The In's and Out's of DFSMSdss Exit Processing and Patch Options

Robert Gensler
IBM
rgensle@us.ibm.com

Session 16132
August 5, 2014

www.SHARE.org

#SHAREorg

Legal Disclaimer

NOTICES AND DISCLAIMERS

Product information and data has been reviewed for accuracy as of the date of initial publication. Product information and data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements and/or changes in the product(s) and/or programs(s) described herein at any time without notice.

References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Consult your local IBM representative or IBM Business Partner for information about the product and services available in your area.

Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectually property rights, may be used instead. It is the user's responsibility to evaluate and verify the operation of any non-IBM product, program or service.

THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted according to the terms and conditions of the agreements (e.g., IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. IBM is not responsible for the performance or interoperability of any non-IBM products discussed herein.

Trademarks

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries.

BookManager*
CICS*
DB2*
DB2 Universal Database
developerWorks*
DFSMSdfp
DFSMSdss
DFSMShsm
DFSMSrmm
DFSORT
Domino
Enterprise Storage Server*
ES/9000*
FlashCopy*
GDPS*
HiperSockets
IBM*
IBM eServer
IBM e(logo)server*
IBM logo*
IMS
InfoPrint*
IP PrintWay
Language Environment*
Lotus*
Multiprise*
MVS
Notes*
OS/390*
Parallel Sysplex*
RACF*
RAMAC*
RMF
S/370
S/390*
Tivoli*
TotalStorage*
WebSphere*
z/Architecture
z/OS*
zSeries*

Intel is a trademark of the Intel Corporation in the United States and other countries.
Java and all Java-related trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc., in the United States and other countries.
Microsoft, Windows and Windows NT are registered trademarks of Microsoft Corporation.
UNIX is a registered trademark of The Open Group in the United States and other countries.

* All other products may be trademarks or registered trademarks of their respective companies.

Notes:

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

This presentation and the claims outlined in it were reviewed for compliance with US law. Adaptations of these claims for use in other geographies must be reviewed by the local country counsel for compliance with local laws.

Agenda

- Controlling DFSMSdss
  - Customizing DFSMSdss using Patches
  - Zapping the ADRPATCH area
- Patches of Interest
- Installation Exits
  - Dynamic Exit (new in V2R1)
- Application Programming Interface
  - Calling Block Structure
  - Cross Memory API

Zapping the ADRPATCH area

- Allows users to customize DFSMSdss functions
  - Dynamic vs. Permanent
  - Most patches can be set both permanently and dynamically
    - Few patches can **only** be set dynamically
  - AMASPZAP – Permanent
  - SET PATCH – Dynamic
- Can be protected using RACF facility class profile
  - STGADMIN.ADR.PATCH
    - *Only applies to SET PATCH*

Permanently Patching DFSMSdss

```
//PATCH    JOB...
//*
//*************************************************************
//*                                                           *
//* SAMPLE JCL TO SET THE FLAGS IN ADRPATCH.                  *
//*                                                           *
//*************************************************************
//ZAP      EXEC PGM=AMASPZAP,PARM='IGNIDRFULL'
//SYSPRINT DD   SYSOUT=*
//SYSLIB   DD   DISP=SHR,DSN=LIBNAME.LINKLIB
//SYSIN DD *
   NAME ADRDSSU ADRPATCH
   VER   offset    value  REP   offset    value

/*
```

This should point to the library containing the ADRDSSU load module

These <offset, value> pairs are specified in hexadecimal format

Dynamically Patching DFSMSdss

```
//STEPT006 EXEC PGM=ADRDSSU,PARM='UTILMSG=YES'
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
  SET PATCH 5B=FF                    /* enable source eattr override    */
  COPY                               /* move data set to EAV            */ -
    DS(INCL(PATCHX5B.MIG.EAV))       /* data set to be moved            */ -
    FR(PREF)                         /* prefer fast replication         */ -
    FCTOPPRCPRIMARY(PMR)             /* ensure mirror stays full duplex */ -
    DEBUG(FRMSG(DTL))                /* detailed fast replication msgs  */ -
    STORCLAS(SCMIXTGT)               /* target storage class            */ -
    ADMIN                            /* bypass authorization checking   */ -
    DELETE                           /* delete source when finished     */
  SET PATCH 5B=00                    /* disable source eattr override   */
/*
```

Output in SYSPRINT

```
ADR109I (R/I)-RI01 (01), 2014.207 07:18:19 INITIAL SCAN OF USER CONTROL STATEMENTS COMPLETED
ADR113I (R/I)-RI01 (01), PATCH BYTE AT OFFSET 005B = FF
ADR016I (002)-PRIME(01), RACF LOGGING OPTION IN EFFECT FOR THIS TASK
ADR006I (002)-STEND(01), 2014.207 07:18:20 EXECUTION BEGINS
ADR711I (002)-NEWDS(01), DATA SET PATCHX5B.MIG.EAV HAS BEEN ALLOCATED USING STORCLAS SCMIXTGT, DATAC
ADR806I (002)-TOMI (01), DATA SET PATCHX5B.MIG.EAV COPIED USING A FAST REPLICATION FUNCTION
ADR431I (002)-CNVSM(02), DATA SET PATCHX5B.MIG.EAV IN CATALOG SYS1.MVSRES9.MASTCAT HAS BEEN DELETED
ADR801I (002)-DDDS (01), 2014.207 07:18:20 DATA SET FILTERING IS COMPLETE. 1 OF 1 DATA SETS WERE SEL
                         AND 0 FAILED FOR OTHER REASONS
ADR454I (002)-DDDS (02), THE FOLLOWING DATA SETS WERE SUCCESSFULLY PROCESSED
                          PATCHX5B.MIG.EAV
ADR006I (002)-STEND(02), 2014.207 07:18:20 EXECUTION ENDS
ADR013I (002)-CLTSK(01), 2014.207 07:18:20 TASK COMPLETED WITH RETURN CODE 0000
ADR113I (R/I)-RI01 (01), PATCH BYTE AT OFFSET 005B = 00
ADR012I (SCH)-DSSU (01), 2014.207 07:18:20                                    ODE IS 0000
```

ADR113I provides acknowledgment that the SET command was received

Output at the Console

ADR111I issued at the console

Agenda

- Controlling DFSMSdss
  - Customizing DFSMSdss using Patches
  - Zapping the ADRPATCH area
- Patches of Interest
- Installation Exits
  - Dynamic Exit (new in V2R1)
- Application Programming Interface
  - Calling Block Structure
  - Cross Memory API

Patches of Interest

- Suppose you want to migrate your nonVSAM data sets defined with EATTR = NO to an EAV
  - And make them eligible to be created in the EAS

- Override source EATTR to OPT for nonVSAM data sets during logical COPY (OA42848)
  - VER 5B 00
  - REP 5B FF

- UA70890-V1R12, UA70891-V1R13, UA70892-V2R1

Moving nonVSAM data set to EAV

```
//STEPT006 EXEC PGM=ADRDSSU,PARM='UTILMSG=YES'
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
  SET PATCH 5B=FF                    /* enable source eattr override    */
  COPY                               /* move data set to EAV            */ -
    DS(INCL(PATCHX5B.MIG.EAV))       /* data set to be moved            */ -
    FR(PREF)                         /* prefer fast replication         */ -
    FCTOPPRCPRIMARY(PMR)             /* ensure mirror stays full duplex */ -
    DEBUG(FRMSG(DTL))                /* detailed fast replication msgs  */ -
    STORCLAS(SCMIXTGT)               /* target storage class            */ -
    ADMIN                            /* bypass authorization checking   */ -
    DELETE                           /* delete source when finished     */
  SET PATCH 5B=00                    /* disable source eattr override   */
/*
```

Moving nonVSAM data set to EAV

Patches of Interest

- Resetting the data set changed indicator during physical full volume or tracks RESTORE
  - VER 52 00
    - DFSMSdss turns off the DS1DSCHA bit during a physical full volume or tracks RESTORE operation
  - REP 52 FF
    - DFSMSdss does not reset the DS1DSCHA indicator during a physical full volume or tracks RESTORE operation

- When migrating to V2R1 you can eliminate the use of this patch
  - New RESET keyword on RESTORE

Patches of Interest

- Enabling Catalog Search Interface
  - VER 54 00
    - DFSMSdss uses the traditional generic locate to generate a list of data sets to process
  - REP 54 11
    - DFSMSdss uses the CSI to convert generic filter criteria into a fully qualified list of data sets
- Individual results may vary

Patches of Interest

- Suppose you want to want to get some timing information with regard to a DFSMSdss job
  - Add timestamps to messages (V1R13)
    - X'00' Specific messages
    - X'80' Informational messages
    - X'40' Warning messages
    - X'20' Error messages
    - X'10' Terminating messages
- The values may be added together to get combinations of messages.
  - if you want timestamps on warning and error messages, set the value to X'60'.

Agenda

- Controlling DFSMSdss
  - Customizing DFSMSdss using Patches
  - Zapping the ADRPATCH area
- Patches of Interest
- Installation Exits
  - Dynamic Exit (new in V2R1)
- Application Programming Interface
  - Calling Block Structure
  - Cross Memory API

Installation Exits

- Intended for use by system programmers
  - DFSMSdss ships dummy exits that are linked into the ADRDSSU load module
- As a system programmer you can write your own and link them into ADRDSSU
- Permanent patches affect all invocations of DFSMSdss
  - Not just batch, but application invocations as well (DB2, CICS, IMS, HSM, etc.)
- Calling Environment
  - AMODE 31, Key 8, problem state

Authorization Installation Exit (ADRUPSWD)

- Enables system programmer to control the level of authorization checking done by DFSMSdss
  - BUILDSA, CONSOLIDATE, DEFRAG
    - Called at the volume level only
  - COMPRESS, COPY, DUMP, RESTORE, PRINT, RELEASE
    - Called at the volume level and the data set level
  - CGCREATED, CONVERTV, COPYDUMP
    - Not called

Authorization Installation Exit (ADRUPSWD)

Return Code	End Processing for Volume?	Data Set Level Checks Required?	Should ADRUPSWD be Entered at Data Set Level?
0	No	No	No
8	No	Yes	Yes
12	No	Yes	No
20	Yes	-	-

Authorization Installation Exit (ADRUPSWD)

Return Code	End Processing for Volume?	End Processing of Data Set?	Perform Authorization Checking for Data Set?	Should ADRUPSWD be Entered Again?
0	No	No	No	No
4	No	No	No	Yes
8	No	No	Yes	Yes
12	No	No	Yes	No
16	No	Yes	-	Yes
20	Yes	-	-	-

ADRUPSWD considerations

- Does not apply to facility class profiles
  - Only volume and data set authorization checks
- Not called if PPT statement for ADRDSSU is added to SCHEDxx and NOPASS is specified
- Not called when DASDVOL facility is active and user has proper authority to DASDVOL profile
  - Note that DFSMSdss does not perform DASDVOL authorization checks during logical data set operations of SMS managed data sets

Enqueue Installation Exit (ADRUENQ)

- Enables system programmer to request DFSMSdss perform a 'short VTOC enqueue'
  - physical FULL and TRACKS versions of COPY and DUMP
  - physical data set DUMP
  - PRINT TRACKS
- Consider the tradeoffs
  - Integrity vs. Availability
    - Create, Extend, Scratch

Enqueue Installation Exit (ADRUENQ)

Return Code	Description
0	Enqueue on the volume being dumped or copied for the duration of the operation.
4	Enqueue on the volume being dumped or copied only for the duration of VTOC access.

Options Installation Exit Routine (ADRUIXIT)

- Enable system programmer to override PARMs, keywords...and more!
  - There exist options in this exit that do not correspond to any external parm, keyword, or patch
- Called when DFSMSdss is invoked and before processing of each task
  - First call provides the programmer the PARM statements specified on the invocation of DFSMSdss
  - Second, and subsequent, calls provide the programmer information about the command that DFSMSdss is about to execute

Bits of interest in ADRUIXIT

- UFOHDR
  - UFOBYFCK - Bypass Facility Class Checking
  - UFFREWCL – Rewind on close
  - UFBYFRVF – Bypass Checking for existing FlashCopy relationships during FastReverseRestore
- UFOFUNCT
  - UFOERASE – Erase DASD tracks
  - UFOIACPY - Dump must produce all output copies or none at all
  - UFORACLG – Force RACF Logging

Options Installation Exit Routine (ADRUIXIT)

Return Code	Description
0	No changes were made by this exit routine.
4	The parameter list has been modified.
8	Do not schedule this function (valid only for function command entry).

Reblock Installation Exit Routine (ADRREBLK)

- Enables system programmer to verify and change block sizes of sequential and partitioned data sets
- Called when either of the following are true:
    - Options installation exit indicates that the reblock exit should be called for every data set (UFORBLK)
        - even if it does not match the filtering criteria of the REBLOCK keyword
    - Data set being processed matches the filtering criteria of the REBLOCK keyword, and the options installation exit did not indicate reblocking should not be performed for any data set

*Reblocking exit ignores request to reblock zEDC data sets in order to avoid decompression and recompression of data sets during COPY and RESTORE

Reblock Installation Exit Routine (ADRREBLK)

Return Code	Description
0	Block size not changed. Use the DFSMSdss-selected block size.
4	Block size has been changed by the exit; the new block size is indicated in last 2 bytes of word 6 of the parameter list.
8	Use the input block size (do not reblock).
12	System determined block size is used. In addition, the reblockable indicator in the Format 1 DSCB (DS1REBLK) is turned on.

Data Set Notification Exit – ADRDYEXT_EXIT1

- Applicable when DFSMSdss is invoked by an application program
  - Called during logical data set COPY when application sets EI22DSSRL=EISRLEXIT in the bypass verification exit (exit 22)
  - DFSMSdss could not obtain serialization on the input data set
- Application can write custom exit to close a data set and have DFSMSdss retry serialization
  - Called after the data set is copied to have the data set reopened

Data Set Notification Exit - ADRDYEXT_EXIT1

- ADRDYEXT_EXIT1 is defined during IPL
  - Connects a dummy exit ADRDYX01

See ADRDEX01 in SAMPLIB

- User provided exit routines need to be connected the DFSMSdss dynamic exit, ADRDYEXT_EXIT1, in order to be called by the system.
  - CSVDYNEX REQUEST(ADD) from programs
  - SETPROG EXIT operator command
    - *SETPROG EXIT,ADD,EXITNAME=ADRDYEXT_EXIT1,MODNAME=userext1*
  - EXIT statement in PROGxx parmlib
    - *EXIT ADD EXITNAME(ADRDYEXT_EXIT1) MODNAME(userext1)*

Data Set Notification Exit - ADRDYEXT_EXIT1

- Displaying the exits connected to the DFSMSdss dynamic exit
  - d prog,exit,exitname=adrdyext_exit1

```
00- 09.48.29 SYSTEM1           d prog,exit,exitname=adrdyext_exit1
    09.48.29 SYSTEM1           CSV461I 09.48.29 PROG,EXIT DISPLAY 828       C
    EXIT             MODULE   STATE MODULE   STATE MODULE   STATE
    ADRDYEXT_EXIT1   ADRDYX01   A
```

Agenda

- Controlling DFSMSdss
  - Customizing DFSMSdss using Patches
  - Zapping the ADRPATCH area
- Patches of Interest
- Installation Exits
  - Dynamic Exit (new in V2R1)
- Application Programming Interface
  - Calling Block Structure
  - Cross Memory API

Application Programming Interface

- A programmable interface provided for developers to build applications using the base functions of DFSMSdss
  - Leverage the unique qualities of DFSMSdss and focus on the features important to you and your application
    - *Filtering capabilities*
    - *Knowledge of data set types*
    - *Serialization and integrity*

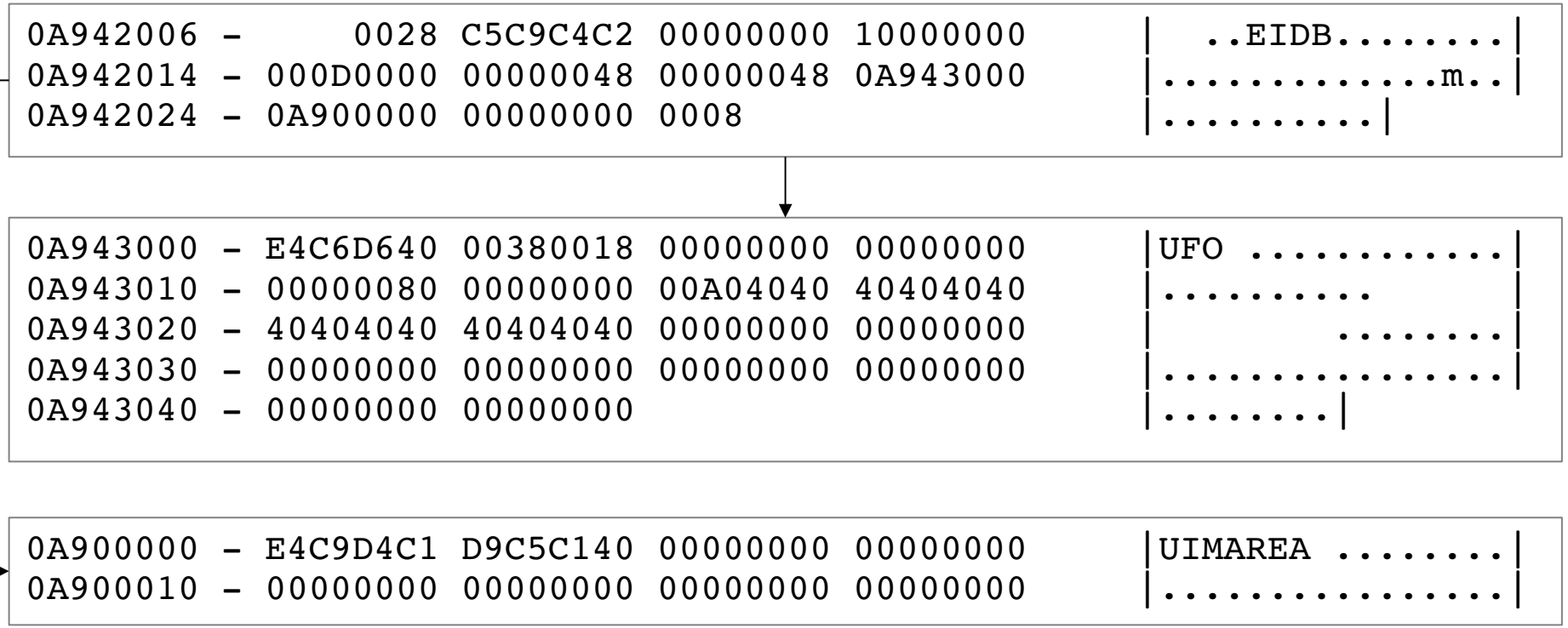
- From within a program you would call, link or attach ADRDSSU
  - Optionally ADRXMAIA for cross memory application interface

User Interaction Module

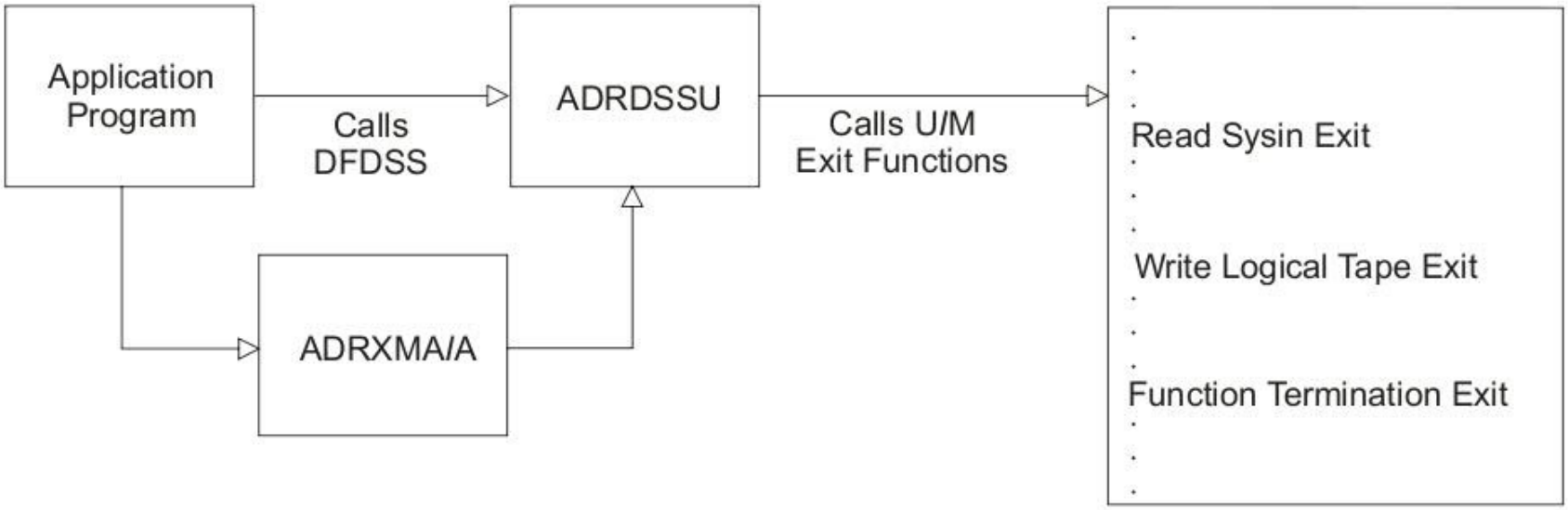
- A User Interaction Module (UIM) is a CSECT provided by the application for DFSMSdss to interact with at various points in processing
    - called exit points or exits

- Upon entry REG1 contains a parameter list with a header by the name of ADREIB
    - Mapped by ADREID0 in SYS1.MACLIB
    - See ADRUFO mapping in DFSMS Installation Exits for Exit 13 (Presenting UFO Record)

Exit 13 - Presenting the ADRUFO record



DFSMSdss calling a User Interaction Module



Application Programming Interface

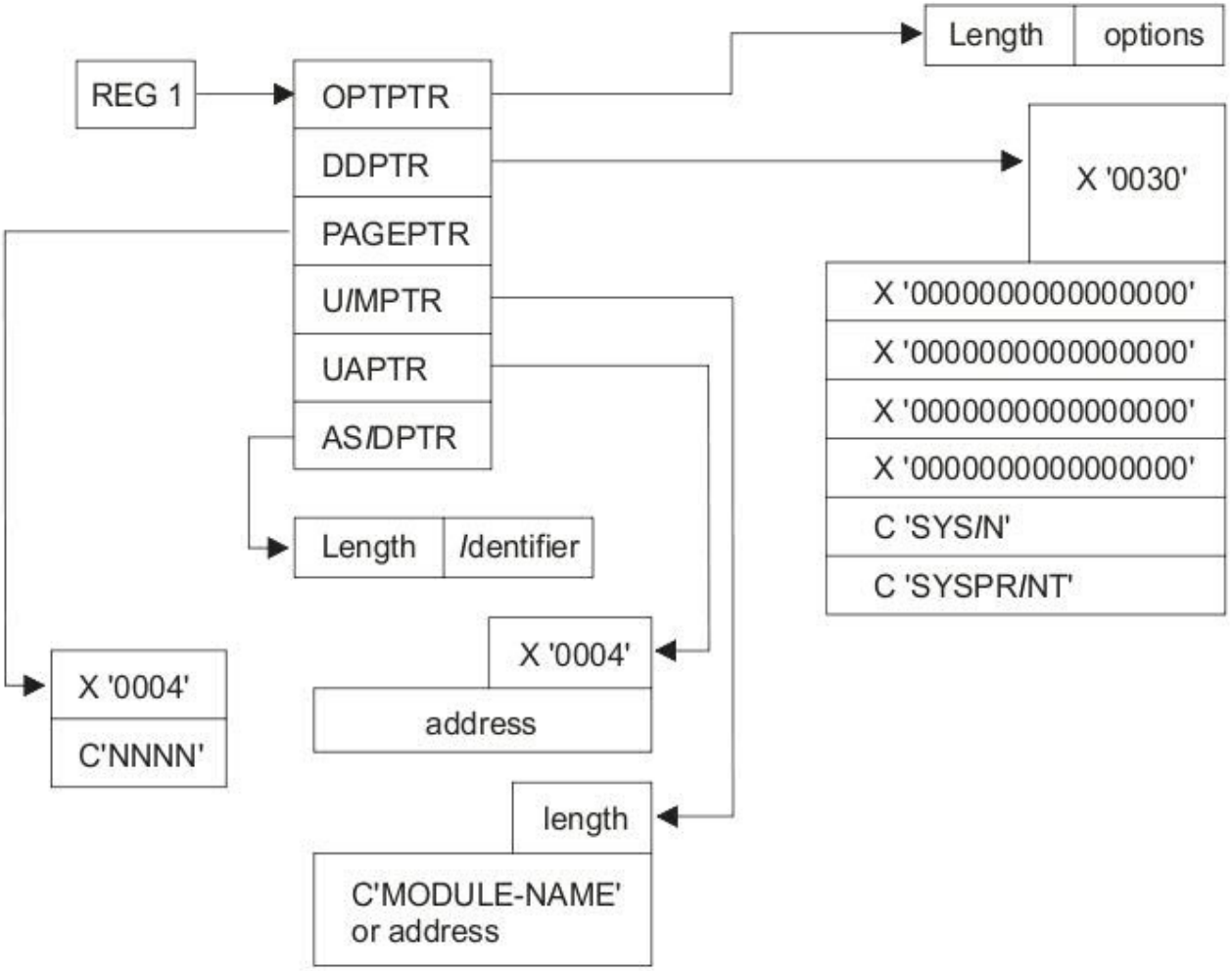


Figure 24. DFSMSdss Application Interface Structure

Application Programming Interface

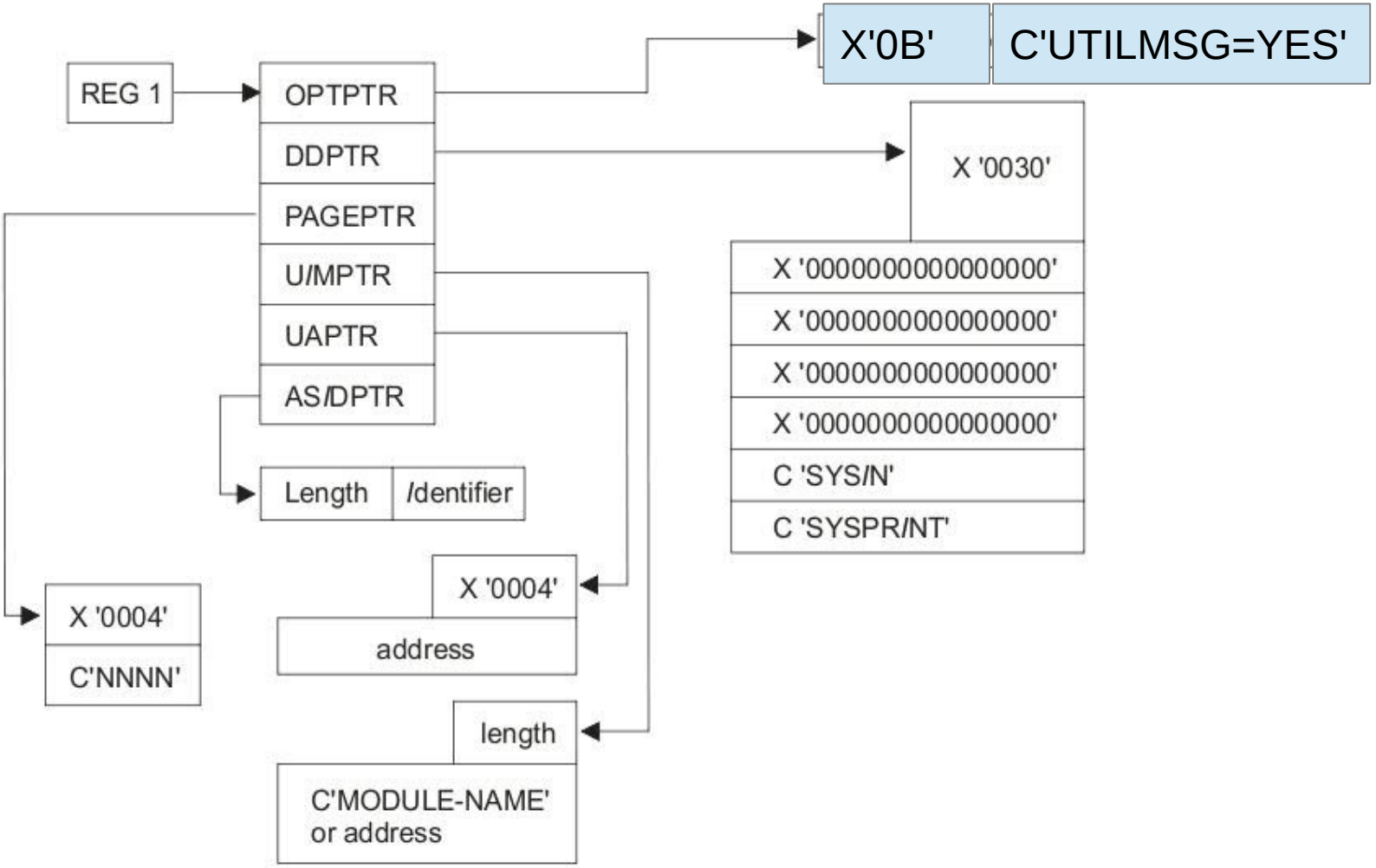


Figure 24. DFSMSdss Application Interface Structure

Application Programming Interface

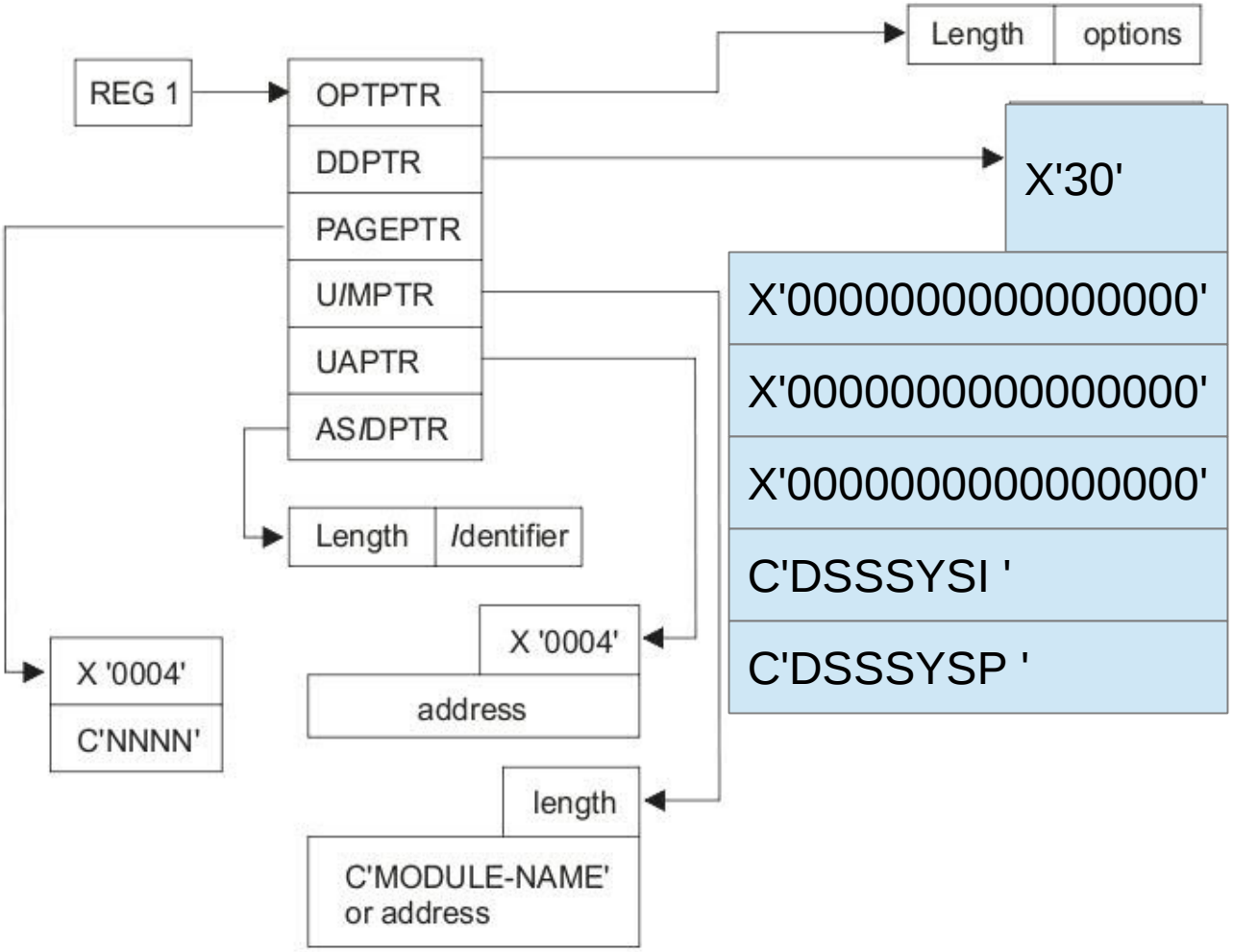


Figure 24. DFSMSdss Application Interface Structure

Application Programming Interface

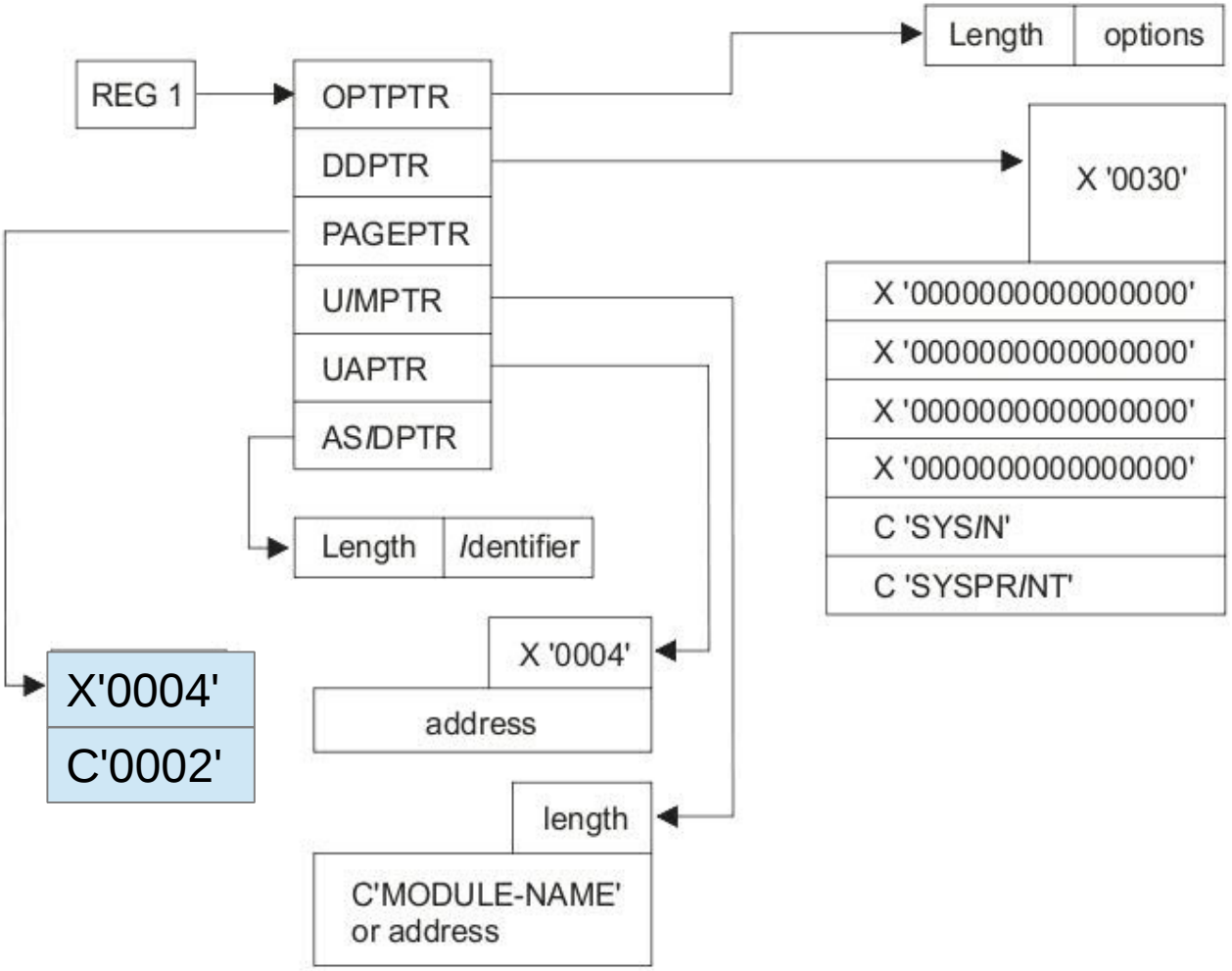


Figure 24. DFSMSdss Application Interface Structure

Application Programming Interface

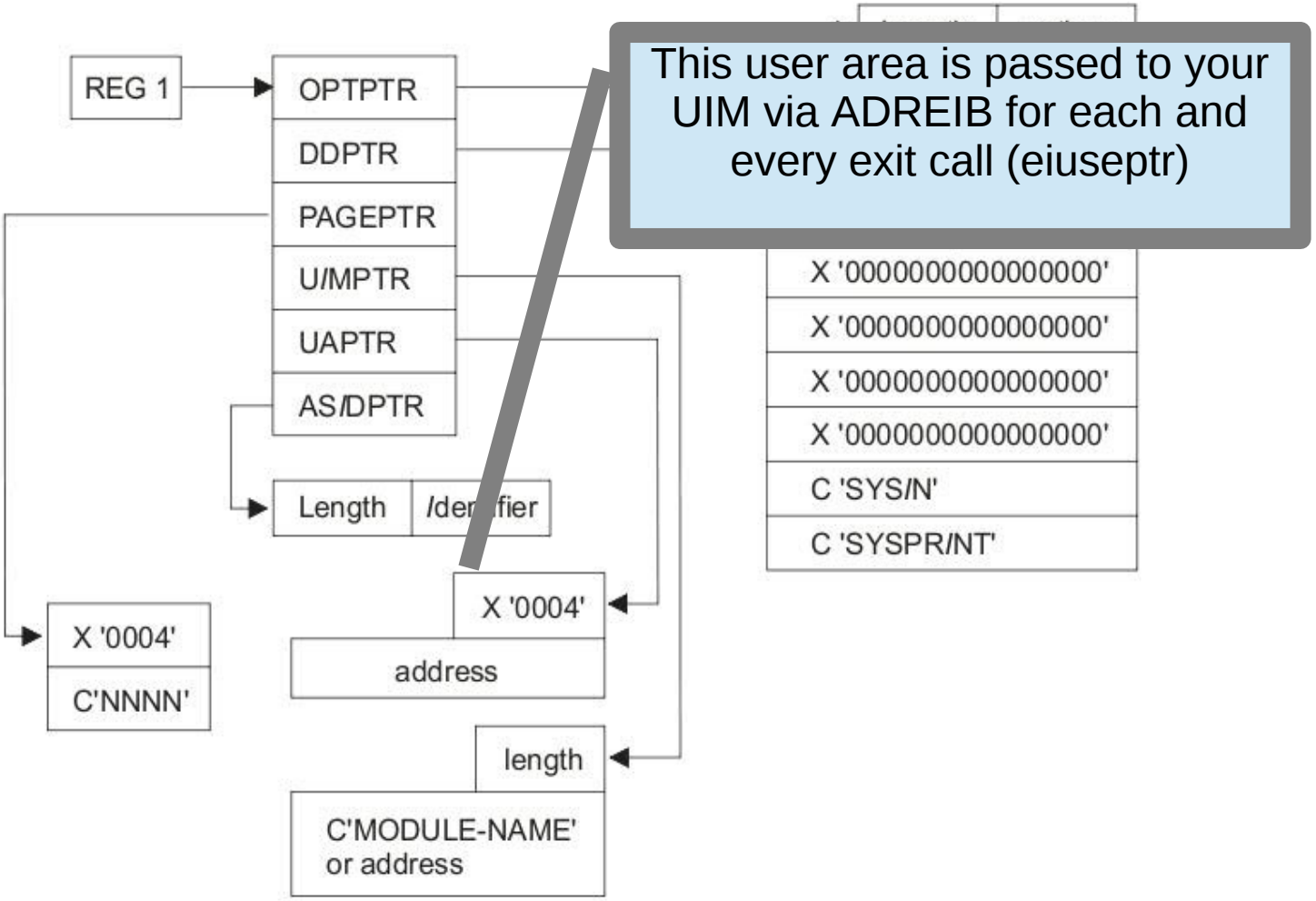


Figure 24. DFSMSdss Application Interface Structure

Application Programming Interface

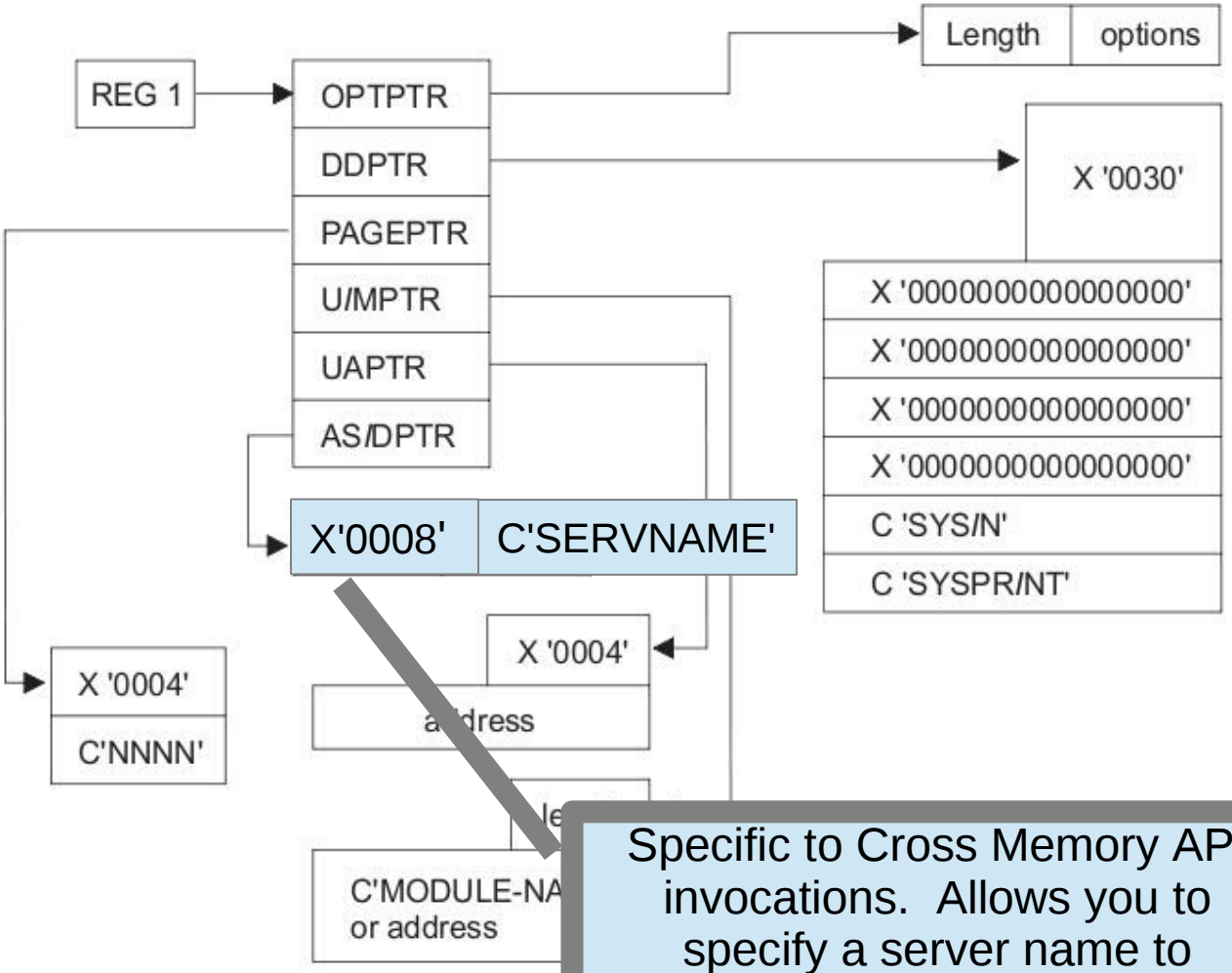


Figure 24. DFSMSdss Application Interf

Specific to Cross Memory API invocations. Allows you to specify a server name to connect to.

Application Programming Interface

- Examples calling ADRDSSU using ATTACH, LINK, CALL

```
ATTACH EP=ADRDSSU,PARAM=(OPTPTR,DDPTR,PAGEPTR,UIMPTR,UAPTR),VL=1
LINK EP=ADRDSSU,PARAM=(OPTPTR,DDPTR,PAGEPTR,UIMPTR,UAPTR),VL=1
CALL (15),(OPTPTR,DDPTR,PAGEPTR,UIMPTR,UAPTR),VL
```

- Examples calling ADRXMAIA using ATTACH, LINK, CALL

```
ATTACH EP=ADRXMAIA,PARAM=(OPTPTR,DDPTR,PAGEPTR,UIMPTR,UAPTR),VL=1
LINK EP=ADRXMAIA,PARAM=(OPTPTR,DDPTR,PAGEPTR,UIMPTR,UAPTR),VL=1
CALL (15),(OPTPTR,DDPTR,PAGEPTR,UIMPTR,UAPTR),VL
```

Application Programming Interface

```
000061B0 - 000061C4 00006260 00006294 0000629A      |../D...-...m....|
000061C0 - 800062A2                                 |...s|
```

```
000061C4 - 000BE4E3 C9D3D4E2 C77EE8C5 E2            |..UTILMSG=YES|
```

```
00006260 - 00300000 00000000 00000000 00000000      |................|
00006270 - 00000000 00000000 00000000 00000000      |................|
00006280 - 0000E2E8 E2C9D540 40400000 00000000      |..SYSIN   ......|
00006290 - 0000                                     |..|
```

```
00006294 - 0000                                     |..|
```

```
0000629A -     0004 80007360                        |  .....-|
```

```
000062A2 -     0004 0A900000                        |  ......|
```

Cross Memory API

- Value
  - Virtual Storage Constraint Relief
    - *ADRDSSU is ~2MB+ below the line*
  - Supports callers in supervisor state as well
- Trade-off
  - Extra overhead for communication between client and server
    - *Usually negligible*
  - Tape processing incurs extra buffer copy between client and server

Cross Memory API

- If a server does not exist when the DFSMSdss cross memory client is invoked
  - Client will start a server using a default server name
    - *batch JCL – DSSBATCH*
      - *ASPACE parm to specify a server name other than the default*
    - *applications – DFSMSDSS*
      - *ASIDPTR to specify a server name other than the default*

Cross Memory API

- Since the server may be started programatically, and not by an operator
  - Server shuts down after a given period of inactivity
    - *Between one and eight minutes*
    - *See SRVRTIME for details*

- If the Cross Memory Application Interface is invoked from JCL, but the ASPACE parameter is not specified, the server will shut down after 4 minutes.
- If the Cross Memory Application Interface is invoked with JCL and the ASPACE parameter is specified, the server will wait 1 minute.
- If the Cross Memory Application Interface is invoked using the LINK, CALL, or ATTACH macros, but an ASPACE name wasn't provided in the ASIDPTR field, the server will wait 8 minutes.
- If the Cross Memory Application Interface is invoked using the LINK, CALL, or ATTACH macros and an ASPACE name was provided in the ASIDPTR field, the server will wait 1 minute.

Cross Memory API

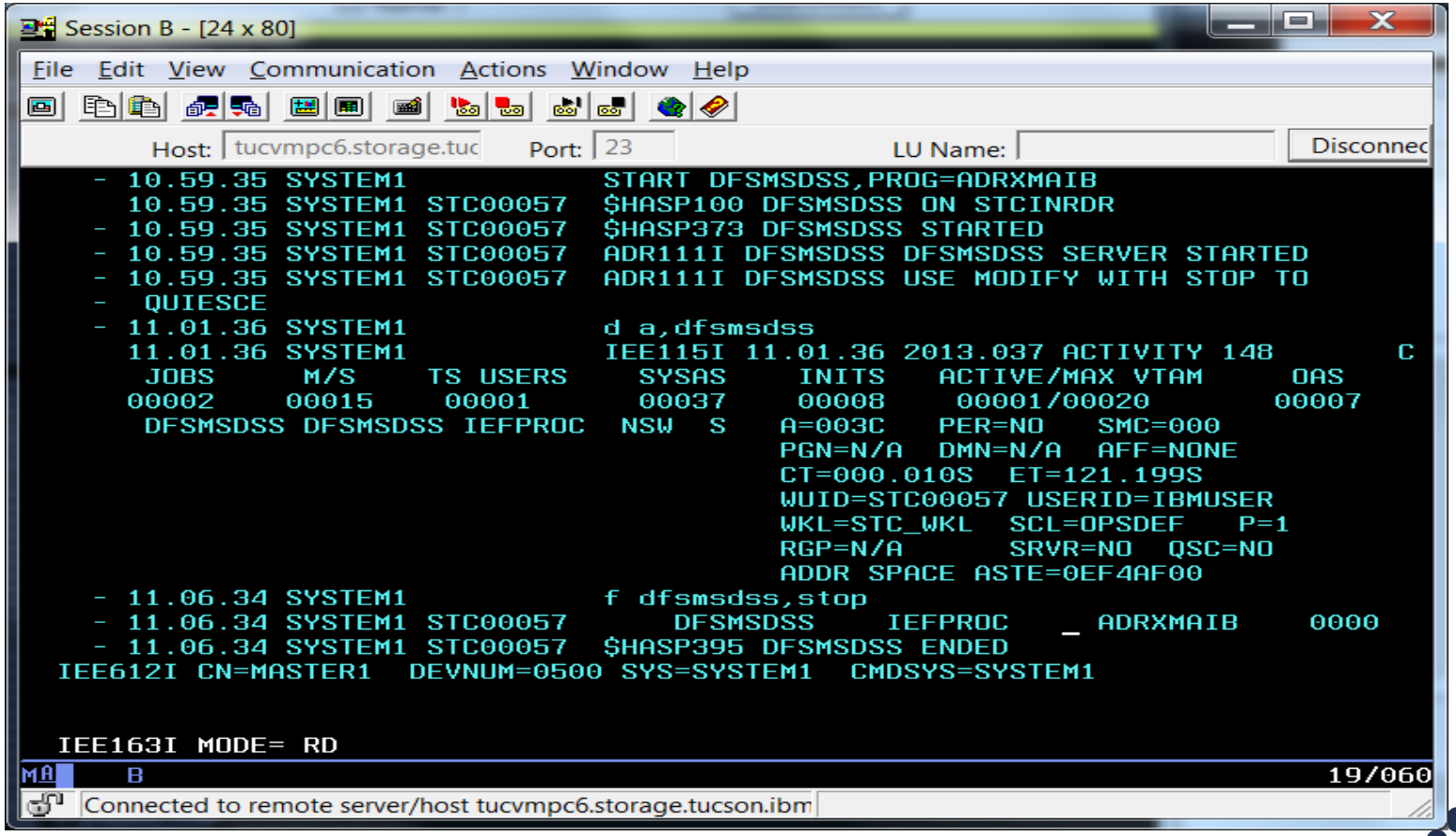
- As a system programmer one could also create specific procedures to start a DFSMSdss server with a particular server name
  - START DFSMSDSS,PROG=ADRXMAIB

Shutdown triggered by explicit stop command

```
//*******************************************************************
//*   THIS PROCEDURE WILL CREATE AN APPROPRIATE DFSMSDSS CROSS       *
//*   MEMORY SERVER TO BE USED WITH APPLICATIONS THAT INVOKE CROSS   *
//*   MEMORY REQUESTING THE DEFAULT DFSMSDSS SERVER NAME.            *
//*                                                                  *
//*   TO USE, ENTER THE FOLLOWING AT A CONSOLE:                      *
//* START DFSMSDSS,PROG=ADRXMAIB                                     *
//*                                                                  *
//*   WHEN THE DFSMSDSS CROSS MEMORY SERVER IS NO LONGER REQUIRED    *
//*   ISSUE THE FOLLOWING MODIFY COMMAND:                            *
//* F DFSMSDSS,STOP                                                  *
//*                                                                  *
//*******************************************************************
//DFSMSDSS PROC PROG=IEFBR14
//IEFPROC  EXEC PGM=&PROG,REGION=0M,TIME=1440,DYNAMNBR=1635
```

Cross Memory API

- Displaying and Stopping a DFSMSdss server



Reference

- DFSMSdss Storage Administration
- DFSMS Installation Exits
- z/OS MVS Authorized Assembler Services Reference (ALE-DYN)
- z/OS MVS System Commands

System Z Social Media Channels

Top Facebook pages related to System z:
IBM System z
IBM Academic Initiative System z
IBM Master the Mainframe Contest
IBM Destination z
Millennial Mainframer
IBM Smarter Computing

Top LinkedIn groups related to System z:
System z Advocates
SAP on System z
IBM Mainframe- Unofficial Group
IBM System z Events
Mainframe Experts Network
System z Linux
Enterprise Systems
Mainframe Security Gurus

Twitter profiles related to System z:
IBM System z
IBM System z Events
IBM DB2 on System z
Millennial Mainframer
Destination z
IBM Smarter Computing

YouTube accounts related to System z:
IBM System z
Destination z
IBM Smarter Computing

- **Top System z blogs to check out:**
  - Mainframe Insights
  - Smarter Computing
  - Millennial Mainframer
  - Mainframe & Hybrid Computing
  - The Mainframe Blog
  - Mainframe Watch Belgium
  - Mainframe Update
  - Enterprise Systems Media Blog
  - Dancing Dinosaur
  - DB2 for z/OS
  - IBM Destination z
  - DB2utor

The In's and Out's of DFSMSdss Exit Processing and Patch Options

Robert Gensler
IBM
rgensle@us.ibm.com

Session 16132
August 5, 2014

www.SHARE.org

#SHAREorg