete rows from authorization catalogs)
Trace level: 2
Type: Report
Description: Open scan on SYSUSERAUTH

```
Trace point:  6199
Subcomponent: Authorization
Module:       ARIXA04 (delete rows from authorization
              catalogs)
Trace level:  2
Type:         Report
Description:  Delete from SYSUSERAUTH

Trace point:  6200
Subcomponent: Authorization
Module:       ARIXA04 (delete rows from authorization
              catalogs)
Trace level:  2
Type:         Report
Description:  Next from SYSUSERAUTH

Trace point:  6201
Subcomponent: Authorization
Module:       ARIXA04 (delete rows from authorization
              catalogs)
Trace level:  2
Type:         Report
Description:  Close scan on SYSUSERAUTH

Trace point:  6202
Subcomponent: Authorization
Module:       ARIXA04 (delete rows from authorization
              catalogs)
Trace level:  2
Type:         Report
Description:  Open scan on SYSTABAUTH

Trace point:  6203
Subcomponent: Authorization
Module:       ARIXA04 (delete rows from authorization
              catalogs)
Trace level:  2
Type:         Report
Description:  Delete from SYSTABAUTH

Trace point:  6204
Subcomponent: Authorization
Module:       ARIXA04 (delete rows from authorization
              catalogs)
Trace level:  2
Type:         Report
Description:  Next from SYSTABAUTH

Trace point:  6205
Subcomponent: Authorization
Module:       ARIXA04 (delete rows from authorization
              catalogs)
Trace level:  2
Type:         Report
Description:  Close scan on SYSTABAUTH

Trace point:  6206
Subcomponent: Authorization
Module:       ARIXA04 (delete rows from authorization
              catalogs)
Trace level:  2
Type:         Report
Description:  Open scan on SYSTABAUTH

Trace point:  6207
Subcomponent: Authorization
Module:       ARIXA04 (delete rows from authorization
              catalogs)
Trace level:  2
Type:         Report
Description:  Delete from SYSTABAUTH

Trace point:  6208
Subcomponent: Authorization
Module:       ARIXA04 (delete rows from authorization
              catalogs)
Trace level:  2
Type:         Report
Description:  Next from SYSTABAUTH

Trace point:  6209
Subcomponent: Authorization
Module:       ARIXA04 (delete rows from authorization
              catalogs)
Trace level:  2
Type:         Report
Description:  Close scan on SYSTABAUTH

Trace point:  6213
Subcomponent: Authorization
Module:       ARIXA04 (delete rows from authorization
              catalogs)
Trace level:  1
Type:         Module exit
Description:  Exit from module ARIXA04
```

Trace point: 6214
Subcomponent: Authorization
Module: ARIXA06 (grant authority)
Trace level: 1
Type: Module entry
Description: Entry to module ARIXA06

Trace point: 6215
Subcomponent: Authorization
Module: ARIXA06 (grant authority)
Trace level: 2
Type: Report
Description: Open scan on SYSPROGAUTH

Trace point: 6216
Subcomponent: Authorization
Module: ARIXA06 (grant authority)
Trace level: 2
Type: Report
Description: Delete from SYSPROGAUTH

Trace point: 6217
Subcomponent: Authorization
Module: ARIXA06 (grant authority)
Trace level: 2
Type: Report
Description: Next from SYSPROGAUTH

Trace point: 6218
Subcomponent: Authorization
Module: ARIXA06 (grant authority)
Trace level: 2
Type: Report
Description: Close scan on SYSPROGAUTH

Trace point: 6219
Subcomponent: Authorization
Module: ARIXA06 (grant authority)
Trace level: 2
Type: Report
Description: Insert into SYSPROGAUTH

Trace point: 6220
Subcomponent: Authorization
Module: ARIXA06 (grant authority)
Trace level: 2
Type: Report
Description: Fetch from SYSUSERAUTH

Trace point: 6221
Subcomponent: Authorization
Module: ARIXA06 (grant authority)
Trace level: 2
Type: Report
Description: Update SYSUSERAUTH

Trace point: 6222
Subcomponent: Authorization
Module: ARIXA06 (grant authority)
Trace level: 2
Type: Report
Description: Insert into SYSUSERAUTH

Trace point: 6223
Subcomponent: Authorization
Module: ARIXA06 (grant authority)
Trace level: 2
Type: Report
Description: Fetch from SYSTABAUTH

Trace point: 6224
Subcomponent: Authorization
Module: ARIXA06 (grant authority)
Trace level: 2
Type: Report
Description: UPDATE SYSTABAUTH

Trace point: 6225
Subcomponent: Authorization
Module: ARIXA06 (grant authority)
Trace level: 2
Type: Report
Description: Insert into SYSTABAUTH

Trace point: 6226
Subcomponent: Authorization
Module: ARIXA06 (grant authority)
Trace level: 2
Type: Report
Description: Fetch from SYSCOLAUTH

Trace point: 6227
Subcomponent: Authorization
Module: ARIXA06 (grant authority)
Trace level: 2
Type: Report
Description: Insert into SYSCOLAUTH

Trace point: 6231
Subcomponent: Authorization
Module: ARIXA06 (grant authority)
Trace level: 1
Type: Module exit
Description: Exit from module ARIXA06

Trace point: 6232
Subcomponent: Authorization
Module: ARIXA07 (grant privilege)
Trace level: 1
Type: Module entry
Description: Entry to module ARIXA07

Trace point: 6233
Subcomponent: Authorization
Module: ARIXA07 (grant privilege)
Trace level: 2
Type: Report
Description: In RUNGRANT. About to grant RUNAUTH.

Trace point: 6234
Subcomponent: Authorization
Module: ARIXA07 (grant privilege)
Trace level: 2
Type: Report
Description: Fetch from SYSACCESS

Trace point: 6235
Subcomponent: Authorization
Module: ARIXA07 (grant privilege)
Trace level: 2
Type: Report
Description: In USERGRNT. About to grant special privilege.

Trace point: 6236
Subcomponent: Authorization
Module: ARIXA07 (grant privilege)
Trace level: 2
Type: Report
Description: In TABGRANT. About to grant privilege on table.

Trace point: 6237
Subcomponent: Authorization
Module: ARIXA07 (grant privilege)
Trace level: 2
Type: Report
Description: Fetch from SYSCATALOG

Trace point: 6238
Subcomponent: Authorization
Module: ARIXA07 (grant privilege)
Trace level: 2
Type: Report
Description: Fetch from SYSCOLUMNS

Trace point: 6239
Subcomponent: Authorization
Module: ARIXA07 (grant privilege)
Trace level: 2
Type: Report
Description: Entry to procedure DOGRANTS

Trace point: 6240
Subcomponent: Authorization
Module: ARIXA07 (grant privilege)
Trace level: 2
Type: Report
Description: Exit from procedure DOGRANTS

Trace point: 6250
Subcomponent: Authorization
Module: ARIXA07 (grant privilege)
Trace level: 1
Type: Module exit
Description: Exit from module ARIXA07

Trace point: 6251
Subcomponent: Authorization
Module: ARIXA08 (revoke privilege)
Trace level: 1
Type: Module entry
Description: Entry to module ARIXA08

Trace point: 6252
Subcomponent: Authorization
Module: ARIXA08 (revoke privilege)
Trace level: 2
Type: Report
Description: In CASE_0

Trace point: 6253
Subcomponent: Authorization
Module: ARIXA08 (revoke privilege)
Trace level: 2
Type: Report
Description: Open scan from SYSTABAUTH

```
Trace point:  6254
Subcomponent: Authorization
Module:       ARIXA08 (revoke privilege)
Trace level:  2
Type:         Report
Description:  Row for this object is found in SYSTABAUTH

Trace point:  6255
Subcomponent: Authorization
Module:       ARIXA08 (revoke privilege)
Trace level:  2
Type:         Report
Description:  Next from SYSTABAUTH

Trace point:  6256
Subcomponent: Authorization
Module:       ARIXA08 (revoke privilege)
Trace level:  2
Type:         Report
Description:  No more rows left. Close scan on SYSTABAUTH

Trace point:  6257
Subcomponent: Authorization
Module:       ARIXA08 (revoke privilege)
Trace level:  2
Type:         Report
Description:  In CASE_1

Trace point:  6258
Subcomponent: Authorization
Module:       ARIXA08 (revoke privilege)
Trace level:  2
Type:         Report
Description:  Open scan on SYSPROGAUTH

Trace point:  6259
Subcomponent: Authorization
Module:       ARIXA08 (revoke privilege)
Trace level:  2
Type:         Report
Description:  Next from SYSPROGAUTH

Trace point:  6260
Subcomponent: Authorization
Module:       ARIXA08 (revoke privilege)
Trace level:  2
Type:         Report
Description:  Delete from SYSPROGAUTH

Trace point:  6261
Subcomponent: Authorization
Module:       ARIXA08 (revoke privilege)
Trace level:  2
Type:         Report
Description:  Next from SYSPROGAUTH

Trace point:  6262
Subcomponent: Authorization
Module:       ARIXA08 (revoke privilege)
Trace level:  2
Type:         Report
Description:  Close scan on SYSPROGAUTH

Trace point:  6263
Subcomponent: Authorization
Module:       ARIXA08 (revoke privilege)
Trace level:  2
Type:         Report
Description:  In CASE_2

Trace point:  6264
Subcomponent: Authorization
Module:       ARIXA08 (revoke privilege)
Trace level:  2
Type:         Report
Description:  Open scan on SYSTABAUTH

Trace point:  6265
Subcomponent: Authorization
Module:       ARIXA08 (revoke privilege)
Trace level:  2
Type:         Report
Description:  Fetch from SYSPROGAUTH

Trace point:  6266
Subcomponent: Authorization
Module:       ARIXA08 (revoke privilege)
Trace level:  2
Type:         Report
Description:  Update RUNAUTH field in SYSPROGAUTH

Trace point:  6267
Subcomponent: Authorization
Module:       ARIXA08 (revoke privilege)
Trace level:  2
Type:         Report
Description:  Open scan on SYSCOLAUTH
```

```
Trace point:  6268
Subcomponent: Authorization
Module:       ARIXA08 (revoke privilege)
Trace level:  2
Type:         Report
Description:  Next from SYSCOLAUTH

Trace point:  6269
Subcomponent: Authorization
Module:       ARIXA08 (revoke privilege)
Trace level:  2
Type:         Report
Description:  Delete from SYSCOLAUTH

Trace point:  6270
Subcomponent: Authorization
Module:       ARIXA08 (revoke privilege)
Trace level:  2
Type:         Report
Description:  Next from SYSCOLAUTH

Trace point:  6271
Subcomponent: Authorization
Module:       ARIXA08 (revoke privilege)
Trace level:  2
Type:         Report
Description:  Close scan on SYSCOLUMNS

Trace point:  6272
Subcomponent: Authorization
Module:       ARIXA08 (revoke privilege)
Trace level:  2
Type:         Report
Description:  Delete from SYSTABAUTH

Trace point:  6273
Subcomponent: Authorization
Module:       ARIXA08 (revoke privilege)
Trace level:  2
Type:         Report
Description:  Update SYSTABAUTH

Trace point:  6274
Subcomponent: Authorization
Module:       ARIXA08 (revoke privilege)
Trace level:  2
Type:         Report
Description:  Next from SYSTABAUTH
```

```
Trace point:  6275
Subcomponent: Authorization
Module:       ARIXA08 (revoke privilege)
Trace level:  2
Type:         Report
Description:  Close scan on SYSTABAUTH

Trace point:  6285
Subcomponent: Authorization
Module:       ARIXA08 (revoke privilege)
Trace level:  1
Type:         Module exit
Description:  Exit from module ARIXA08

Trace point:  6286
Subcomponent: Authorization
Module:       ARIXA09 (revoke privilege)
Trace level:  1
Type:         Module entry
Description:  Entry to module ARIXA09

Trace point:  6287
Subcomponent: Authorization
Module:       ARIXA09 (revoke privilege)
Trace level:  2
Type:         Report
Description:  In RUNREVOK

Trace point:  6288
Subcomponent: Authorization
Module:       ARIXA09 (revoke privilege)
Trace level:  2
Type:         Report
Description:  Fetch from SYSACCESS

Trace point:  6289
Subcomponent: Authorization
Module:       ARIXA09 (revoke privilege)
Trace level:  2
Type:         Report
Description:  Open scan from SYSPROGAUTH

Trace point:  6290
Subcomponent: Authorization
Module:       ARIXA09 (revoke privilege)
Trace level:  2
Type:         Report
Description:  Next from SYSPROGAUTH
```

```
Trace point:    6291
Subcomponent:   Authorization
Module:         ARIXA09 (revoke privilege)
Trace level:    2
Type:           Report
Description:    Delete from SYSPROGAUTH

Trace point:    6292
Subcomponent:   Authorization
Module:         ARIXA09 (revoke privilege)
Trace level:    2
Type:           Report
Description:    Next from SYSPROGAUTH

Trace point:    6293
Subcomponent:   Authorization
Module:         ARIXA09 (revoke privilege)
Trace level:    2
Type:           Report
Description:    Close scan on SYSPROGAUTH

Trace point:    6294
Subcomponent:   Authorization
Module:         ARIXA09 (revoke privilege)
Trace level:    2
Type:           Report
Description:    In USRREVOK

Trace point:    6295
Subcomponent:   Authorization
Module:         ARIXA09 (revoke privilege)
Trace level:    2
Type:           Report
Description:    Fetch from SYSUSERAUTH

Trace point:    6296
Subcomponent:   Authorization
Module:         ARIXA09 (revoke privilege)
Trace level:    2
Type:           Report
Description:    Delete from SYSUSERAUTH

Trace point:    6297
Subcomponent:   Authorization
Module:         ARIXA09 (revoke privilege)
Trace level:    2
Type:           Report
Description:    Update SYSUSERAUTH

Trace point:    6298
Subcomponent:   Authorization
Module:         ARIXA09 (revoke privilege)
Trace level:    2
Type:           Report
Description:    Open scan on SYSUSERAUTH

Trace point:    6299
Subcomponent:   Authorization
Module:         ARIXA09 (revoke privilege)
Trace level:    2
Type:           Report
Description:    Next from SYSUSERAUTH

Trace point:    6300
Subcomponent:   Authorization
Module:         ARIXA09 (revoke privilege)
Trace level:    2
Type:           Report
Description:    Next from SYSUSERAUTH

Trace point:    6301
Subcomponent:   Authorization
Module:         ARIXA09 (revoke privilege)
Trace level:    2
Type:           Report
Description:    Close scan on SYSUSERAUTH

Trace point:    6302
Subcomponent:   Authorization
Module:         ARIXA09 (revoke privilege)
Trace level:    2
Type:           Report
Description:    In TABREVOK

Trace point:    6303
Subcomponent:   Authorization
Module:         ARIXA09 (revoke privilege)
Trace level:    2
Type:           Report
Description:    Fetch from SYSCATLOG

Trace point:    6304
Subcomponent:   Authorization
Module:         ARIXA09 (revoke privilege)
Trace level:    2
Type:           Report
Description:    Open scan from SYSTABAUTH
```

```
Trace point:  6305
Subcomponent: Authorization
Module:       ARIXA09 (revoke privilege)
Trace level:  2
Type:         Report
Description:  Next from SYSTABAUTH

Trace point:  6306
Subcomponent: Authorization
Module:       ARIXA09 (revoke privilege)
Trace level:  2
Type:         Report
Description:  Delete from SYSTABAUTH

Trace point:  6307
Subcomponent: Authorization
Module:       ARIXA09 (revoke privilege)
Trace level:  2
Type:         Report
Description:  Update SYSTABAUTH

Trace point:  6308
Subcomponent: Authorization
Module:       ARIXA09 (revoke privilege)
Trace level:  2
Type:         Report
Description:  Delete from SYSTABAUTH

Trace point:  6309
Subcomponent: Authorization
Module:       ARIXA09 (revoke privilege)
Trace level:  2
Type:         Report
Description:  Next from SYSTABAUTH

Trace point:  6310
Subcomponent: Authorization
Module:       ARIXA09 (revoke privilege)
Trace level:  2
Type:         Report
Description:  Close scan on SYSTABAUTH

Trace point:  6320
Subcomponent: Authorization
Module:       ARIXA09 (revoke privilege)
Trace level:  1
Type:         Module exit
Description:  Exit from module ARIXA09
```

```
Trace point:  6321
Subcomponent: Authorization
Module:       ARIXA10 (calculates minimum time)
Trace level:  1
Type:         Module entry
Description:  Entry to module ARIXA10

Trace point:  6322
Subcomponent: Authorization
Module:       ARIXA10 (calculates minimum time)
Trace level:  2
Type:         Report
Description:  Entry to procedure RUNCHECK

Trace point:  6323
Subcomponent: Authorization
Module:       ARIXA10 (calculates minimum time)
Trace level:  2
Type:         Report
Description:  In procedure RUNCHECK. Open scan on
              SYSPROGAUTH

Trace point:  6324
Subcomponent: Authorization
Module:       ARIXA10 (calculates minimum time)
Trace level:  2
Type:         Report
Description:  In procedure RUNCHECK. Next from SYSPROGAUTH

Trace point:  6325
Subcomponent: Authorization
Module:       ARIXA10 (calculates minimum time)
Trace level:  2
Type:         Report
Description:  In procedure RUNCHECK. Close scan on
              SYSPROGAUTH

Trace point:  6326
Subcomponent: Authorization
Module:       ARIXA10 (calculates minimum time)
Trace level:  2
Type:         Report
Description:  Exit from procedure RUNCHECK

Trace point:  6327
Subcomponent: Authorization
Module:       ARIXA10 (calculates minimum time)
Trace level:  2
Type:         Report
Description:  Entry to procedure TABCHECK
```

```
Trace point:  6328
Subcomponent: Authorization
Module:       ARIXA10 (calculates minimum time)
Trace level:  2
Type:         Report
Description:  In procedure TABCHECK. Open scan on SYSTABAUTH

Trace point:  6329
Subcomponent: Authorization
Module:       ARIXA10 (calculates minimum time)
Trace level:  2
Type:         Report
Description:  In procedure TABCHECK. Fetch from SYSCOLAUTH

Trace point:  6330
Subcomponent: Authorization
Module:       ARIXA10 (calculates minimum time)
Trace level:  2
Type:         Report
Description:  In procedure TABCHECK. Next from SYSTABAUTH

Trace point:  6331
Subcomponent: Authorization
Module:       ARIXA10 (calculates minimum time)
Trace level:  2
Type:         Report
Description:  In procedure TABCHECK. Close scan on
              SYSTABAUTH

Trace point:  6332
Subcomponent: Authorization
Module:       ARIXA10 (calculates minimum time)
Trace level:  2
Type:         Report
Description:  Exit from procedure TABCHECK

Trace point:  6337
Subcomponent: Authorization
Module:       ARIXA10 (calculates minimum time)
Trace level:  1
Type:         Module exit
Description:  Exit from module ARIXA10

Trace point:  6338
Subcomponent: Authorization
Module:       ARIXA11 (minimum authorization for a view)
Trace level:  1
Type:         Module entry
Description:  Entry to module ARIXA11

Trace point:  6339
Subcomponent: Authorization
Module:       ARIXA11 (minimum authorization for a view)
Trace level:  2
Type:         Report
Description:  All cases except update

Trace point:  6340
Subcomponent: Authorization
Module:       ARIXA11 (minimum authorization for a view)
Trace level:  2
Type:         Report
Description:  Open scan on SYSTABAUTH

Trace point:  6341
Subcomponent: Authorization
Module:       ARIXA11 (minimum authorization for a view)
Trace level:  2
Type:         Report
Description:  Next from SYSTABAUTH

Trace point:  6342
Subcomponent: Authorization
Module:       ARIXA11 (minimum authorization for a view)
Trace level:  2
Type:         Report
Description:  Next from SYSTABAUTH

Trace point:  6343
Subcomponent: Authorization
Module:       ARIXA11 (minimum authorization for a view)
Trace level:  2
Type:         Report
Description:  Close scan on SYSTABAUTH

Trace point:  6344
Subcomponent: Authorization
Module:       ARIXA11 (minimum authorization for a view)
Trace level:  2
Type:         Report
Description:  No authorization left on a view.

Trace point:  6345
Subcomponent: Authorization
Module:       ARIXA11 (minimum authorization for a view)
Trace level:  2
Type:         Report
Description:  Open scan on SYSTABAUTH
```

Trace point: 6346
Subcomponent: Authorization
Module: ARIXAll (minimum authorization for a view)
Trace level: 2
Type: Report
Description: Next from SYSTABAUTH

Trace point: 6347
Subcomponent: Authorization
Module: ARIXAll (minimum authorization for a view)
Trace level: 2
Type: Report
Description: Update SYSTABAUTH

Trace point: 6348
Subcomponent: Authorization
Module: ARIXAll (minimum authorization for a view)
Trace level: 2
Type: Report
Description: Next from SYSTABAUTH

Trace point: 6349
Subcomponent: Authorization
Module: ARIXAll (minimum authorization for a view)
Trace level: 2
Type: Report
Description: Close scan on SYSTABAUTH

Trace point: 6350
Subcomponent: Authorization
Module: ARIXAll (minimum authorization for a view)
Trace level: 2
Type: Report
Description: Update case

Trace point: 6351
Subcomponent: Authorization
Module: ARIXAll (minimum authorization for a view)
Trace level: 2
Type: Report
Description: Fetch from SYSTABAUTH

Trace point: 6352
Subcomponent: Authorization
Module: ARIXAll (minimum authorization for a view)
Trace level: 2
Type: Report
Description: Delete from SYSTABAUTH

Trace point: 6359
Subcomponent: Authorization
Module: ARIXAll (minimum authorization for a view)
Trace level: 1
Type: Module exit
Description: Exit from module ARIXAll

Trace point: 6380
Subcomponent: Subroutine
Module: ARIXSLN (calculate length)
Trace level: 1
Type: Module entry
Description: Entry to module ARIXSLN

Trace point: 6383
Subcomponent: Subroutine
Module: ARIXSLN (calculate length)
Trace level: 1
Type: Module exit
Description: Exit from Module ARIXSLN

Trace point: 6384
Subcomponent: Subroutine
Module: ARIXSRT (ROLLBACK WORK)
Trace level: 1
Type: Module entry
Description: Entry to module ARIXSRT

Trace point: 6385
Subcomponent: Subroutine
Module: ARIXSRT (ROLLBACK WORK)
Trace level: 2
Type: Report
Description: Rollback work call to DBSS

Trace point: 6386
Subcomponent: Subroutine
Module: ARIXSRT (ROLLBACK WORK)
Trace level: 1
Type: Report
Description: Entry through entry point ARIXSRTI

Trace point: 6388
Subcomponent: Subroutine
Module: ARIXSRT (ROLLBACK WORK)
Trace level: 1
Type: Module exit
Description: Exit from module ARIXSRT

Trace point: 6389
Subcomponent: Subroutine
Module: ARIXSDB (convert to characters)
Trace level: 1
Type: Module entry
Description: Entry to module ARIXSDB

Trace point: 6391
Subcomponent: Subroutine
Module: ARIXSDB (convert to characters)
Trace level: 1
Type: Module exit
Description: Exit from module ARIXSDB

Trace point: 6392
Subcomponent: Subroutine
Module: ARIXSUT (unique character string)
Trace level: 1
Type: Module entry
Description: Entry to module ARIXSUT

Trace point: 6393
Subcomponent: Subroutine
Module: ARIXSUT (unique character string)
Trace level: 2
Type: Report
Description: Display timestamp

Trace point: 6395
Subcomponent: Subroutine
Module: ARIXSUT (unique character string)
Trace level: 1
Type: Module exit
Description: Exit from module ARIXSUT

Trace point: 6396
Subcomponent: Subroutine
Module: ARIXSTM (time of a day)
Trace level: 1
Type: Module entry
Description: Entry to module ARIXSTM

Trace point: 6397
Subcomponent: Subroutine
Module: ARIXSTM (time of a day)
Trace level: 1
Type: Module exit
Description: Exit from module ARIXSTM

Trace point: 6398
Subcomponent: Subroutine
Module: ARIXSDT (get date)
Trace level: 1
Type: Module entry
Description: Entry to module ARIXSDT

Trace point: 6399
Subcomponent: Subroutine
Module: ARIXSDT (get date)
Trace level: 1
Type: Module exit
Description: Exit from module ARIXSDT

Trace point: 6400
Subcomponent: Security/Audit
Module: ARIXA02 (check authority)
Trace level: 1 or 2
Type: Report
Description: Check RUNAUTH, SELECT, INSERT, DELETE, ALTER, INDEX, UPDATE, RESOURCE, and DBA

Trace point: 6405
Subcomponent: Security/Audit
Module: ARIXA07 (GRANT privilege)
Trace level: 1 or 2
Type: Report
Description: Grants of a special privilege and RUN authority

Trace point: 6406
Subcomponent: Security/Audit
Module: ARIXA06 (Grant authority)
Trace level: 1 or 2
Type: Report
Description: Initial grants of RUN authority to the program prepper

Trace point: 6410
Subcomponent: Security/Audit
Module: ARIXA01 (CONNECT)
Trace level: 1 or 2
Type: Report
Description: CONNECT and SCHEDULE

Trace Point: 6500
Subcomponent: Interpreter
Module: ARIXI20 (update statistics)
Trace level: 2
Type: Report
Description: Open scan on syscolumns

```
Trace Point:  6501
Subcomponent: Interpreter
Module:       ARIXI20 (update statistics)
Trace level:  2
Type:         Report
Description:  Next from syscolumns
```

```
Trace Point:  6502
Subcomponent: Interpreter
Module:       ARIXI20 (update statistics)
Trace level:  2
Type:         Report
Description:  Update syscolumns
```

```
Trace Point:  6503
Subcomponent: Interpreter
Module:       ARIXI20 (update statistics)
Trace level:  2
Type:         Report
Description:  Close scan on syscolumns
```

```
Trace Point:  6510
Subcomponent: Interpreter
Module:       ARIXIST (statistics for non-indexed columns)
Trace level:  1
Type:         Module entry
Description:  Entry to ARIXIST
```

```
Trace Point:  6511
Subcomponent: Interpreter
Module:       ARIXIST (statistics for non-indexed columns)
Trace level:  1
Type:         Module exit
Description:  Early exit
```

```
Trace Point:  6512
Subcomponent: Interpreter
Module:       ARIXIST (statistics for non-indexed columns)
Trace level:  1
Type:         Module exit
Description:  Early exit
```

```
Trace Point:  6513
Subcomponent: Interpreter
Module:       ARIXIST (statistics for non-indexed columns)
Trace level:  1
Type:         Report
Description:  Map size
```

```
Trace Point:  6514
Subcomponent: Interpreter
Module:       ARIXIST (statistics for non-indexed columns)
Trace level:  2
Type:         Report
Description:  Free allocated storage
```

```
Trace Point:  6515
Subcomponent: Interpreter
Module:       ARIXIST (statistics for non-indexed columns)
Trace level:  2
Type:         Report
Description:  Allocate storage
```

```
Trace Point:  6516
Subcomponent: Interpreter
Module:       ARIXIST (statistics for non-indexed columns)
Trace level:  2
Type:         Report
Description:  Count bits on
```

```
Trace Point:  6517
Subcomponent: Interpreter
Module:       ARIXIST (statistics for non-indexed columns)
Trace level:  2
Type:         Report
Description:  Index of next larger prime
```

```
Trace Point:  6519
Subcomponent: Interpreter
Module:       ARIXIST (statistics for non-indexed columns)
Trace level:  2
Type:         Report
Description:  Initialize storage
```

```
Trace Point:  6520
Subcomponent: Interpreter
Module:       ARIXIST (statistics for non-indexed columns)
Trace level:  2
Type:         Report
Description:  Allocate hash tables
```

```
Trace Point:  6521
Subcomponent: Interpreter
Module:       ARIXIST (statistics for non-indexed columns)
Trace level:  2
Type:         Report
Description:  Linear counting scan
```

```
Trace Point:  6522
Subcomponent: Interpreter
Module:       ARIXIST (statistics for non-indexed columns)
Trace level:  2
Type:         Report
Description:  Calling COMPENS
```

```
Trace Point:  6523
Subcomponent: Interpreter
Module:       ARIXIST (statistics for non-indexed columns)
Trace level:  1
Type:         Report
Description:  After a SPLIT
```

```
Trace Point:  6524
Subcomponent: Interpreter
Module:       ARIXIST (statistics for non-indexed columns)
Trace level:  2
Type:         Report
Description:  Check for a hit
```

```
Trace Point:  6525
Subcomponent: Interpreter
Module:       ARIXIST (statistics for non-indexed columns)
Trace level:  2
Type:         Report
Description:  Result
```

```
Trace Point:  6526
Subcomponent: Interpreter
Module:       ARIXIST (statistics for non-indexed columns)
Trace level:  2
Type:         Report
Description:  Entered COMPENS
```

```
Trace Point:  6527
Subcomponent: Interpreter
Module:       ARIXIST (statistics for non-indexed columns)
Trace level:  2
Type:         Report
Description:  Called logbase
```

```
Trace Point:  6528
Subcomponent: Interpreter
Module:       ARIXIST (statistics for non-indexed columns)
Trace level:  2
Type:         Report
Description:  COMPENS exits
```

```
Trace Point:  6529
Subcomponent: Interpreter
Module:       ARIXIST (statistics for non-indexed columns)
Trace level:  2
Type:         Report
Description:  Enter ENCODLIN
```

```
Trace Point:  6530
Subcomponent: Interpreter
Module:       ARIXIST (statistics for non-indexed columns)
Trace level:  2
Type:         Report
Description:  ENCODLIN returns
```

```
Trace Point:  6531
Subcomponent: Interpreter
Module:       ARIXIST (statistics for non-indexed columns)
Trace level:  2
Type:         Report
Description:  After hashing
```

```
Trace Point:  6532
Subcomponent: Interpreter
Module:       ARIXIST (statistics for non-indexed columns)
Trace level:  2
Type:         Report
Description:  Match table
```

```
Trace Point:  6533
Subcomponent: Interpreter
Module:       ARIXIST (statistics for non-indexed columns)
Trace level:  2
Type:         Report
Description:  Match found
```

```
Trace Point:  6535
Subcomponent: Interpreter
Module:       ARIXIST (statistics for non-indexed columns)
Trace level:  2
Type:         Report
Description:  Open scan for logarithmic counting
```

```
Trace Point:  6536
Subcomponent: Interpreter
Module:       ARIXIST (statistics for non-indexed columns)
Trace level:  2
Type:         Report
Description:  UPDMAP returns
```

Trace Point: 6537
Subcomponent: Interpreter
Module: ARIXIST (statistics for non-indexed columns)
Trace level: 2
Type: Report
Description: Result returns - L0 function

Trace Point: 6538
Subcomponent: Interpreter
Module: ARIXIST (statistics for non-indexed columns)
Trace level: 2
Type: Report
Description: Count calculation

Trace Point: 6539
Subcomponent: Interpreter
Module: ARIXIST (statistics for non-indexed columns)
Trace level: 2
Type: Report
Description: ENCODLOG input

Trace Point: 6540
Subcomponent: Interpreter
Module: ARIXIST (statistics for non-indexed columns)
Trace level: 2
Type: Report
Description: Word to be squared

Trace Point: 6541
Subcomponent: Interpreter
Module: ARIXIST (statistics for non-indexed columns)
Trace level: 2
Type: Report
Description: Output from ENCODLOG

Trace Point: 6542
Subcomponent: Interpreter
Module: ARIXIST (statistics for non-indexed columns)
Trace level: 2
Type: Report
Description: Input to prehash

Trace Point: 6543
Subcomponent: Interpreter
Module: ARIXIST (statistics for non-indexed columns)
Trace level: 2
Type: Report
Description: Returned from CALHASH

Trace Point: 6544
Subcomponent: Interpreter
Module: ARIXIST (statistics for non-indexed columns)
Trace level: 2
Type: Report
Description: Field returns

Trace Point: 6599
Subcomponent: Interpreter
Module: ARIXIST (statistics for non-indexed columns)
Trace level: 1
Type: Module Exit
Description: Exit from ARIXIST

9900 - 9909

DBSS Service Temporary Trace Point Descriptors (9900 - 9909) are located on page 435.

RDS Service Temporary Trace Point Descriptors

Note: RDS Service Temporary Trace Points (below) are contained in module ARIXTR5. See page 448 for other ARIXTR5 trace points.

Trace point: 9950 (hex 26DE)
Subcomponent: RDS Service Temporary (any subcomponent)
(Displays as Code Generator: COMP=CODE)
Module: Any RDS (Displays 'RDSANY1')
Trace level: 1
Type: Report
Description: Provides trace output for a temporary trace point inserted into any RDS module. Displays up to four passed local variables:
- The first is called VARNRA1 and is assumed to be 4 bytes, displayed hex.
- The second is called VARNRA2 and is assumed to be 16 bytes, displayed hexdump.
- The third is called VARNRA3 and is assumed to be 16 bytes, displayed hexdump.
- The fourth is called VARNRA4 and is assumed to be 64 bytes, displayed hexdump.
- To suppress display of any of the above, pass 0 data address in call parameter array.

All output is displayed level 1 (only requiring trace to be active for the current agent).

Trace point: 9951 (hex 26DF)
Subcomponent: RDS Service Temporary (any subcomponent)
(Displays as Code Generator: COMP=CODE)
Module: Any RDS (Displays 'RDSANY2')
Trace level: 1
Type: Report
Description: Provides trace output for a temporary trace point inserted into any RDS module. Displays up to four passed local variables:
- The first is called VARNRB1 and is assumed to be 4 bytes, displayed hex.
- The second is called VARNRB2 and is assumed to be 16 bytes, displayed hexdump.
- The third is called VARNRB3 and is assumed to be 16 bytes, displayed hexdump.
- The fourth is called VARNRB4 and is assumed to be 64 bytes, displayed hexdump.
- To suppress display of any of the above, pass 0 data address in call parameter array.

All output is displayed level 1 (only requiring trace to be active for the current agent).

Trace point: 9952 (hex 26E0)
Subcomponent: RDS Service Temporary (any subcomponent)
(Displays as Code Generator: COMP=CODE)
Module: Any RDS (Displays 'RDSANY3')
Trace level: 1
Type: Report
Description: Provides trace output for a temporary trace point inserted into any RDS module. Displays up to two passed local variables:
- The first is called VARNRC1 and is assumed to be 4 bytes, displayed hex.
- The second is called VARNRC2 and is assumed to be 16 bytes, displayed hexdump.
- To suppress display of any of the above, pass 0 data address in call parameter array.

All output is displayed level 1 (only requiring trace to be active for the current agent).

Trace point: 9953 (hex 26E1)
Subcomponent: RDS Service Temporary (any subcomponent)
(Displays as Code Generator: COMP=CODE)
Module: Any RDS (Displays 'RDSANY4')
Trace level: 1
Type: Report
Description: Provides trace output for a temporary trace point inserted into any RDS module. Displays up to two passed local variables:
- The first is called VARNRD1 and is assumed to be 4 bytes, displayed hex.
- The second is called VARNRD2 and is assumed to be 64 bytes, displayed hexdump.
- To suppress display of any of the above, pass 0 data address in call parameter array.

All output is displayed level 1 (only requiring trace to be active for the current agent).

Trace point: 9954 (hex 26E2)
Subcomponent: RDS Service Temporary (any subcomponent)
(Displays as Code Generator: COMP=CODE)
Module: Any RDS (Displays 'RDSANY5')
Trace level: 1
Type: Report
Description: Provides trace output for a temporary trace point inserted into any RDS module. Displays up to two passed local variables:
- The first is called VARNRE1 and is assumed to be 4 bytes, displayed hex.
- The second is called VARNRE2 and its length is passed as a parameter (a fullword) immediately following the variable itself. The variable is displayed hexdump.
- To suppress display of any of the above, pass 0 data address in call parameter array.

All output is displayed level 1 (only requiring trace to be active for the current agent).

Trace point: 9955 (hex 26E3)
Subcomponent: RDS Service Temporary (any subcomponent)
(Displays as Code Generator: COMP=CODE)
Module: Any RDS (Displays 'RDSANY6')
Trace level: 1
Type: Report
Description: Provides trace output for a temporary trace point inserted into any RDS module. Displays up to three passed local variables:
- The first is called VARNRF1 and is assumed to be 4 bytes, displayed hex.
- The second is called VARNRF2 and its length is passed as a parameter (a fullword) immediately following the variable itself. The variable is displayed hexdump.
- The third is called VARNRF3 and its length is passed as a parameter (a fullword) immediately following the variable itself. The variable is displayed hexdump.
- To suppress display of any of the above, pass 0 data address in call parameter array.

All output is displayed level 1 (only requiring trace to be active for the current agent).

Trace point: 9956 (hex 26E4)
Subcomponent: RDS Service Temporary (any subcomponent)
(Displays as Code Generator: COMP=CODE)
Module: Any RDS (Displays 'RDSANY7')
Trace level: 1
Type: Report
Description: Provides trace output for a temporary trace point inserted into any RDS module. Displays up to two passed local variables:
- The first is called VARNRG1 and its length is passed as a parameter (a fullword) immediately following the variable itself. The variable is displayed hexdump.
- The second is called VARNRG2 and its length is passed as a parameter (a fullword) immediately following the variable itself. The variable is displayed hexdump.
- To suppress display of any of the above, pass 0 data address in call parameter array.

All output is displayed level 1 (only requiring trace to be active for the current agent).

I/O TRACE (VSE ONLY)

The I/O trace facility takes advantage of the common entry ARISYSD5 and linkage structure for all I/O requests to sequential files. It works for the following files:

```
SYSIPT  ┐
SYSLST  > VSE system logical units
SYSPCH  ┘
SYS001  (PREP work file 1)
SYS002  (PREP work file 2)
SYS003  (PREP work file 3)
Sequential files on disk
Sequential files on tape
Source Statement Library (READ MEMBER statement)
```

The I/O trace function is not activated for normal use. To activate it:

1. ZAP branch instruction to a "no-op" instruction.
2. Set UPSI bits to inform trace what to do.

PERFORMING THE ZAP

The fastest way to activate I/O trace is via VSE PDZAP. Use the 'scan function' for instruction "0A21". Subtract 8 from this address and you get the address of the instruction to be "zapped" - change instruction "47F0" to "4700".

Remember that ARISYSDD is linked with different components of SQL/DS. One ZAP affects only one component. For example, if you ZAP phase ARIPRPA (assembler PREP), it will trace I/O requests to SYSIPT, SYSPCH, SYSLST, and SYS001. If you ZAP phase ARISQLDS (DSC/DBSS), it will trace I/O requests to tape (archive, trace), SYSIPT, and SYSLST.

SETTING THE UPSI BITS

Specify trace control information by means of the VSE JCL UPSI bits. The following are valid UPSI bit specifications:

Bit(s)	Meaning
0 - 3	Not used
4	Output to console and SYSLST
5	Output to console only
6	Trace level 2 (default output to SYSLST)
7	Trace level 1 (default output to SYSLST)

UPSI Examples:

```
// UPSI 00000010    (Trace level 2 to SYSLST)
// UPSI 00001001    (Trace level 1 to SYSLST and console)
// UPSI 00000111    (Trace level 2 to console)
```

I/O TRACE EXAMPLES

The following two examples are copies of portions of I/O trace output to SYSLST, with some space added to provide for comments.

Example 1: Sequential File I/O Trace Level 1 SYSLST Output

```
// JOB ONLINE RESOURCE MANAGER PREPARATION
// UPSI 00001001
* ************************************************************
* SQL/DS  MULTI USER MODE      STEP 1 : PREP ARIRSQL          *
* ************************************************************
// PAUSE      HIT ENTER TO CONTINUE
// EXEC ARIPRPA,SIZE=AUTO,PARM='PREP=ARITEST,USERID=SQLDBA/SQLDBAPW'

 OPENOUT SYSLST  06123C 000000          R=000079    C=000000
 OPENOUT SYSLST  06129A 000000 DISK     R=000050    C=000000
 OPENIN  SYSIPT  061180 000000          R=000050    C=000000
 PUT     SYS001  06129A 081C1A DISK     R=000050    C=000001
 PUT     SYS001  06129A 081C1A DISK     R=000050    C=000002
 PUT     SYS001  06129A 081C1A DISK     R=000050    C=000003
 PUT     SYS001  06129A 081C1A DISK     R=000050    C=000004
 PUT     SYS001  06129A 081C1A DISK     R=000050    C=000005
 PUT     SYS001  06129A 081C1A DISK     R=000050    C=000006
 PUT     SYS001  06129A 081C1A DISK     R=000050    C=000007
 PUT     SYS001  06129A 081C1A DISK     R=000050    C=000008
 PUT     SYS001  06129A 081C1A DISK     R=000050    C=000009
 PUT     SYS001  06129A 081C1A DISK     R=000050    C=00000A
 GET     SYSIPT  061180 081CBA          R=000050    C=000001
 GET     SYSIPT   SSL MEMBER = A.ARIRSQL     @PROCESS CC('@') ADEFS M
 PUT     SYS001  06129A 081D5A DISK     R=000050    C=00000B
 GET     SYSIPT   SSL MEMBER = A.ARIRSQL        STATS IDR GMAR(
 PUT     SYS001  06129A 081D5A DISK     R=000050    C=00000C
 GET     SYSIPT   SSL MEMBER = A.ARIRSQL     TITLE('ARIRSQL - ONLINE
 PUT     SYS001  06129A 081D5A DISK     R=000050    C=00000D

 GET     SYSIPT   SSL MEMBER = A.ARIRSQL     /*A000000-999999

 PUT     SYS001  06129A 081D5A DISK     R=000050    C=00000E

 GET     SYSIPT   SSL MEMBER = A.ARIRSQL     /**********START OF SPECI

 PUT     SYS001  06129A 081D5A DISK     R=000050    C=00000F

 GET     SYSIPT   SSL MEMBER = A.ARIRSQL     /*
```

CALLER'S PARAMETERS:
- CALLTYPE
- DNAME
- ↑ FILE DESCRIPTOR BLOCK
- BUFFER ADDRESS

DEVICE TYPE

RECORD LENGTH

RECORD COUNT

```
PUT     SYS001  06129A 081D5A DISK    R=000050    C=000010
GET     SYSIPT   SSL MEMBER = A.ARIRSQL     /*01 MODULE-NAME= ARIRS
PUT     SYS001  06129A 081D5A DISK    R=000050    C=000011
GET     SYSIPT   SSL MEMBER = A.ARIRSQL     /*
PUT     SYS001  06129A 081D5A DISK    R=000050    C=000012
GET     SYSIPT   SSL MEMBER = A.ARIRSQL     /*01 DSCRIPTIVE-NAME= R
PUT     SYS001  06129A 081D5A DISK    R=000050    C=000013
GET     SYSIPT   SSL MEMBER = A.ARIRSQL     /*
PUT     SYS001  06129A 081D5A DISK    R=000050    C=000014
GET     SYSIPT   SSL MEMBER = A.ARIRSQL     /*
PUT     SYS001  06129A 081D5A DISK    R=000050    C=000015
GET     SYSIPT   SSL MEMBER = A.ARIRSQL     /*  STATUS =
PUT     SYS001  06129A 081D5A DISK    R=000050    C=000016
GET     SYSIPT   SSL MEMBER = A.ARIRSQL     /*
PUT     SYS001  06129A 081D5A DISK    R=000050    C=000017
GET     SYSIPT   SSL MEMBER = A.ARIRSQL     /*  FUNCTION =
PUT     SYS001  06129A 081D5A DISK    R=000050    C=000018
GET     SYSIPT   SSL MEMBER = A.ARIRSQL     /*
PUT     SYS001  06129A 081D5A DISK    R=000050    C=000019
GET     SYSIPT   SSL MEMBER = A.ARIRSQL     /*  NOTES =
PUT     SYS001  06129A 081D5A DISK    R=000050    C=00001A
GET     SYSIPT   SSL MEMBER = A.ARIRSQL     /*
PUT     SYS001  06129A 081D5A DISK    R=000050    C=00001B
GET     SYSIPT   SSL MEMBER = A.ARIRSQL     /*
PUT     SYS001  06129A 081D5A DISK    R=000050    C=00001C
GET     SYSIPT   SSL MEMBER = A.ARIRSQL     /*      DEPENDENCIES =
PUT     SYS001  06129A 081D5A DISK    R=000050    C=00001D
```

FIRST 30 DATA BYTES. DISPLAYED IN TRACE LEVEL 1 ONLY WHEN INPUT FROM SOURCE STATEMENT LIBRARY

INSTEAD FROM SYSIPT, INPUT IS READ FROM LIBRARY MEMBER A.ARIRSQL.

NOTE: OUTPUT TO SYSLST MAY INTERFERE WITH REGULAR SYSLST OUTPUT

Example 2: Sequential File I/O Trace Level 2 SYSLST Output

```
// JOB ONLINE RESOURCE MANAGER PREPARATION
// UPSI 00001011
* ************************************************************
* SQL/DS  MULTI USER MODE      STEP 1 : PREP ARIRSQL          *
* ************************************************************
// PAUSE      HIT ENTER TO CONTINUE
// EXEC ARIPRPA,SIZE=AUTO,PARM='PREP=ARITEST,USERID=SQLDBA/SQLDBAPW'
```

```
OPENOUT SYSLST  06123C 000000          R=000079    C=000000

06123C 000000  C1D9C9C6 C4C5E2C3 0000006E 00000000  * ARIFDESC........ *
06124C 000010  00000079 00000000 00000000 00068F88[1] * ..............<. *
06125C 000020  00000000 00000000 00000000 00061800[2] * ................. *
06126C 000030  00000000 00000000 00000000 00000000  * ................. *
06127C 000040  010001D0 E8000000 00000000 00000000  * ....Y........... *
06128C 000050  00000000 00000000 00000000 00000000  * ................. *
06129C 000060  00000000 00000000 00000000 00000000  * ..........DI..   *

OPENOUT SYS001  06129A 000000          R=000050    C=000000

06129A 000000  C1D9C9C6 C4C5E2C3 0000006E 00000000  * ARIFDESC....DI.. *
0612AA 000010  00000050 00000050 00000000 001D1788[1] * ...&...&........ *
0612BA 000020  00000000 00000000 00065900 00061800[2] * ................. *
0612CA 000030  00066D00 00065180 00065E00 00000000  * ..........$..... *
0612DA 000040  010001D0 E8E80000 00000000 00000000  * ....YY........... *
0612EA 000050  00000000 00000000 00000000 00000000  * ................. *
0612FA 000060  00066A00 00000000 00000006 00000000  * ..............   *

OPENIN  SYSIPT  061180 000000          R=000050    C=000000

061180 000000  C1D9C9C6 C4C5E2C3 0000006E 00008000  * ARIFDESC........ *
061190 000010  00000050 00000050 00000000 0006A930[1] * ...&...&........ *
0611A0 000020  00000000 80000000 00069E80 00061800[2] * ................. *
0611B0 000030  00000000 00065180 0006A380 00000000  * ................. *
0611C0 000040  010080D0 E8000000 00000000 00000000  * ....Y........... *
0611D0 000050  00000000 00000000 00000000 00000000  * ................. *
0611E0 000060  00000000 00000000 00000000           * ..............   *

PUT     SYS001  06129A 081C1A DISK    R=000050    C=000001
                DATA =>           MACRO

PUT     SYS001  06129A 081C1A DISK    R=000050    C=000002
                DATA =>           ARISECTS

PUT     SYS001  06129A 081C1A DISK    R=000050    C=000003
                DATA =>           GBLC  &DVKSECT
```

CALLER'S PARAMETERS:
- CALLTYPE
- DNAME (OR LOG UNIT)
- FILE DESCRIPTOR BLOCK ADDRESS
- BUFFER ADDRESS

[1] DTF address
[2] Module ARISD000 address

FILE DESCRIPTOR BLOCKS FOR:
- SYSLST
- SYS001
- SYSIPT

```
PUT     SYS001  06129A 081C1A DISK    R=000050    C=000004
        DATA =>   &DVKSECT SETC  '&SYSECT'
PUT     SYS001  06129A 081C1A DISK    R=000050    C=000005
        DATA =>            MEND
PUT     SYS001  06129A 081C1A DISK    R=000050    C=000006
        DATA =>            MACRO
PUT     SYS001  06129A 081C1A DISK    R=000050    C=000007
        DATA =>            ARISECTR
                                3          4          5
PUT     SYS001  06129A 081C1A DISK    R=000050    C=000008
        DATA =>            GBLC  &DVKSECT
PUT     SYS001  06129A 081C1A DISK    R=000050    C=000009
        DATA =>   &DVKSECT CSECT
PUT     SYS001  06129A 081C1A DISK    R=000050    C=00000A
        DATA =>            MEND
GET     SYSIPT  061180 081CBA         R=000050    C=000001
        DATA =>   READ MEMBER ARIRSQL
GET     SYSIPT   SSL MEMBER = A.ARIRSQL     @PROCESS CC('@') ADEFS M
PUT     SYS001  06129A 081D5A DISK    R=000050    C=00000B
        DATA =>    @PROCESS CC('@') ADEFS MACH(S370R) MAR(
GET     SYSIPT  SSL MEMBER = A.ARIRSQL           STATS IDR GMAR(
PUT     SYS001  06129A 081D5A DISK    R=000050    C=00000C
        DATA =>            STATS IDR GMAR(1,72)
```

3 DEVICE TYPE
4 RECORD LENGTH (HEX)
5 RECORD COUNT (HEX)

FIRST 30 BYTES OF RECORD (DATA)

INSTEAD FROM SYSIPT, INPUT IS READ FROM LIBRARY MEMBER A.ARIRSQLP

ARISCAT (CATALOG GENERATION SOURCE INPUT)

ARISCAT contains the input records used by catalog generation to build the SQL/DS catalogs and update them with the required information. GENCAT reads and processes each record until it reaches end-of-file. When all records have been processed, the information is recorded in the data base through DBSI (ARIYM00) calls.

```
INITIALIZE DBSPACE 1:    NHEADER=8, PCTINDX=60, PCTFREE=0, LOCKMODE=T
INITIALIZE DBSPACE 2:    NHEADER=8, PCTINDX=0, PCTFREE=1,LOCKMODE=P
SET DEFAULT PCTINDX     = 33
SET DEFAULT NHEADER     =  8
SET DEFAULT SEGPCTFREE = 15
SET DEFAULT INDXPCTFREE = 10
SET DEFAULT PAGE        =  1
* WARNING: SYSBOOT MUST BE FIRST RELATION DEFINED IN  DBSPACE 1 OF
*          THE DBSS. IT MUST HAVE RID OF '8001'.
DEFREL SYSBOOT
   ( VERSION       SMALLINT,
     DATE          CHAR(6),
     BOOT          VARCHAR(254) )
DEFREL SYSDBSPACES
  (  DBSPACENAME   VARCHAR(MNL),
     DBSPACENO     DBAHW,
     OWNER         CHARACTER(8),
     DBSPACETYPE   SMALLINT,
     NTABS         DBAHW,
     NPAGES        INTEGER,
     NRHEADER      DBAHW,
     PCTINDX       DBAHW,
     FREEPCT       SMALLINT,
     LOCKMODE      CHARACTER(1),
     NACTIVE       DBAINT,
     POOL          DBAHW         )
DEFIMG IDBSPACE SYSDBSPACES ( OWNER, DBSPACENAME )
DEFIMG IDBSPACE2 SYSDBSPACES ( DBSPACETYPE, OWNER, NPAGES )
DEFUIM IDBSPACE3 SYSDBSPACES ( DBSPACENO )
FILL SYSDBSPACES
DEFREL SYSCATALOG
   ( TNAME         VARCHAR(MNL),
     CREATOR       CHAR(8),
     TABLETYPE     CHAR(1),
     NCOLS         SMALLINT,
     REMARKS       VARCHAR(254),
     DBSPACENO     DBAHW,
     DBSPACENAME   VARCHAR(MNL),
     TABID         DBAHW,
     CLUSTERTYPE   CHAR(1),
     CLUSTERROW    DBAINT,
     AVGROWLEN     DBAHW,
     ROWCOUNT      DBAINT,
     NPAGES        DBAINT,
     PCTPAGES      DBAHW,
```

```
     NOVERFLOW     DBAINT,
     LFDTABID      DBAHW,
     LFDLINK       DBAHW,
     LFDDBSPACE    DBAHW )
DEFREL SYSSYNONYMS
    (  USERID        CHARACTER(8),
       ALTNAME       VARCHAR(MNL),
       CREATOR       CHARACTER(8),
       TNAME         VARCHAR(MNL)  )
DEFREL SYSCOLUMNS
   ( CNAME          VARCHAR(MNL),
     TNAME          VARCHAR(MNL),
     CREATOR        CHAR(8),
     COLNO          SMALLINT,
     COLTYPE        CHAR(8),
     LENGTH         CHAR(7),
     SYSLENGTH      DBAHW,
     NULLS          CHAR(1),
     REMARKS        VARCHAR(254),
     COLCOUNT       DBAINT,
     HIGH2KEY       VARCHAR(12),
     LOW2KEY        VARCHAR(12),
     AVGCOLLEN      DBAHW,
     ORDERFIELD     CHAR(1) )
DEFREL  SYSINDEXES
   ( INAME          VARCHAR(MNL),
     ICREATOR       CHAR(8),
     TNAME          VARCHAR(MNL),
     CREATOR        CHAR(8),
     COLNAMES       VARCHAR(100),
     INDEXTYPE      CHAR(1),
     CLUSTER        CHAR(1),
     IID            DBAHW,
     COLNUMBERS     VARCHAR(34),
     KEYLEN         DBAHW,
     FIRSTKEYCOUNT  DBAINT,
     FULLKEYCOUNT   DBAINT,
     LOCKMODE       CHAR(1),
     NLEAF          DBAINT,
     NLEVELS        DBAHW,
     IPCTFREE       SMALLINT  )
DEFREL SYSLINKS
   ( LINKNAME       VARCHAR(MNL),
     LCREATOR       CHAR(8),
     PRNAME         VARCHAR(MNL),
     PCREATOR       CHAR(8),
     CRNAME         VARCHAR(MNL),
     CCREATOR       CHAR(8),
     LINKTEXT       VARCHAR(254),
     LID            DBAHW,
     PARENTCOLS     VARCHAR(34),
     CHILDCOLS      VARCHAR(34),
     ORDERCOLS      VARCHAR(34),
     ORPHANS        CHAR(1),
```

```
    UNIQ           CHAR(1),
    CLUSTERLINK    CHAR(1),
    NPARENT        DBAINT,
    NORPHANS       DBAINT,
    MAXTWINCHAIN   DBAINT,
    MAX2TWINCHAIN  DBAINT,
    AVGTWINCHAIN   DBAINT,
    USEABLE        CHAR(1)    )
DEFREL SYSVIEWS
  (  VIEWNAME      VARCHAR(MNL),
    VCREATOR       CHAR(8),
    SEQNO          SMALLINT,
    VIEWTEXT       VARCHAR(254)    )
DEFREL SYSUSAGE
  (  BNAME         VARCHAR(MNL),
    BCREATOR       CHAR(8),
    BTYPE          CHAR(1),
    DNAME          VARCHAR(MNL),
    DCREATOR       CHAR(8),
    DTYPE          CHAR(1),
    TIMESTAMP      CHAR(8)    )
DEFREL SYSTABAUTH
  (  GRANTOR       CHAR(8),
    GRANTEE        VARCHAR(8),
    GRANTEETYPE    CHAR(1),
    SCREATOR       CHAR(8),
    STNAME         VARCHAR(MNL),
    TCREATOR       CHAR(8),
    TTNAME         VARCHAR(MNL),
    TIMESTAMP      CHAR(12),
    UPDATECOLS     CHAR(1),
    SELECTAUTH     CHAR(1),
    INSERTAUTH     CHAR(1),
    UPDATEAUTH     CHAR(1),
    DELETEAUTH     CHAR(1),
    ALTERAUTH      CHAR(1),
    INDEXAUTH      CHAR(1) )
DEFREL SYSPROGAUTH
  (  GRANTOR       CHAR(8),
    GRANTEE        CHAR(8),
    CREATOR        CHAR(8),
    PROGNAME       VARCHAR(MNL),
    TIMESTAMP      CHAR(12),
    RUNAUTH        CHAR(1) )
DEFREL SYSCOLAUTH
  (  GRANTOR       CHAR(8),
    GRANTEE        CHAR(8),
    CREATOR        CHAR(8),
    TNAME          VARCHAR(MNL),
    TIMESTAMP      CHAR(12),
    COLNAME        VARCHAR(MNL)    )
DEFREL SYSUSERAUTH
  (  NAME          CHAR(8),
    AUTHOR         CHAR(8),
```

```
     RESOURCEAUTH  CHAR(1),
     DBAAUTH       CHAR(1),
     PASSWORD      CHAR(8),
     SCHEDULEAUTH  CHAR(1) )
DEFREL SYSACCESS
  (  TNAME         VARCHAR(MNL),
     CREATOR       CHAR(8),
     DBSPACENO     DBAHW,
     TABID         DBAHW,
     LINKID        DBAHW,
     FIRSTROW      DBAINT,
     TIMESTAMP     CHARACTER(17),
     VALID         CHARACTER(1),
     TABTYPE       CHARACTER(1)   )
DEFREL SYSDROP
  (  DBSPACENO     DBAHW,
     TABID         DBAHW,
     QUALF         CHARACTER(1)   )
***   INDEXES   ***
DEFUIM ICAT SYSCATALOG ( TNAME, CREATOR )
DEFUIM ISYN SYSSYNONYMS ( USERID, ALTNAME )
DEFUIM ICOL SYSCOLUMNS ( TNAME, CREATOR, CNAME )
DEFIMG IINDX  SYSINDEXES  ( TNAME, CREATOR )
DEFUIM IINDX2 SYSINDEXES  ( INAME, ICREATOR )
DEFUIM ILINK SYSLINKS ( LINKNAME, LCREATOR )
DEFUIM IVIEWS SYSVIEWS ( VIEWNAME, VCREATOR, SEQNO )
DEFIMG IUSAGE SYSUSAGE ( BNAME, BCREATOR, BTYPE )
DEFIMG IUSAGE2 SYSUSAGE ( DNAME, DCREATOR, DTYPE )
DEFIMG ITABAUTH1 SYSTABAUTH ( GRANTEE, TCREATOR, TTNAME, GRANTEETYPE )
DEFIMG ITABAUTH2 SYSTABAUTH ( GRANTOR, SCREATOR, STNAME )
* THE FOLLOWING INDEX IS DEFINED ON SYSTABAUTH BUT THE NAME ON THE
* INSERT STATEMENT HAS BEEN SHORTENED TO ALLOW THE COMMAND TO FIT ON
* ONE CONTROL RECORD.  A SPECIAL CHECK WAS MADE IN ARISDFM TO HANDLE.
DEFIMG ITABAUTH3 SYSTAB ( GRANTOR,GRANTEE,SCREATOR,STNAME,GRANTEETYPE )
DEFIMG ITABAUTH4 SYSTABAUTH ( SCREATOR, STNAME )
DEFIMG ITABAUTH5 SYSTABAUTH ( TCREATOR, TTNAME )
DEFIMG IPROGAUTH1 SYSPROGAUTH ( CREATOR, PROGNAME, GRANTEE, RUNAUTH )
DEFIMG IPROGAUTH2 SYSPROGAUTH ( GRANTOR, GRANTEE,  CREATOR, PROGNAME )
DEFIMG ICOLAUTH1 SYSCOLAUTH ( CREATOR, TNAME, COLNAME, GRANTEE )
DEFIMG ICOLAUTH2 SYSCOLAUTH ( TIMESTAMP, COLNAME )
DEFIMG ICOLAUTH3 SYSCOLAUTH ( GRANTEE, TIMESTAMP, COLNAME )
DEFUIM IUSERAUTH SYSUSERAUTH ( AUTHOR, NAME )
DEFUIM IACCESS SYSACCESS ( TNAME, CREATOR, TABTYPE )
*    DEFINE LINKS BETWEEN SYSTEM TABLES
DEFLINK COLLINK   FROM SYSCATALOG TO SYSCOLUMNS
DEFLINK INDXLINK  FROM SYSCATALOG TO SYSINDEXES
DEFLINK AUTHLINK  FROM SYSTABAUTH TO SYSCOLAUTH
DEFLINK CLINK     FROM SYSCATALOG TO SYSLINKS
DEFLINK PLINK     FROM SYSCATALOG TO SYSLINKS
*
*   INSERT THE TUPLES INTO THE RELATIONS WE DEFINED ABOVE
*
INSERT INTO SYSCATALOG ( 'SYSCATALOG', 'SYSTEM', 'R', 18,
                         'COMMENT', 1, 'SYS0001', 0, 'D', 0, -1, -1, -1,
```

```
                          -1, -1, 0, 0, 0 )
INSERT INTO SYSINDEXES ( 'ICAT', 'SYSTEM', 'SYSCATALOG', 'SYSTEM',
                         '+TNAME, +CREATOR', 'U', 'W', -1,
                         <2,1,2>, -1, -1, -1, 'K',
                          -1, -1, 0, )
INSERT INTO SYSCATALOG ( 'SYSSYNONYMS', 'SYSTEM', 'R', 4,
                        'COMMENT', 1, 'SYS0001', 0, 'I', 0, -1, -1, -1,
                          -1, -1, 0, 0, 0 )
INSERT INTO SYSINDEXES ( 'ISYN', 'SYSTEM', 'SYSSYNONYMS', 'SYSTEM',
                          '+USERID, +ALTNAME', 'U', 'F', 0,
                          <2,1,2>,  -1, -1, -1, 'K',
                          -1, -1, 0, )
INSERT INTO SYSCATALOG ( 'SYSCOLUMNS', 'SYSTEM', 'R', 14,
                        'COMMENT', 1, 'SYS0001', 0, 'L', 0, -1, -1, -1,
                          -1, -1, 0, 0, 0 ).
INSERT INTO SYSINDEXES ( 'ICOL', 'SYSTEM', 'SYSCOLUMNS', 'SYSTEM',
                         '+TNAME, +CREATOR, +CNAME', 'U', 'F', -1,
                         <3,2,3,1>, -1, -1, -1, 'K',
                          -1, -1, 0, )
INSERT INTO SYSLINKS   ( 'COLLINK', 'SYSTEM',
                        'SYSCATALOG', 'SYSTEM', 'SYSCOLUMNS', 'SYSTEM',
                'CREATE LINK COLLINK FROM SYSCATALOG ( TNAME, CREATOR )
 TO SYSCOLUMNS ( TNAME, CREATOR ) ORDER BY COLNO',
                         -1, <2,1,2>, <2,2,3>, <1,4>, 'N', 'U', 'Y',
                         -1, -1, -1, -1, -1, ' ' )
INSERT INTO SYSCATALOG ( 'SYSINDEXES', 'SYSTEM', 'R', 16,
                        'COMMENT', 1, 'SYS0001', 0, 'L', 0, -1, -1, -1,
                          -1, -1, 0, 0, 0 )
INSERT INTO SYSINDEXES  ( 'IINDX', 'SYSTEM', 'SYSINDEXES', 'SYSTEM',
                          '+TNAME, +CREATOR', 'D', 'W', -1,
                          <2,3,4>, -1, -1, -1, 'K',
                          -1, -1, 0, )
INSERT INTO SYSINDEXES  ( 'IINDX2', 'SYSTEM', 'SYSINDEXES', 'SYSTEM',
                          '+INAME, +ICREATOR', 'U', 'N', -1,
                          <2,1,2>, -1, -1, -1, 'K',
                          -1, -1, 0, )
INSERT INTO SYSLINKS   ( 'INDXLINK', 'SYSTEM',
                         'SYSCATALOG', 'SYSTEM', 'SYSINDEXES', 'SYSTEM',
'CREATE LINK INDXLINK FROM SYSCATALOG ( TNAME, CREATOR ) TO SYSINDEXES
 ( TNAME, CREATOR )',
                         -1, <2,1,2>, <2,3,4>, '', 'N', 'N', 'Y',
                         -1, -1, -1, -1, -1, ' ' )
INSERT INTO SYSCATALOG ( 'SYSLINKS', 'SYSTEM', 'R', 20,
                        'COMMENT', 1, 'SYS0001', 0, 'L', 0, -1, -1, -1,
                          -1, -1, 0, 0, 0 )
INSERT INTO SYSINDEXES  ( 'ILINK', 'SYSTEM', 'SYSLINKS', 'SYSTEM',
                          '+LINKNAME, +LCREATOR', 'U', 'W', -1,
                          <2,1,2>, -1, -1, -1, 'K',
                          -1, -1, 0, )
INSERT INTO SYSLINKS   ( 'CLINK', 'SYSTEM',
                         'SYSCATALOG', 'SYSTEM', 'SYSLINKS', 'SYSTEM',
'CREATE LINK CLINK FROM SYSCATALOG ( TNAME, CREATOR ) TO SYSLINKS
 ( CRNAME, CCREATOR )',
                         -1, <2,1,2>, <2,5,6>, '', 'N', 'N', 'Y',
```

```
                        -1, -1, -1, -1, -1, ' ' )
INSERT INTO SYSLINKS   ( 'PLINK', 'SYSTEM',
                         'SYSCATALOG', 'SYSTEM', 'SYSLINKS', 'SYSTEM',
'CREATE LINK PLINK FROM SYSCATALOG ( TNAME, CREATOR ) TO SYSLINKS
 (PRNAME, PCREATOR )',
                         -1, <2,1,2>, <2,3,4>, '', 'N', 'N', 'N',
                         -1, -1, -1, -1, -1, ' ' )
INSERT INTO SYSCATALOG ( 'SYSVIEWS', 'SYSTEM', 'R', 4,
                        'COMMENT', 1, 'SYS0001', 0, 'I', 0, -1, -1, -1,
                         -1, -1, 0, 0, 0 )
INSERT INTO SYSINDEXES  ( 'IVIEWS', 'SYSTEM', 'SYSVIEWS', 'SYSTEM',
                          '+VIEWNAME, +VCREATOR, +SEQNO', 'U', 'F', -1,
                          <3,1,2,3>, -1, -1, -1, 'K',
                          -1, -1, 0, )
INSERT INTO SYSCATALOG ( 'SYSUSAGE', 'SYSTEM', 'R', 7,
                        'COMMENT', 1, 'SYS0001', 0, 'I', 0, -1, -1, -1,
                         -1, -1, 0, 0, 0 )
INSERT INTO SYSINDEXES  ( 'IUSAGE', 'SYSTEM', 'SYSUSAGE', 'SYSTEM',
                          '+BNAME, +BCREATOR, +BTYPE', 'D', 'F', -1,
                          <3,1,2,3>, -1, -1, -1, 'K',
                          -1, -1, 0, )
INSERT INTO SYSINDEXES  ( 'IUSAGE2', 'SYSTEM', 'SYSUSAGE', 'SYSTEM',
                          '+DNAME, +DCREATOR, +DTYPE', 'D', 'N', -1,
                          <3,4,5,6>, -1, -1, -1, 'K',
                          -1, -1, 0, )
INSERT INTO SYSCATALOG ( 'SYSTABAUTH', 'SYSTEM', 'R', 15,
                        'COMMENT', 1, 'SYS0001', 0, 'I', 0, -1, -1, -1,
                         -1, -1, 0, 0, 0 )
INSERT INTO SYSINDEXES ( 'ITABAUTH1', 'SYSTEM', 'SYSTABAUTH', 'SYSTEM',
                         '+GRANTEE, +TCREATOR, +TTNAME, +GRANTEETYPE',
                         'D', 'F', -1,
                         <5,2,6,7,3>, -1, -1, -1, 'K',
                          -1, -1, 0, )
INSERT INTO SYSINDEXES ( 'ITABAUTH2', 'SYSTEM', 'SYSTABAUTH', 'SYSTEM',
                         '+GRANTOR, +SCREATOR, +STNAME', 'D', 'N', -1,
                         <3,1,4,5>, -1, -1, -1, 'K',
                          -1, -1, 0, )
INSERT INTO SYSINDEXES ( 'ITABAUTH3', 'SYSTEM', 'SYSTABAUTH', 'SYSTEM',
               '+GRANTOR, +GRANTEE, +SCREATOR, +STNAME, +GRANTEETYPE',
                         'D', 'N', -1, <5,1,2,4,5,3>, -1, -1, -1, 'K',
                          -1, -1, 0, )
INSERT INTO SYSINDEXES ( 'ITABAUTH4', 'SYSTEM', 'SYSTABAUTH', 'SYSTEM',
                         '+SCREATOR, +STNAME',
                         'D', 'N', -1, <2,4,5>, -1, -1, -1, 'K',
                          -1, -1, 0, )
INSERT INTO SYSINDEXES ( 'ITABAUTH5', 'SYSTEM', 'SYSTABAUTH', 'SYSTEM',
                         '+TCREATOR, +TTNAME',
                         'D', 'N', -1, <2,6,7>, -1, -1, -1, 'K',
                          -1, -1, 0, )
INSERT INTO SYSCATALOG ( 'SYSPROGAUTH', 'SYSTEM', 'R', 6,
                        'COMMENT', 1, 'SYS0001', 0, 'N', 0, -1, -1, -1,
                         -1, -1, 0, 0, 0 )
INSERT INTO SYSINDEXES('IPROGAUTH1', 'SYSTEM', 'SYSPROGAUTH', 'SYSTEM',
                         '+CREATOR, +PROGNAME, +GRANTEE, +RUNAUTH',
```

```
                          'D', 'W', -1, <4,3,4,2,6>, -1, -1, -1, 'K',
                           -1, -1, 0, )
INSERT INTO SYSINDEXES('IPROGAUTH2', 'SYSTEM', 'SYSPROGAUTH', 'SYSTEM',
                          '+GRANTOR, +GRANTEE, +CREATOR, +PROGNAME',
                          'D', 'N', -1, <4,1,2,3,4>, -1, -1, -1, 'K',
                           -1, -1, 0, )
INSERT INTO SYSCATALOG ( 'SYSCOLAUTH', 'SYSTEM', 'R', 6,
                         'COMMENT', 1, 'SYS0001', 0, 'N', 0, -1, -1, -1,
                           -1, -1, 0, 0, 0 )
INSERT INTO SYSINDEXES ( 'ICOLAUTH1', 'SYSTEM', 'SYSCOLAUTH', 'SYSTEM',
                          '+CREATOR, +TNAME, +COLNAME, +GRANTEE',
                          'D', 'W', -1,
                          <4,3,4,6,2>, -1, -1, -1, 'K',
                           -1, -1, 0, )
INSERT INTO SYSINDEXES ( 'ICOLAUTH2', 'SYSTEM', 'SYSCOLAUTH', 'SYSTEM',
                          '+TIMESTAMP, +COLNAME', 'D', 'N', -1,
                          <2,5,6>, -1, -1, -1, 'K',
                           -1, -1, 0, )
INSERT INTO SYSINDEXES ( 'ICOLAUTH3', 'SYSTEM', 'SYSCOLAUTH', 'SYSTEM',
                         '+GRANTEE, +TIMESTAMP, +COLNAME', 'D', 'N', -1,
                          <3,2,5,6>, -1, -1, -1, 'K',
                           -1, -1, 0, )
INSERT INTO SYSCATALOG ( 'SYSUSERAUTH', 'SYSTEM', 'R', 6,
                         'COMMENT', 1, 'SYS0001', 0, 'I', 0, -1, -1, -1,
                           -1, -1, 0, 0, 0 )
INSERT INTO SYSINDEXES('IUSERAUTH', 'SYSTEM', 'SYSUSERAUTH', 'SYSTEM',
                          '+AUTHOR, +NAME', 'U', 'F', -1,
                           <2,2,1>, -1, -1, -1, 'K',
                           -1, -1, 0, )
INSERT INTO SYSCATALOG ( 'SYSACCESS', 'SYSTEM', 'R', 9,
                        'COMMENT', 1, 'SYS0001', 0, 'I', 0, -1, -1, -1,
                           -1, -1, 0, 0, 0 )
INSERT INTO SYSINDEXES ( 'IACCESS', 'SYSTEM', 'SYSACCESS', 'SYSTEM',
                         '+TNAME, +CREATOR, +TABTYPE', 'U', 'W', 0,
                        <3,1,2,9>, -1, -1, -1, 'K',
                           -1, -1, 0, )
INSERT INTO SYSCATALOG ( 'SYSDBSPACES', 'SYSTEM', 'R', 12,
                        'COMMENT', 1, 'SYS0001', 0, 'I', 0, -1, -1, -1,
                           -1, -1, 0, 0, 0 )
INSERT INTO SYSINDEXES ( 'IDBSPACE', 'SYSTEM', 'SYSDBSPACES', 'SYSTEM',
                          '+OWNER, +DBSPACENAME', 'D', 'F', 0,
                          <2,1,3>, -1, -1, -1, 'K',
                           -1, -1, 0, )
INSERT INTO SYSINDEXES( 'IDBSPACE2', 'SYSTEM', 'SYSDBSPACES', 'SYSTEM',
                          '+DBSPACETYPE, +OWNER, +NPAGES', 'D', 'N', 0,
                          <3,4,3,6>, -1, -1, -1, 'K',
                           -1, -1, 0, )
INSERT INTO SYSINDEXES( 'IDBSPACE3', 'SYSTEM', 'SYSDBSPACES', 'SYSTEM',
                          '+DBSPACENO', 'U', 'N', 0,
                          <1,2>, -1, -1, -1, 'K',
                           -1, -1, 0, )
INSERT INTO SYSCATALOG ( 'SYSDROP', 'SYSTEM', 'R', 3,
                         'COMMENT', 1, 'SYS0001', 0, 'N', 0, -1, -1, -1,
```

```
                          -1, -1, 0, 0, 0 )
INSERT INTO SYSVIEWS ( 'ZZZZZZZZ', 'DUMMY   ', 1, ' ' )
INSERT INTO SYSUSAGE( 'ZZZZZZZZ', 'DUMMY   ', ' ', ' ', ' ', ' ', ' ' )
INSERT INTO SYSTAB ( 'SYSTEM', 'SYSTEM', ' ', 'SYSTEM', 'SYSCATALOG'
                          'SYSTEM', 'SYSCATALOG', 'AAAAAAAAAAAA', ' ',
                          'G', ' ', ' ', ' ', ' ', ' ' )
INSERT INTO SYSTAB ('SYSTEM', 'SYSTEM', ' ', 'SYSTEM', 'SYSCOLUMNS',
                          'SYSTEM', 'SYSCOLUMNS', 'AAAAAAAAAAAA', ' ',
                          'G', ' ', ' ', ' ', ' ', ' ' )
INSERT INTO SYSTAB ( 'SYSTEM', 'SYSTEM', ' ', 'SYSTEM', 'SYSSYNONYMS',
                          'SYSTEM', 'SYSSYNONYMS', 'AAAAAAAAAAAA', ' ',
                          'G', ' ', ' ', ' ', ' ', ' ' )
INSERT INTO SYSTAB ( 'SYSTEM', 'SYSTEM', ' ', 'SYSTEM', 'SYSINDEXES',
                          'SYSTEM', 'SYSINDEXES', 'AAAAAAAAAAAA', ' ',
                          'G', ' ', ' ', ' ', ' ', ' ' )
INSERT INTO SYSTAB ( 'SYSTEM', 'SYSTEM', ' ', 'SYSTEM', 'SYSVIEWS',
                          'SYSTEM', 'SYSVIEWS', 'AAAAAAAAAAAA', ' ',
                          'G', ' ', ' ', ' ', ' ', ' ' )
INSERT INTO SYSTAB ( 'SYSTEM', 'SYSTEM', ' ', 'SYSTEM', 'SYSUSAGE',
                          'SYSTEM', 'SYSUSAGE', 'AAAAAAAAAAAA', ' ',
                          'G', ' ', ' ', ' ', ' ', ' ' )
INSERT INTO SYSTAB ( 'SYSTEM', 'SYSTEM', ' ', 'SYSTEM', 'SYSTABAUTH',
                          'SYSTEM', 'SYSTABAUTH', 'AAAAAAAAAAAA', ' ',
                          'G', ' ', ' ', ' ', ' ', ' ' )
INSERT INTO SYSTAB ( 'SYSTEM', 'SYSTEM', ' ', 'SYSTEM', 'SYSCOLAUTH',
                          'SYSTEM', 'SYSCOLAUTH', 'AAAAAAAAAAAA', ' ',
                          'G', ' ', ' ', ' ', ' ', ' ' )
INSERT INTO SYSTAB ( 'SYSTEM', 'SYSTEM', ' ', 'SYSTEM', 'SYSPROGAUTH',
                          'SYSTEM', 'SYSPROGAUTH', 'AAAAAAAAAAAA', ' ',
                          'G', ' ', ' ', ' ', ' ', ' ' )
INSERT INTO SYSTAB ( 'SYSTEM', 'SYSTEM', ' ', 'SYSTEM', 'SYSACCESS',
                          'SYSTEM', 'SYSACCESS', 'AAAAAAAAAAAA', ' ',
                          'G', ' ', ' ', ' ', ' ', ' ' )
INSERT INTO SYSTAB ( 'SYSTEM', 'SYSTEM', ' ', 'SYSTEM', 'SYSDBSPACES',
                         'SYSTEM', 'SYSDBSPACES', 'AAAAAAAAAAAA', ' ',
                          'G', ' ', ' ', ' ', ' ', ' ' )
INSERT INTO SYSTAB ( 'SYSTEM', 'SYSTEM', ' ', 'SYSTEM', 'SYSDROP',
                          'SYSTEM', 'SYSDROP', 'AAAAAAAAAAAA', ' ',
                          'G', ' ', ' ', ' ', ' ', ' ' )
INSERT INTO SYSTAB ( 'SYSTEM', 'SQLDBA', ' ', 'SYSTEM', 'SYSCATALOG'
                          'SYSTEM', 'SYSCATALOG', 'AAAAAAAAAAAA', ' ',
                          'G', ' ', ' ', ' ', ' ', ' ' )
INSERT INTO SYSTAB ('SYSTEM', 'SQLDBA', ' ', 'SYSTEM', 'SYSCOLUMNS',
                          'SYSTEM', 'SYSCOLUMNS', 'AAAAAAAAAAAA', ' ',
                          'G', ' ', ' ', ' ', ' ', ' ' )
INSERT INTO SYSTAB ( 'SYSTEM', 'SQLDBA', ' ', 'SYSTEM', 'SYSSYNONYMS',
                          'SYSTEM', 'SYSSYNONYMS', 'AAAAAAAAAAAA', ' ',
                          'G', ' ', ' ', ' ', ' ', ' ' )
INSERT INTO SYSTAB ( 'SYSTEM', 'SQLDBA', ' ', 'SYSTEM', 'SYSINDEXES',
                          'SYSTEM', 'SYSINDEXES', 'AAAAAAAAAAAA', ' ',
                          'G', ' ', ' ', ' ', ' ', ' ' )
INSERT INTO SYSTAB ( 'SYSTEM', 'SQLDBA', ' ', 'SYSTEM', 'SYSVIEWS',
                          'SYSTEM', 'SYSVIEWS', 'AAAAAAAAAAAA', ' ',
```

```
                          'G', ' ', ' ', ' ', ' ', ' ' )
INSERT INTO SYSTAB ( 'SYSTEM', 'SQLDBA', ' ', 'SYSTEM', 'SYSUSAGE',
                          'SYSTEM', 'SYSUSAGE', 'AAAAAAAAAAAA', ' ',
                          'G', ' ', ' ', ' ', ' ', ' ' )
INSERT INTO SYSTAB ( 'SYSTEM', 'SQLDBA', ' ', 'SYSTEM', 'SYSTABAUTH',
                          'SYSTEM', 'SYSTABAUTH', 'AAAAAAAAAAAA', ' ',
                          'G', ' ', ' ', ' ', ' ', ' ' )
INSERT INTO SYSTAB ( 'SYSTEM', 'SQLDBA', ' ', 'SYSTEM', 'SYSCOLAUTH',
                          'SYSTEM', 'SYSCOLAUTH', 'AAAAAAAAAAAA', ' ',
                          'G', ' ', ' ', ' ', ' ', ' ' )
INSERT INTO SYSTAB ( 'SYSTEM', 'SQLDBA', ' ', 'SYSTEM', 'SYSPROGAUTH',
                          'SYSTEM', 'SYSPROGAUTH', 'AAAAAAAAAAAA', ' ',
                          'G', ' ', ' ', ' ', ' ', ' ' )
INSERT INTO SYSTAB ( 'SYSTEM', 'SQLDBA', ' ', 'SYSTEM', 'SYSUSERAUTH',
                          'SYSTEM', 'SYSUSERAUTH', 'AAAAAAAAAAAA', ' ',
                          'G', ' ', ' ', ' ', ' ', ' ' )
INSERT INTO SYSTAB ( 'SYSTEM', 'SQLDBA', ' ', 'SYSTEM', 'SYSACCESS',
                          'SYSTEM', 'SYSACCESS', 'AAAAAAAAAAAA', ' ',
                          'G', ' ', ' ', ' ', ' ', ' ' )
INSERT INTO SYSTAB ( 'SYSTEM', 'SQLDBA', ' ', 'SYSTEM', 'SYSDBSPACES',
                         'SYSTEM', 'SYSDBSPACES', 'AAAAAAAAAAAA', ' ',
                          'G', ' ', ' ', ' ', ' ', ' ' )
INSERT INTO SYSTAB ( 'SYSTEM', 'SQLDBA', ' ', 'SYSTEM', 'SYSDROP',
                          'SYSTEM', 'SYSDROP', 'AAAAAAAAAAAA', ' ',
                          'G', ' ', ' ', ' ', ' ', ' ' )
INSERT INTO SYSCOLAUTH  ( ' ', 'ZZZDUMMY', ' ', 'DUMMY ROW', ' ', ' ' )
INSERT INTO SYSPROGAUTH ( ' ', 'ZZZDUMMY', ' ', 'DUMMY ROW', ' ', ' ' )
INSERT INTO SYSUSERAUTH ( 'SQLDBA', ' ', 'Y', 'Y', 'SQLDBAPW', 'Y' )
INSERT INTO SYSACCESS ( 'ZZZZZZZZ', 'DUMMY   ', 0, 0, 0, 0, ' ', ' ',
    ' ' )
INSERT INTO SYSSYNONYMS ( 'ZZZZZZZZ', 'DUMMY ROW', ' ', ' ' )
UPDATE SYSCATALOG
SET    CLUSTERROW = TID OF SYSLINKS ( COLLINK )
WHERE  TNAME = SYSCOLUMNS
UPDATE SYSCATALOG
SET    CLUSTERROW = TID OF SYSLINKS ( CLINK )
WHERE  TNAME = SYSLINKS
```

APPENDIX A: ABBREVIATIONS AND ACRONYMS

ACB	VSE/VSAM Access Control Block
ASL	Access Specification Language
AUX	Access Module (stored in the data base)
BNF	Backus Normal Form
BRM	Batch Resource Manager (VSE)
CICS	Customer Information Control System/VS
CPC	Cross-Partition Communication (also XPCC)
DBCS	Double-Byte Character Set
DBS(U)	Data Base Services (Utility)
DBSI	Data Base Storage Linkage
DBSPACE	Data Base Space
DBSS	Data Base Storage System
DBSS/DC	The DBSS Data Control subcomponent
DCE	SQL/DS Dispatcher Control Element
DCR	Domain Control Record
DCSS	Discontiguous Shared Segment (VM)
DNF	Disjunctive Normal Form
DSC	SQL/DS Data System Control
ECB	Event Control Block
EDSF	Extended Dynamic Statement Facility
ICCF	Interactive Computing and Control Facility (VSE)
ICR	Index Control Record
IID	Index ID
ISQL	Interactive Structured Query Language
IUCV	Inter-User Communication Vehicle (VM)
LCR	Link Control Record
LID	Link ID
LRB	DBSS Lock Request Block
LUW	Logical Unit of Work
MCR	Master Control Record
MPM	Multi-Partition Mode (VSE)
MUM	Multiple User Mode
MVMM	Multiple Virtual Machine Mode (VM)
ORM	Online Resource Manager (VSE)
PDS	Parameter Data Set
PLCR	Parent Link Control Record
RDS	Relational Data System
RID	Relation (table) ID
RM	Resource Manager
RMGR	Resource Manager
RPL	VSE/VSAM Request Parameter List
SCB	Scan Control Block
SCR	Segment (DBSPACE) Control Record
SPM	Single-Partition Mode (VSE)
SQL	Structured Query Language
SQL/DS	Structured Query Language / Data System
SUM	Single User Mode
SVMM	Single Virtual Machine Mode (VM)
TCI	Table Column Identification
TID	Tuple (row) ID
VRM	VM Resource Manager
WSG	Working Storage Area (or Manager)
XPCC	Cross-Partition Communication (also CPC)

APPENDIX B: DICTIONARY OF EQUIVALENT TERMS

Some of the terminology in this manual will not match the terminology seen in the microfiche code (and terminology may also vary within the microfiche code itself). This "Dictionary of Equivalent Terms" lists the terms with their various equivalent names.

BASE	RSIBASE, DBSS Linkage Structure
Block Disk	Directory (data set)
Block 1	Space Block
Block 0 (see Op Tree)	
column	domain, field
data disk (see DBEXTENT)	
data set (flat file) (see table)	
DBEXTENT	data disk, EXTENT
DBSI	PLIRSS (ARIYM00), RSI, SI
DBSPACE	segment
DBSS	RSS
Directory (data set)	Block Disk
domain (see column)	
DS/2 (see SQL/DS)	
EXTENT (see DBEXTENT)	
field (see column)	
image	index
index	image
LUW (Logical Unit of Work)	transaction
OP (see Op Tree)	
Op Area (see Op Tree)	
Op Tree	Block 0, OP, Op Area, Parse Tree, PTREE
Parse Tree (see Op Tree)	
PLIRSS (ARIYM00) (see DBSI)	
PTREE (see Op Tree)	
record (see row)	
relation (see table)	
row	record, tuple
RSI (see DBSI)	
RSIBASE (see BASE)	
RSS	DBSS
segment	DBSPACE
SI (see DBSI)	
Space Block	Block 1
SQL/DS	DS/2, System R
System R (see SQL/DS)	
table	data set (flat file), relation
tuple (see row)	
transaction	LUW (Logical Unit of Work)
UNLOAD	UPDATE STATISTICS
UPDATE STATISTICS	UNLOAD

A

B

C

D

E

F

G

H

M

N

O

P

Q

R

S

T

U

V

W

Y

SQL/Data System Logic
Volume 2
LY24-5217-2

READER'S COMMENT FORM

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Your comments will be sent to the author's department for whatever review and action, if any, are deemed appropriate. Comments may be written in your own language; English is not required.

Note: *Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.*

Note: Staples can cause problems with automated mail sorting equipment. Please use pressure sensitive or other gummed tape to seal this form.

	Yes	*No*
◦ Does the publication meet your needs?	☐	☐
◦ Did you find the material:		
Easy to read and understand?	☐	☐
Organized for convenient use?	☐	☐
Complete?	☐	☐
Well illustrated?	☐	☐
Written for your technical level?	☐	☐

◦ What is your occupation? _____

◦ How do you use this publication:

As an introduction to the subject?	☐	As an instructor in class?	☐
For advanced knowledge of the subject?	☐	As a student in class?	☐
To learn about operating procedures?	☐	As a reference manual?	☐

Your comments:

If you would like a reply, please supply your name and address on the reverse side of this form.

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail directly to the address in the Edition Notice on the back of the title page.)

Reader's Comment Form

Fold and Tape | Please Do Not Staple | Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE:

International Business Machines Corporation
Department G60
P. O. Box 6
Endicott, New York 13760

Fold | Fold

If you would like a reply, *please print:*

Your Name _____

Company Name _____ *Department* _____

Street Address _____

City _____

State _____ *Zip Code* _____

IBM Branch Office serving you _____

IBM®

Cut or Fold Along Line

SQL/Data System Logic, Volume 2 (File No. S370/4300-50) Printed in U.S.A. LY24-5217-2

SQL/Data System Logic
Volume 2
LY24-5217-2

READER'S
COMMENT
FORM

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Your comments will be sent to the author's department for whatever review and action, if any, are deemed appropriate. Comments may be written in your own language; English is not required.

Note: *Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.*

Note: Staples can cause problems with automated mail sorting equipment.
Please use pressure sensitive or other gummed tape to seal this form.

	Yes	*No*
◦ Does the publication meet your needs?	☐	☐
◦ Did you find the material:		
Easy to read and understand?	☐	☐
Organized for convenient use?	☐	☐
Complete?	☐	☐
Well illustrated?	☐	☐
Written for your technical level?	☐	☐

◦ What is your occupation? _____

◦ How do you use this publication:

As an introduction to the subject?	☐	As an instructor in class?	☐
For advanced knowledge of the subject?	☐	As a student in class?	☐
To learn about operating procedures?	☐	As a reference manual?	☐

Your comments:

If you would like a reply, please supply your name and address on the reverse side of this form.

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail directly to the address in the Edition Notice on the back of the title page.)

LY24-5217-2

Reader's Comment Form

Cut or Fold Along Line

Fold and Tape | Please Do Not Staple | Fold and Tape

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE:

International Business Machines Corporation
Department G60
P. O. Box 6
Endicott, New York 13760

Fold | Fold

If you would like a reply, *please print:*

Your Name _____

Company Name _____ *Department* _____

Street Address _____

City _____

State _____ *Zip Code* _____

IBM Branch Office serving you _____

IBM®

SQL/Data System Logic, Volume 2 (File No. S370/4300-50) Printed in U.S.A. LY24-5217-2

SQL/Data System Logic
Volume 2
LY24-5217-2

READER'S COMMENT FORM

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Your comments will be sent to the author's department for whatever review and action, if any, are deemed appropriate. Comments may be written in your own language; English is not required.

Note: *Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.*

Note: Staples can cause problems with automated mail sorting equipment. Please use pressure sensitive or other gummed tape to seal this form.

	Yes	*No*
• Does the publication meet your needs?	☐	☐
• Did you find the material:		
Easy to read and understand?	☐	☐
Organized for convenient use?	☐	☐
Complete?	☐	☐
Well illustrated?	☐	☐
Written for your technical level?	☐	☐

• What is your occupation? _____

• How do you use this publication:

As an introduction to the subject?	☐	As an instructor in class?	☐
For advanced knowledge of the subject?	☐	As a student in class?	☐
To learn about operating procedures?	☐	As a reference manual?	☐

Your comments:

If you would like a reply, please supply your name and address on the reverse side of this form.

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail directly to the address in the Edition Notice on the back of the title page.)

Reader's Comment Form

Cut or Fold Along Line

SQL/Data System Logic, Volume 2 (File No. S370/4300-50) Printed in U.S.A. LY24-5217-2

Fold and Tape Please Do Not Staple Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE:

International Business Machines Corporation
Department G60
P. O. Box 6
Endicott, New York 13760

Fold Fold

If you would like a reply, *please print:*

Your Name _____

Company Name _____ *Department* _____

Street Address _____

City _____

State _____ *Zip Code* _____

IBM Branch Office serving you _____

IBM®

SQL/Data System Logic, Volume 2 (File No. S370/4300-50) Printed in U.S.A. LY24-5217-2