231

CICS update

February 2005

In this issue

A xephon PUBLICATION

Published by

Xephon Inc
PO Box 550547
Dallas, Texas 75355
USA

Phone: 214-340-5690
Fax: 214-341-7081

Editor

Trevor Eddolls
E-mail: trevore@xephon.com

Publisher

Colin Smith
E-mail: info@xephon.com

Subscriptions and back-issues

A year's subscription to *CICS Update*, comprising twelve monthly issues, costs $270.00 in the USA and Canada; £175.00 in the UK; £181.00 in Europe; £187.00 in Australasia and Japan; and £185.50 elsewhere. In all cases the price includes postage. Individual issues, starting with the December 2000 issue, are available separately to subscribers for $24.00 (£16.00) each including postage.

CICS Update on-line

Code from *CICS Update*, and complete issues in Acrobat PDF format, can be downloaded from our Web site at http://www.xephon.com/cics; you will need to supply a word from the printed issue.

Contributions

When Xephon is given copyright, articles published in *CICS Update* are paid for at the rate of $160 (£100 outside North America) per 1000 words and $80 (£50) per 100 lines of code for the first 200 lines of original material. The remaining code is paid for at the rate of $32 (£20) per 100 lines. To find out more about contributing an article, without any obligation, please download a copy of our *Notes for Contributors* from www.xephon.com/nfc.

Printed in England.

Programs to incorporate in your CICS PROC

Here are two REXX programs you can incorporate in your CICS PROC to manage start-ups more effectively. The first one will establish a known return code based on the start-up type; the second one will recreate VSAM files from a model. The first program is called PRECICS. It will accept the &START value from your CICS PROC and establish a return code to make decisions to execute optional steps and also print a nice easy-to-find WTO stating the CICS start type. The second is called PREVSAM. It can be used to do IDCAMS deletes, defines, and inits for common VSAM files (like the GCD and LCD). Here is an example of a CSPM CMAS PROC that uses both PRECICS and PREVSAM:

```
//CMASPROD  PROC CICLVL=CTS22Ø
//*********************************************************
//***           I N I T I A L  S T A R T  ?      ********
//*********************************************************
//PRECICS   EXEC PGM=IKJEFTØ1,PARM='PRECICS &START'
//SYSEXEC   DD   DSN=your.exec.pds,DISP=SHR
//SYSTSPRT  DD   SYSOUT=*
//SYSTSIN   DD   DUMMY
//          IF   (PRECICS.RC = 4Ø) THEN
//PREVSAM   EXEC PGM=IKJEFTØ1,DYNAMNBR=99,
//          PARM='PREVSAM PROD &RGA 5'
//SYSEXEC   DD   DSN=your.exec.pds,DISP=SHR
//VSAMØØ1   DD   DSN=CICS.PROD.IDCAMS.CTLLIB(GCDMODXX),DISP=SHR
//VSAMØØ2   DD   DSN=CICS.PROD.IDCAMS.CTLLIB(LCDMODXX),DISP=SHR
//SYSTSPRT  DD   SYSOUT=*
//SYSPRINT  DD   SYSOUT=*
//SYSTSIN   DD   DUMMY
//* INITIALIZE DFHGCD FILE
//INITGCD   EXEC PGM=DFHRMUTL,REGION=1M
//STEPLIB   DD DSN=CICS.PROD.&CICLVL..SDFHLOAD,DISP=SHR
//SYSPRINT  DD SYSOUT=*
//SYSUDUMP  DD SYSOUT=*
//DFHGCD    DD  DISP=SHR,DSN=CICS.PROD.&RGA..DFHGCD
//SYSIN     DD  DISP=SHR,DSN=CICS.PROD.IDCAMS.CTLLIB(INITGCD)
//* INITIALIZE DFHLCD FILE
//INITLCD   EXEC PGM=DFHCCUTL
//STEPLIB   DD DSN=CICS.PROD.&CICLVL..SDFHLOAD,DISP=SHR
//SYSPRINT  DD  SYSOUT=*
//SYSUDUMP  DD  SYSOUT=*
```

```
//DFHLCD    DD  DISP=SHR,DSN=CICS.PROD.&RGA..DFHLCD
//*******************************************************************
//* CREATE NEW GDG BACK-UP OF EYUDREP
//*******************************************************************
//DREPBACK  EXEC PGM=IDCAMS
//SYSPRINT  DD   SYSOUT=*
//INDD      DD   DISP=SHR,DSN=CICS.PROD.&RGA..EYUDREP
//OUDD      DD   DSN=CICS.PROD.&RGA..EYUDREP.BKUP(+1),
//           DISP=(,CATLG),UNIT=SYSDA,SPACE=(CYL,(100,100),RLSE),
//           RECFM=VB,LRECL=8192,BLKSIZE=0
//SYSIN     DD   DISP=SHR,DSN=CICS.PROD.IDCAMS.CTLLIB(REPDREP)
//*******************************************************************
//* DELETE/DEFINE AND RESTORE ONLY OCCUR IF BACK-UP WAS GOOD
//*******************************************************************
//           IF   (DREPBACK.RC = 0) THEN
//*******************************************************************
//* DELETE AND DEFINE THE NEW DREP IF BACK-UP WAS GOOD
//*******************************************************************
//DREPDEL   EXEC PGM=IKJEFT01,DYNAMNBR=99,PARM='PREVSAM PROD &RGA'
//SYSEXEC   DD   DISP=SHR,DSN=CICS.PROD.EXEC
//VSAM001   DD   DISP=SHR,DSN=CICS.PROD.IDCAMS.CTLLIB(DREPDEF)
//SYSTSPRT  DD   SYSOUT=*
//SYSPRINT  DD   SYSOUT=*
//SYSTSIN   DD   DUMMY
//*******************************************************************
//* RESTORE THE BACK-UP OF THE DREP TO THE NEWLY-DEFINED CLUSTER
//*******************************************************************
//DREPREST  EXEC PGM=IDCAMS
//SYSOUT    DD   SYSOUT=*
//SYSPRINT  DD   SYSOUT=*
//INDD      DD   DISP=SHR,DSN=CICS.PROD.&RGA..EYUDREP.BKUP(+1)
//OUDD      DD   DISP=SHR,DSN=CICS.PROD.&RGA..EYUDREP
//SYSIN     DD   DISP=SHR,DSN=CICS.PROD.IDCAMS.CTLLIB(REPDREP)
//*******************************************************************
//* END OF DREP REORG
//*******************************************************************
//           ENDIF
//*******************************************************************
//* END OF INITIAL PROCESSING
//*******************************************************************
//           ENDIF
//*******************************************************************
//* CMAS JCL
//*******************************************************************
//CMASSTRT  EXEC PGM=EYU9XECS,
//           TIME=1440,
//           REGION=0K,
//           PARM=('SI,START=&START')
//STEPLIB   DD  DISP=SHR,DSN=CICS.PROD.&CICLVL..SEYUAUTH
//           DD  DISP=SHR,DSN=CICS.PROD.&CICLVL..SDFHAUTH
```

```
//          DD  DISP=SHR,DSN=CICS.PROD.&CICLVL..TABLIB
//*-------------------------------------------------------------
//* CICS Datasets
//*-------------------------------------------------------------
//DFHRPL    DD  DISP=SHR,DSN=CICS.PROD.&CICLVL..SEYULOAD
//          DD  DISP=SHR,DSN=CICS.PROD.&CICLVL..SDFHLOAD
//          DD  DISP=SHR,DSN=CICS.PROD.&CICLVL..TABLIB
//* CICS PARMETERS
//SYSIN     DD  DISP=SHR,DSN=CICS.PROD.&CICLVL..CTLLIB(&SIN)
//*-------------------------------------------------------------
//* CICS DATASETS
//*-------------------------------------------------------------
//DFHINTRA  DD  DISP=SHR,DSN=CICS.PROD.&RGA..DFHINTRA
//DFHLCD    DD  DISP=SHR,DSN=CICS.PROD.&RGA..DFHLCD
//DFHGCD    DD  DISP=SHR,DSN=CICS.PROD.&RGA..DFHGCD
//DFHTEMP   DD  DISP=SHR,DSN=CICS.PROD.&RGA..DFHTEMP
//DFHDMPA   DD  DISP=SHR,DSN=CICS.PROD.&RGA..DFHDMPA
//DFHDMPB   DD  DISP=SHR,DSN=CICS.PROD.&RGA..DFHDMPB
//DFHAUXT   DD  DISP=SHR,DSN=CICS.PROD.&RGA..DFHAUXT
//DFHBUXT   DD  DISP=SHR,DSN=CICS.PROD.&RGA..DFHBUXT
//DFHLRQ    DD  DISP=SHR,DSN=CICS.PROD.&RGA..DFHLRQ
//DFHCXRF   DD  SYSOUT=*
//* EXTRAPARTITION DATASETS
//LOGUSR    DD  SYSOUT=*
//MSGUSR    DD  SYSOUT=*
//EYULOG    DD  SYSOUT=*
//PLIMSG    DD  SYSOUT=*
//COUT      DD  SYSOUT=*
//MSGCSML   DD  SYSOUT=*,DCB=(DSORG=PS,RECFM=V,BLKSIZE=136)
//*-------------------------------------------------------------
//* CPSM Datasets
//*-------------------------------------------------------------
//EYUDREP   DD  DISP=SHR,DSN=CICS.PROD.&RGA..EYUDREP
//EYUPARM   DD  DISP=SHR,DSN=CICS.PROD.&CICLVL..SEYUPARM(&RGA)
//BBIPARM   DD  DISP=SHR,DSN=CICS.PROD.&CICLVL..EYUIPRM
//BBACTDEF  DD  DISP=SHR,DSN=CICS.PROD.&CICLVL..SEYUADEF
//BBVDEF    DD  DISP=SHR,DSN=CICS.PROD.&CICLVL..SEYUVDEF
//*****************************************************************
//* USE IEBGENER TO COPY DFHAUXT DATASET TO GDG, EVEN DURING ABEND
//*****************************************************************
//COPYAUX   EXEC PGM=IEBGENER,COND=EVEN
//SYSUT1    DD   DSN=CICS.PROD.&RGA..DFHAUXT,DISP=SHR
//SYSUT2    DD   DSN=CICS.PROD.&RGA..AUXTRAA(+1),DISP=(,CATLG),
//          UNIT=SYSDA,SPACE=(CYL,(100,100),RLSE),REFDD=*.SYSUT1
//SYSPRINT  DD   SYSOUT=*
//SYSIN     DD   DUMMY
//*****************************************************************
//* USE IEBGENER TO COPY DFHBUXT DATASET TO GDG, EVEN DURING ABEND
//*****************************************************************
//COPYBUX   EXEC PGM=IEBGENER,COND=EVEN
```

```
//SYSUT1     DD   DSN=CICS.PROD.&RGA..DFHBUXT,DISP=SHR
//SYSUT2     DD   DSN=CICS.PROD.&RGA..AUXTRAB(+1),DISP=(,CATLG),
//           UNIT=SYSDA,SPACE=(CYL,(100,100),RLSE),REFDD=*.SYSUT1
//SYSPRINT   DD   SYSOUT=*
//SYSIN      DD   DUMMY
```

PRECICS

```
/*********************************************************************/
/*                               REXX                                */
/*********************************************************************/
/* Purpose: Simple EXEC to set a batch RC based on an input string   */
/*-------------------------------------------------------------------*/
/* Syntax:  PRECICS parm                                             */
/*-------------------------------------------------------------------*/
/* Parms: parm       - input string                                  */
/*                                                                   */
/* Notes: Sample JCL to run in batch:                                */
/* //SETRC      EXEC PGM=IKJEFT01,PARM='PRECICS &START'             */
/* //SYSEXEC    DD   DSN=your.exec.pds,DISP=SHR                     */
/* //SYSTSPRT   DD   SYSOUT=*                                       */
/* //SYSTSIN    DD   DUMMY                                          */
/*********************************************************************/
/*                      Change Log                                   */
/*********************************************************************/
/* Accept and uppercase the input                                    */
/*********************************************************************/
arg parm
/*********************************************************************/
/* Parse out anything after the start type (looks for a comma)       */
/*********************************************************************/
if pos(',',parm) <> 0 then
    parse var parm start_type ',' .
else
    start_type = parm
/*********************************************************************/
/* Save the start time and get the current jobname                   */
/*********************************************************************/
start_time = date() time()
cics = mvsvar('SYMDEF','JOBNAME')
/*********************************************************************/
/* Build a message                                                   */
/*********************************************************************/
msg = cics 'Start Type: "'start_type'" Parm received: "'parm'"'
/*********************************************************************/
/* Use select to provide multiple mutually exclusive alternatives    */
/* Simply duplicate the 'when' line with any new input values        */
/*********************************************************************/
select
```

```
   when start_type = 'AUTO' then EXITRC = 1Ø
   when start_type = 'EMERGENCY' then EXITRC = 2Ø
   when start_type = 'COLD' then EXITRC = 3Ø
   when start_type = 'INITIAL' then EXITRC = 4Ø
   otherwise EXITRC = Ø
end
/*********************************************************************/
/* Use TSO SEND to make an easy to find eyecatcher in the JOBLOG     */
/*********************************************************************/
say start_time msg', RC='EXITRC
"SEND '"msg"'"
/*********************************************************************/
/* Shutdown                                                          */
/*********************************************************************/
shutdown: exit(EXITRC)
```

PREVSAM

```
/*********************************************************************/
/*                                REXX                               */
/*********************************************************************/
/* Purpose: Simple EXEC to modify and execute IDCAMS statements      */
/*-------------------------------------------------------------------*/
/* Syntax:  PREVSAM parms                                            */
/*-------------------------------------------------------------------*/
/* Parms: parms      - Any number of parms                           */
/*                                                                   */
/* If <PARM1> is found in the VSAMxxxx input, it will be replaced    */
/* with the first positional parm                                    */
/* If <PARM2> is found in the VSAMxxxx input, it will be replaced    */
/* with the second positional parm                                   */
/* If <PARM3> is found in the VSAMxxxx input, it will be replaced    */
/* with the third positional parm                                    */
/* and so on                                                         */
/*                                                                   */
/* Notes: Sample JCL to run in batch                                 */
/* //IDCAMS     EXEC PGM=IKJEFTØ1,PARM='PREVSAM P1A7 CTS13Ø 2Ø'      */
/* //SYSEXEC    DD   DSN=your.exec.pds,DISP=SHR                      */
/* //SYSTSPRT   DD   SYSOUT=*         * PREVSAM output *            */
/* //SYSPRINT   DD   SYSOUT=*         * IDCAMS output  *            */
/* //VSAMxxxx   DD   DSN=your.ctllib.pds(model1),DISP=SHR           */
/* //VSAMyyyy   DD   DSN=your.ctllib.pds(model2),DISP=SHR           */
/* //SYSTSIN    DD   DUMMY                                          */
/*                                                                   */
/* Any number of unique VSAMxxxx DDs are supported                   */
/* PREVSAM will stop if any non-zero return code is encountered      */
/*                                                                   */
/* model member sample:                                              */
/* /* COMMENTS */                                                   */
```

```
/* DELETE 'your.<PARM2>.<PARM1>.cluster'                                */
/* IF LASTCC > Ø THEN SET MAXCC = Ø                                     */
/* /* COMMENTS */                                                      */
/* DEFINE CLUSTER(NAME('your.<PARM2>.<PARM1>.cluster') -                */
/*        INDEXED -                                                     */
/*        CYLINDERS(<PARM3> 5)  -                                       */
/*        SHR(2) -                                                      */
/*        FREESPACE(1Ø 1Ø) -                                            */
/*        REUSE -                                                       */
/*        CISZ(8192) -                                                  */
/*        KEYS(28 Ø) -                                                  */
/*        )                                                             */
/*********************************************************************/
/*                       Change Log                                  */
/*********************************************************************/
/* Accept and uppercase the input                                    */
/*********************************************************************/
arg parms
/*********************************************************************/
/* Determine the number of parms and set into unique variables       */
/*********************************************************************/
pcount = words(parms)
say pcount 'Parms received:' parms
say
do i=1 to pcount
    parse var parms parm.i parms
    string.i = '<PARM'i'>'
    say string.i 'will be replaced with' parm.i
end
/*********************************************************************/
/* Trap the output from the LISTA STATUS command                     */
/*********************************************************************/
x = outtrap('lines.')
"LISTALC STATUS"
EXITRC = RC
/*********************************************************************/
/* Parse out the DDNAMEs and concatenate into a list                 */
/*********************************************************************/
ddlist = ''
do i=1 to lines.Ø
    if words(lines.i) = 2 then
       do
        parse upper var lines.i ddname .
        ddlist = ddlist ddname
       end
    else
       do
        iterate
       end
end
```

```
/*********************************************************************/
/* Look for "VSAM" DDs in the DDLIST                                 */
/*********************************************************************/
do v=1 to words(ddlist)
    if substr(word(ddlist,v),1,4) = 'VSAM' then
       do
        vsamdd = word(ddlist,v)
        say
        say 'VSAM DD' vsamdd 'found'
        say
/*********************************************************************/
/* Read the contents of the VSAMDD                                   */
/*********************************************************************/
       "EXECIO * DISKR" vsamdd "(STEM VSAMDD. FINIS"
        EXITRC = RC
        if EXITRC <> Ø then signal shutdown
/*********************************************************************/
/* Loop through all the lines in the VSAMDD                          */
/*********************************************************************/
        do m=1 to vsamdd.Ø
           line = substr(vsamdd.m,1,72)
/*********************************************************************/
/* Determine whether any substitution needs to occur                 */
/*********************************************************************/
          do s=1 to pcount
             loc = pos(string.s,line)
             len = length(string.s)
/*********************************************************************/
/* If any <PARMx> found, replace it with input parmx                 */
/*********************************************************************/
             if loc <> Ø then
                do
                 line = insert(parm.s,delstr(line,loc,len),loc-1)
                end
          end
          sysin.m = line
        end
/*********************************************************************/
/* Allocate SYSIN                                                    */
/*********************************************************************/
       "ALLOC F(SYSIN) UNIT(VIO) SPACE(1) LRECL(8Ø) BLKSIZE(Ø)"
        EXITRC = RC
        if EXITRC <> Ø then signal shutdown
/*********************************************************************/
/* Write the lines to SYSIN                                          */
/*********************************************************************/
       "EXECIO * DISKW SYSIN (STEM SYSIN. FINIS"
        EXITRC = RC
        if EXITRC <> Ø then signal shutdown
/*********************************************************************/
```

```
/* Invoke IDCAMS                                                      */
/*********************************************************************/
         address ATTCHMVS 'IDCAMS'
         EXITRC = RC
         if EXITRC <> 0 then signal shutdown
/*********************************************************************/
/* FREE SYSIN                                                        */
/*********************************************************************/
        "FREE F(SYSIN)"
         EXITRC = RC
         if EXITRC <> 0 then signal shutdown
/*********************************************************************/
/* Drop the contents of the VSAMDD stem                              */
/*********************************************************************/
         drop vsamdd.
         drop sysin.
        end
end
/*********************************************************************/
/* Shutdown                                                          */
/*********************************************************************/
shutdown: exit(EXITRC)
```

Robert Zenuk
Systems Programmer (USA)

CICS and HIS 2004

With one bound, Microsoft saves CICS users!

Microsoft has realized that mainframes might be here to stay and that the best way to get part of that huge market is to offer some kind of integration product. Since 1 September 2004, Microsoft has made available Host Integration Server (HIS) 2004 for companies with mainframes that want closer integration with Windows-based servers, software, and SOA (Service-Oriented Architectures – this year's must-have acronym).

Basically, the product links IBM mainframes with Windows-based servers and software. It means that mainframe applications can be associated with Web services and made

available over the Internet at the touch of a URL (so to speak). The Microsoft HIS 2004 Enterprise Edition includes Transaction Integrator tools to liberate (in the way that liberation armies do with things that belonged to the previous regime) the mainframe data via XML or .NET.

“Hang on,” I hear you say, “why do I want rescuing?” Microsoft claims that in addition to the Web services bit, it’s cheaper and more efficient to develop CICS applications on a PC than on a mainframe!

HIS started life as an SNA (System Network Architecture) server in 1992. For most of its life it has been a gateway product, acting as a pass-through for SNA. Microsoft has added links to mainframe-based transaction programs with COM TI (Component Object Model Transaction Integrator) and developed links to DB2.

What Microsoft hadn’t done was offer any kind of integration between its developments tools and the server. So being able to initiate processes on Windows from the mainframe or on the mainframe from Windows is a big leap forward.

HIS 2004 focuses on integrating mainframe transaction-based applications into the Windows infrastructure, such as links to its BizTalk orchestration engine and Visual Studio development tool environment.

Host-Initiated Processing (HIP) lets the host system make calls into the Windows environment, allowing, for example, the mainframe to populate a form on the Windows platform and kick off a print process. HIP makes the mainframe environment think it is making calls locally. There is now two-directional transactional activity. Previously the Windows environment would pull data; now the host CICS application, for example, can push data to the Windows environment.

HIS 2004 includes tighter integration with Visual Studio. The Transaction Integrator design tool, which runs within the project designer in Visual Studio, lets developers expose

CICS and IMS transactions to Windows as COM+ components, .Net packages, or XML-based Web services. Microsoft has added a feature for tunnelling SNA traffic into the mainframe over IP. The new IP-DLC Link Service lets PCs connect to z900 mainframes via IP networks.

Also new is a managed provider for DB2, which allows DB2 data to be published as Web services or integrated into Windows forms, such as those produced by Microsoft's InfoPath application.

With HIP it's now possible to have a large CICS-based system and take a subset of it and move it to Windows without having to change anything on the mainframe.

To be fair, Microsoft has gone to great lengths to ensure interoperability between mainframe technology and Windows Server and MS SQL. For example:

- TI (Transaction Integrator) enables Windows developers to publish and extend business rules in CICS, IMS (and AS/400 applications) as XML Web services.
- Managed provider for DB2 enables data stored in DB2 to be published as XML Web services and integrate DB2 data with applications based on Windows Forms, Web Forms, Web services, or Microsoft Office System productivity applications such as Excel and InfoPath.
- TI HIP (Transaction Integrator Host-Initiated Processing) allows a Windows Server to function as a peer to an IBM mainframe (and AS/400 computer). This allows users to build distributed peer-to-peer applications and to move parts of their host application logic and data to a Windows Server and SQL Server. This they describe as 'a more cost-effective infrastructure'!
- ESSO (Enterprise Single Sign-On) provides a way to authenticate security credentials between Windows Active Directory and non-Windows systems. This means that users need to sign on only once in order to access HIS

2004, BizTalk Server 2004, mainframes, mid-range systems, and mainframe applications such as CICS, IMS, DB2, and MQSeries.

- Internet Protocol-Data Link Control (IP-DLC) Link Service supports SNA over IP routing so that HIS 2004 computers can connect directly to z900 mainframes via high-speed IP networks. This removes the need to remotely administer branch cluster controllers, utilize DLSw-capable (Data Link Switching) routers, or maintain FEPs.

You probably want to know how much this 'rescue' by the superheroes from Redmond is going to cost you. HIS 2004 Standard Edition, with core host access services, is quoted at $2,499. The fully-featured (and come to think of it you'll probably need all of them and more) Enterprise Edition, which includes Transaction Integrator, is priced at $9,999.

"But", I hear you ask, "is Microsoft the only choice for off-mainframe application development?" You'll be pleased to know that the answer is "no" – IBM has its Host Access Transformation Services. This involves WebSphere and Java 2 Platform Enterprise Edition.

There's also a host of third-party vendors (including ClientSoft, Farabi Technology, NetMange, Proginet, and Seagull Holding) that offer tools to include your mainframe as part of a distributed architecture. Companies including Jacada, Neon Systems, and WRQ are selling programmatic integration products.

The basic assumption behind HIS 2004 and the other products is that running everything on the mainframe is costly and complex. So, the solution is to run applications in a distributed environment. That way, the mainframe needs to run only the core system, and logical subsystems can be moved to what they describe as 'lower-cost platforms'. Of course, many would argue that there's more to the cost of a platform than the initial purchase price! For the time being, CICS developers should be aware of these alternatives, but they are unlikely to see any compelling reason to use them.

Nick Nourse
Independent Consultant (UK)

External CICS Interface (EXCI) client interface to DFHEDAP

INTRODUCTION

The EXCI batch client control program (CM420) was covered in the November and December 2004 issues of *CICS Update* (issues 228 and 229). The EXCI client interface to DFHEDAP is an example of how easy it is to write an EXCI client program that is controlled by CM420.

Although the DFHCSDUP utility is most definitely the recommended method for bulk batch CSD updates, the programmable interface to resource definition on-line (DFHEDAP) does have some distinct advantages. The INSTALL command is supported, so you can immediately install an updated resource in the CICS region from batch. Also, all the CEDA commands are logged directly in the CICS region in which they are processed, providing an audit trail. This audit trail contains the userid and terminal (or session) used to process the CEDA command.

THE PROGRAMS

All programs are written in Assembler. However, an EXCI client program could be written in any of the languages supported (C, COBOL, or PL/I).

CM412 is the EXCI client interface to DFHEDAP.

CM411 is the CICS server interface to DFHEDAP.

CM411A01 is the Assembler copybook COMMAREA CM411COM.

CM412

CM412 is linked to by the EXCI batch client control program (CM420), which has established the EXCI environment so that DPL requests can now be processed. CM420 also loads an I/O subprogram, CM419.

Here is an example of the JCL required to link to CM412 from CM420:

```
//* ================================================================ *
//*  EXECUTE AN EXTERNAL CICS INTERFACE (EXCI) CLIENT PROGRAM         *
//* ================================================================ *
//STEP1    EXEC PGM=CM42Ø,
//         PARM='CM412,TESTCICS'
//STEPLIB  DD   DISP=SHR,DSN=CWM.CICSTS.TEST.LOAD
//         DD   DISP=SHR,DSN=CWM.CICSTS.TEST.SDFHEXCI
//SYSPRINT DD   SYSOUT=*
//SYSUDUMP DD   SYSOUT=*
//SYSIN    DD   *
*********************************************************************
*COMMAND  +CONTINUATION                                             *
*********************************************************************
DELETE ALL(*) GROUP(CWMTEST)
*
DEFINE TRANCLASS(CWM1) GROUP(CWMTEST)
          MAXACTIVE(1Ø)
/*
```

CM412 uses the I/O subprogram to read any number of CEDA commands from SYSIN. The CEDA commands are processed sequentially and each command results in a single DPL request. Before the DPL request is processed the CEDA command is written to SYSPRINT, which makes it easier to relate any error messages to the command processed. CM412 does not perform any validation of the CEDA command, it simply provides a 1022-byte buffer for the CEDA command. This has the advantage that any and all future CEDA commands can be supported without having to change CM412. Also,

DFHEDAP always validates the CEDA commands itself before attempting to process them.

CM412 processes a DPL request to the CICS server interface to DFHEDAP (CM411). When control is returned from CM411, CM412 checks the EXCI response, the DPL response and the CEDA diagnostic (command validation), and CEDA execution return codes. If a command is invalid, it will not be processed by DFHEDAP. Appropriate messages are written to SYSPRINT before processing the next CEDA command (if there is one) or returning control to CM420 for EXCI clean-up processing.

Here is an example of the messages written to SYSPRINT:

```
CM420000I *CM420   -CWM00001 01/30/04 09.54*

CM420002I Program CM420    loaded at address X'80007138'.
CM420003I WORKING  storage address X'00006BF8' length 00000856 bytes.
CM420003I CM420COM storage address X'00006EA8' length 00000160 bytes.

CM420002I Program CM419    loaded at address X'80008D10'.
CM420003I CM419COM storage address X'00006C80' length 00000548 bytes.

CM420101I PARM Field - Program=CM412    Region=TESTCICS.

CM420201I INITIALIZE_USER processing....
CM420202I INITIALIZE_USER successful.

CM420301I ALLOCATE_PIPE processing....
CM420302I ALLOCATE_PIPE successful.

CM420401I OPEN_PIPE processing....
CM420402I OPEN_PIPE successful.

CM420001I Loading program CM412    ....
CM420002I Program CM412    loaded at address X'800080C8'.
CM420501I LINKing to sub-program CM412    ...

********************************************************************************
CM412000I *CM412   -CWM00001 02/05/04 12.22*

CM412002I Program CM412    loaded at address X'800080C8'.
CM412003I WORKING  storage address X'0000F758' length 00002212 bytes.
CM412003I CM411COM storage address X'0000F7C0' length 00002100 bytes.

          ************************************************************
          *COMMAND  +CONTINUATION                                    *
```

```
          ************************************************************
          DELETE ALL(*) GROUP(CWMTEST)

CM412501I DPL_CALL to CM411 processing....
CM412503I CEDA Diagnostics Code . : 00000000
CM412504I CEDA Execution Code . . : 00000000
CM412502I DPL_CALL to CM411 complete. Return Code 00000000

          DEFINE TRANCLASS(CWM1) GROUP(CWMTEST)
                    MAXACTIVE(10)

CM412501I DPL_CALL to CM411 processing....
CM412503I CEDA Diagnostics Code . : 00000000
CM412504I CEDA Execution Code . . : 00000000
CM412502I DPL_CALL to CM411 complete. Return Code 00000000

          DEFINE FILE(TEST01) GROUP(CWMTEST)
                    DESCRIPTION(EXCI TEST01 VSAM FILE)
                    DSNAME(CWM.TEST.EXCI.TEST01)
                    LSRPOOLID(1) DSNSHARING(ALLREQS)
                STRINGS(1) KEYLENGTH(9) STATUS(ENABLED) OPENTIME(FIRSTREF)
                    DISPOSITION(SHARE) DATABUFFERS(2) INDEXBUFFERS(1)
                    TABLE(NO)
                   RECORDFORMAT(V) ADD(NO) BROWSE(NO) DELETE(NO) READ(YES)
                    UPDATE(NO) JOURNAL(NO) JNLREAD(NONE) JNLSYNCREAD(NO)
                JNLUPDATE(NO) JNLADD(NONE) JNLSYNCWRITE(NO) RECOVERY(NONE)
                    FWDRECOVLOG(NO) BACKUPTYPE(STATIC)

CM412501I DPL_CALL to CM411 processing....
CM412503I CEDA Diagnostics Code . : 00000000
CM412504I CEDA Execution Code . . : 00000000
CM412502I DPL_CALL to CM411 complete. Return Code 00000000

CM412902I Number of commands processed 00000003
CM412901I Program Terminated. Highest Return Code 00000000
************************************************************************

CM420502I Control returned from sub-program CM412    RC=00000000

CM420601I CLOSE_PIPE processing....
CM420602I CLOSE_PIPE successful.

CM420701I DEALLOCATE_PIPE processing....
CM420702I DEALLOCATE_PIPE successful.

CM420901I Program Terminated. Highest Return Code 00000000
```

If any EXCI or DPL errors occur, appropriate messages would be written to SYSPRINT and subsequent CEDA commands would not be processed.

CM412 and all EXCI client programs must include the CICS-supplied EXCI program stub DFHXCSTB. You can use the CICS-supplied procedure DFHEXTAL to assemble and link-edit CM412 and all EXCI client programs written in Assembler. Procedures are also supplied for EXCI client programs written in other languages.

```
*ASM XOPTS(EXCI)
CM412    TITLE 'CM412 : EXCI CEDA BATCH CLIENT'
*********************************************************************
*                 C A R L   W A D E   M C B U R N I E               *
*                   -  I T   C O N S U L T A N T  -                 *
*                                                                   *
*                             www.cwmit.com                         *
*********************************************************************
* MODULE NAME = CM412                                               *
* MODULE TYPE = CSECT  (Sub-Program)                                *
* DESCRIPTION = CICS/ESA EXCI CEDA Batch Client Program             *
*                                                                   *
*               This program must be passed a parameter list        *
*               in Register 1 :-                                    *
*                                                                   *
*               Address 1 :  CM419COM                               *
*               Address 2 :  CM420COM                               *
*                                                                   *
*               This program reads CEDA commands from the           *
*               SYSIN dataset and issues a DPL call to the          *
*               CICS EXCI CEDA Server Program (CM411) for           *
*               each command.                                       *
*********************************************************************
         EJECT
*********************************************************************
*                                                                   *
* CHANGE HISTORY:                                                   *
* --------------                                                    *
*********************************************************************
         EJECT
*********************************************************************
*                                                                   *
* REGISTER  EQUATES                      USAGE                      *
* --------  --------  ------------------------------------------- *
*                                                                   *
* REG  0    R0        Work Register                                 *
* REG  1    R1        Work Register                                 *
* REG  2    R2        DSECT - CM411COM                              *
* REG  3    BASE      Base Register for CSECT CM412                 *
* REG  4    R4        Work Register                                 *
* REG  5    R5        Work Register                                 *
```

```
* REG  6    R6        Work Register                                      *
* REG  7    R7        Work Register                                      *
* REG  8    R8        DSECT - CM419COM                                   *
* REG  9    R9        DSECT - CM42ØCOM                                   *
* REG 1Ø    R1Ø       DSECT - EXCI_RETURN_CODE                           *
* REG 11    R11       DSECT - EXCI_DPL_RETAREA                           *
* REG 12    R12                                                          *
* REG 13    DYNREG    DSECT - DFHEISTG                                   *
* REG 14    R14       Linkage - return address                           *
* REG 15    R15       Linkage - branch to address                        *
*                     Return Codes                                       *
************************************************************************
         EJECT
*----------------------------------------------------------------------*
*- Copybooks                                                          -*
*----------------------------------------------------------------------*
         EJECT
         COPY  DFHXCRCD               EXCI DSECTS AND RETURN CODES
         COPY  DFHXCPLD               EXCI PARAMTER LIST EQUATES
         COPY  CM411AØ1               DSECT - CM411COM  (COMMAREA)
         COPY  CM419AØ1               DSECT - CM419COM  (COMMAREA)
         COPY  CM42ØAØ1               DSECT - CM42ØCOM  (COMMAREA)
         EJECT
*----------------------------------------------------------------------*
*- Addressability to DFHEISTG will be established by CICS EXCI.      -*
*----------------------------------------------------------------------*
         DFHEISTG                     CICS EXCI DYNAMIC STORAGE
         EJECT
*----------------------------------------------------------------------*
*- CM412 Dynamic Storage - Start                                      -*
*----------------------------------------------------------------------*
DYNSTOR  DS    ØH                     CM412 DYNAMIC STORAGE
*
PLISTPTR DS    F                      --> PARM LIST
CM419ST  DS    F                      --> CM419COM STORAGE
CM42ØST  DS    F                      --> CM42ØCOM STORAGE
*
COMD_CNTR DS   F                      COMMAND COUNTER
COMD_FLAG DS   X                      COMMAND FLAG
FLAG_ON  EQU   X'FF'                  FLAG ON
FLAG_OFF EQU   X'ØØ'                  FLAG OFF
*
HIGH_RC  DS    F                      HIGHEST RETURN CODE
CURR_RC  DS    F                      CURRENT RETURN CODE
*
WORKØ5   DS    CL5                    WORK AREA 5 BYTES
WORKØ9   DS    CL9                    WORK AREA 9 BYTES
WORKDW_1 DS    D                      WORK AREA DOUBLE WORD
WORKDW_2 DS    D                      WORK AREA DOUBLE WORD
*
```

```
A100SR14 DS    F                       SAVE REGISTER 14
A500SR14 DS    F                       SAVE REGISTER 14
A900SR14 DS    F                       SAVE REGISTER 14
Z200SR14 DS    F                       SAVE REGISTER 14
Z300SR14 DS    F                       SAVE REGISTER 14
Z400SR14 DS    F                       SAVE REGISTER 14
Z500SR14 DS    F                       SAVE REGISTER 14
Z600SR14 DS    F                       SAVE REGISTER 14
Z700SR14 DS    F                       SAVE REGISTER 14
*
         DS    0D                      ALIGN STORAGE
CM411ST  DS    CL(CM411COM_LENGTH)     STORAGE FOR CM411COM
*
LINK_PL  CALL ,                                                        X
               (CM419ST,                                               X
               CM420ST),                                               X
               VL,                                                     X
               MF=L
*
DYNSTORL EQU   *-DYNSTOR               LENGTH OF DYNAMIC STORAGE
*-----------------------------------------------------------------------*
*- CM412 Dynamic Storage - End                                         -*
*-----------------------------------------------------------------------*
         EJECT
*-----------------------------------------------------------------------*
*- Register Equates                                                    -*
*-----------------------------------------------------------------------*
         DFHREGS                       CICS STANDARD EQUATES
BASE     EQU   3                       BASE CODE REGISTER
DYNREG   EQU   13                      DYNAMIC STORAGE REGISTER
         EJECT
************************************************************************
*=====================================================================*
*=       E  N  T  R  Y          P  O  I  N  T                        =*
*=====================================================================*
************************************************************************
CM412    DFHEIENT CODEREG=(BASE),DATAREG=(DYNREG)
CM412    AMODE 31
CM412    RMODE 24
*-----------------------------------------------------------------------*
*- Program Identification "Eye-Catchers"                               -*
*-----------------------------------------------------------------------*
         B     A000_MAINLINE           BRANCH OVER EYE-CATCHERS
ASMEYE   DC    C'*'                    ASTERISK
ASMPROG  DC    C'CM412   '             PROGRAM NAME
         DC    C'-'                    HYPHEN
ASMLVL   DC    C'CWM00001'             PROGRAM LEVEL
         DC    C' '                    BLANK
ASMDATE  DC    C'&SYSDATE'             DATE OF ASSEMBLY
         DC    C' '                    BLANK
```

```
ASMTIME  DC    C'&SYSTIME'             TIME OF ASSEMBLY
         DC    C'*'                    ASTERISK
ASMEYEL  EQU   *-ASMEYE                LENGTH OF EYE-CATCHER
*
         EJECT
*********************************************************************
*-------------------------------------------------------------------*
*- A Ø Ø Ø _ M A I N L I N E :  Controls the flow of the program    -*
*-------------------------------------------------------------------*
*********************************************************************
*- R3   BASE CSECT CM412                                           -*
*- R13  DYNREG DSECT DFHEISTG                                      -*
*- R14  Linkage                                                    -*
*- R15  Return Code                                                -*
*********************************************************************
A000_MAINLINE  DS  0H
         BAL   R14,A100_INITIALIZE     PERFORM INITIALIZATION
         CLC   HIGH_RC,=F'4'             IF RC > 4
         BH    A000TERM                    THEN TERMINATE
         BAL   R14,A500_SUB_PROG       EXCI CEDA SUB PROGRAM
A000TERM BAL   R14,A900_TERMINATION    PERFORM TERMINATION
*********************************************************************
*-------------------------------------------------------------------*
*- A Ø Ø Ø _ R E T U R N :  Return Control                         -*
*-------------------------------------------------------------------*
*********************************************************************
A000_RETURN  DS  0H
*
         L     R15,HIGH_RC             LOAD HIGHEST RETURN CODE
RETURN   DFHEIRET RCREG=(15)
*********************************************************************
*===================================================================*
*=       E  X  I  T             P  O  I  N  T                      =*
*===================================================================*
*********************************************************************
         EJECT
*********************************************************************
*-------------------------------------------------------------------*
*- A 1 Ø Ø _ I N I T I A L I Z E :  Perform Initialization        -*
*-------------------------------------------------------------------*
*********************************************************************
*- R1   Parameter List                                             -*
*- R2   DSECT CM411COM                                             -*
*- R3   BASE CSECT CM412                                           -*
*- R4   Work Register                                              -*
*- R5   Work Register                                              -*
*- R6   Work Register                                              -*
*- R7   Work Register                                              -*
*- R8   DSECT CM419COM                                             -*
*- R9   DSECT CM420COM                                             -*
```

```
*- R1Ø  DSECT EXCI_RETURN_CODE                                          -*
*- R11  DSECT EXCI_DPL_RETAREA                                          -*
*- R13  DYNREG DSECT DFHEISTG                                           -*
*- R14  Linkage                                                         -*
************************************************************************
A1ØØ_INITIALIZE  DS  ØH
*----------------------------------------------------------------------*
*- Clear DYNSTOR.                                                      -*
*----------------------------------------------------------------------*
         LA    R4,DYNSTOR              ADDRESS DYNSTOR
         LA    R5,DYNSTORL             LENGTH OF DYNSTOR
         XR    R6,R6                   FROM ADDRESS NOT REQUIRED
         XR    R7,R7                   SET LENGTH TO Ø
         ICM   R7,8,=C' '              SET PADDING TO BLANKS
         MVCL  R4,R6                   CLEAR STORAGE TO BLANKS
*
         ST    R1,PLISTPTR             SAVE PARMLIST PTR FOR LATER
         ST    R14,A1ØØSR14            SAVE REGISTER 14
         XC    HIGH_RC,HIGH_RC         SET HIGHEST RC TO Ø
         XC    CURR_RC,CURR_RC         SET CURRENT RC TO Ø
*----------------------------------------------------------------------*
*- Establish addressability and map CM411COM.                          -*
*----------------------------------------------------------------------*
         LA    R2,CM411ST              ADDRESS CM411COM
         USING CM411COM,R2             MAP CM411COM
         MVC   CM411COM_EYECATCH,=C'<<  CM411COM  >>'  EYECATCHER
*----------------------------------------------------------------------*
*- Process PARMLIST and establish addressability to                    -*
*- CM419COM and CM42ØCOM.                                              -*
*----------------------------------------------------------------------*
         LM    R8,R9,Ø(R1)             --> CM419COM & CM42ØCOM
         ST    R8,CM419ST              SAVE --> CM419COM
         USING CM419COM,R8             MAP CM419COM
         ST    R9,CM42ØST              SAVE --> CM42ØCOM
         USING CM42ØCOM,R9             MAP CM42ØCOM
*
         LA    R1Ø,CM42ØCOM_RETURN_AREA --> RETURN_AREA STORAGE
         USING EXCI_RETURN_CODE,R1Ø         MAP RETURN_AREA
*
         LA    R11,CM42ØCOM_DPL_RETAREA --> DPL_RETAREA STORAGE
         USING EXCI_DPL_RETAREA,R11         MAP DPL_RETAREA
*----------------------------------------------------------------------*
*- Write start message using programs eyecatcher as text.              -*
*----------------------------------------------------------------------*
A1ØØSMSG DS    ØH
         BAL   R14,Z2ØØ_WRITE_SYSPRINT  BLANK LINE
         MVC   CM419COM_SYSPRINT_MSG,=C'CM412ØØØI'     MOVE MSG. NO.
         MVC   CM419COM_SYSPRINT_DATA(ASMEYEL),ASMEYE  MOVE EYECATCHER
         BAL   R14,Z2ØØ_WRITE_SYSPRINT  WRITE MSG TO SYSPRINT
         BAL   R14,Z2ØØ_WRITE_SYSPRINT  BLANK LINE
```

```
*--------------------------------------------------------------------*
*- Issue CM412 loaded message. Convert the EP address to             -*
*- displayable characters.                                           -*
*--------------------------------------------------------------------*
A100412P DS    0H                      ISSUE MESSAGE
         MVC   CM419COM_SYSPRINT_MSG,=C'CM412002I'     MOVE MSG. NO.
         MVC   CM419COM_SYSPRINT_DATA(L'CM412002I),CM412002I  & TEXT
         MVC   CM419COM_SYSPRINT_DATA+8(8),=C'CM412   '       & PGM
         ST    BASE,WORK05             EPA IN WORK FIELD
         UNPK  WORK09,WORK05           UNPACK ADDRESS
         MVC   WORKDW_1(L'WORKDW_1),WORK09      MOVE REQ. BYTES (8)
         TR    WORKDW_1,TRANTAB0-240   TRANSLATE REQ. BYTES
         MVC   CM419COM_SYSPRINT_DATA+37(L'WORKDW_1),WORKDW_1  MOVE
         BAL   R14,Z200_WRITE_SYSPRINT  WRITE MSG TO SYSPRINT
*--------------------------------------------------------------------*
*- Issue WORKING storage message. Convert the address and length to  -*
*- displayable characters.                                           -*
*--------------------------------------------------------------------*
A100WORK DS    0H                      ISSUE MESSAGE
         MVC   CM419COM_SYSPRINT_MSG,=C'CM412003I'     MOVE MSG. NO.
         MVC   CM419COM_SYSPRINT_DATA(L'CM412003I),CM412003I  & TEXT
         MVC   CM419COM_SYSPRINT_DATA(8),=C'WORKING '         & TEXT
         LA    R4,DYNSTOR              --> DYNSTOR
         ST    R4,WORK05               SAVE IN WORK FIELD
         UNPK  WORK09,WORK05           UNPACK ADDRESS
         MVC   WORKDW_1(L'WORKDW_1),WORK09      MOVE REQ. BYTES (8)
         TR    WORKDW_1,TRANTAB0-240   TRANSLATE REQ. BYTES
         MVC   CM419COM_SYSPRINT_DATA+27(L'WORKDW_1),WORKDW_1  MOVE
         LA    R4,DYNSTORL             LOAD LENGTH OF DYNSTOR
         CVD   R4,WORKDW_1             CONVERT TO DECIMAL
         UNPK  WORKDW_2,WORKDW_1       CONVERT TO ...
         OI    WORKDW_2+7,X'F0'        ... DISPLAYABLE DECIMAL
         MVC   CM419COM_SYSPRINT_DATA+44(L'WORKDW_2),WORKDW_2  MOVE
         BAL   R14,Z200_WRITE_SYSPRINT  WRITE MSG TO SYSPRINT
*--------------------------------------------------------------------*
*- Issue CM411COM storage message. Convert the address and length to -*
*- displayable characters.                                           -*
*--------------------------------------------------------------------*
A100411S DS    0H                      ISSUE MESSAGE
         MVC   CM419COM_SYSPRINT_MSG,=C'CM412003I'     MOVE MSG. NO.
         MVC   CM419COM_SYSPRINT_DATA(L'CM412003I),CM412003I  & TEXT
         MVC   CM419COM_SYSPRINT_DATA(8),=C'CM411COM'         & TEXT
         LA    R4,CM411ST              --> CM411COM STORAGE
         ST    R4,WORK05               SAVE IN WORK FIELD
         UNPK  WORK09,WORK05           UNPACK ADDRESS
         MVC   WORKDW_1(L'WORKDW_1),WORK09      MOVE REQ. BYTES (8)
         TR    WORKDW_1,TRANTAB0-240   TRANSLATE REQ. BYTES
         MVC   CM419COM_SYSPRINT_DATA+27(L'WORKDW_1),WORKDW_1  MOVE
         LA    R4,CM411COM_LENGTH      LOAD LENGTH OF CM411COM
         CVD   R4,WORKDW_1             CONVERT TO DECIMAL
```

```
         UNPK  WORKDW_2,WORKDW_1       CONVERT TO ...
         OI    WORKDW_2+7,X'F0'        ... DISPLAYABLE DECIMAL
         MVC   CM419COM_SYSPRINT_DATA+44(L'WORKDW_2),WORKDW_2  MOVE
         BAL   R14,Z200_WRITE_SYSPRINT  WRITE MSG TO SYSPRINT
*--------------------------------------------------------------------*
*- Return to caller                                                  -*
*--------------------------------------------------------------------*
A100RET  BAL   R14,Z400_CHECK_RC       CHECK HIGHEST RC
         L     R14,A100SR14            RESTORE REGISTER 14
         BR    R14                     RETURN TO CALLER
         EJECT
*********************************************************************
*--------------------------------------------------------------------*
*- A 5 0 0 _ S U B _ P R O G  :  Sub-Program for DPL CEDA call       -*
*--------------------------------------------------------------------*
*********************************************************************
*- R2   DSECT CM411COM                                               -*
*- R3   BASE CSECT CM412                                             -*
*- R5   CEDA Command Buffer                                          -*
*- R6   CEDA Command Length                                          -*
*- R8   DSECT CM419COM                                               -*
*- R12  Work Register (Command Counter)                              -*
*- R13  DYNREG DSECT DFHEISTG                                        -*
*- R14  Linkage                                                      -*
*********************************************************************
A500_SUB_PROG  DS  0H
*--------------------------------------------------------------------*
*- Read SYSIN records and perform validation in the following        -*
*- order, the validation of the command content is performed         -*
*- by DFHEDAP in the CICS region:                                    -*
*-                                                                   -*
*- 1. Check for End-Of-File                                          -*
*-    If there is a command in the command buffer submit it         -*
*-    for processing before terminating.                             -*
*- 2. Check for comment line, ie an asterisk "*" in column 1.        -*
*-    If there is a command in the command buffer submit it         -*
*-    for processing before writing the comment line to              -*
*-    SYSPRINT.                                                      -*
*- 3. Check for blank line, ie columns 1 - 72 contain blanks.        -*
*-    If there is a command in the command buffer submit it         -*
*-    for processing before writing a blank line to                  -*
*-    SYSPRINT.                                                      -*
*- 4. Check for a new command, ie columns 1 - 10 contain             -*
*-    non-blank characters.                                          -*
*-    If there is already a command in the command buffer            -*
*-    submit it for processing before writing the new command        -*
*-    to the command buffer and SYSPRINT.                            -*
*- 5. Check for a command continuation (assumed), ie none of         -*
*-    the other conditions are met and there is a current            -*
*-    command in the command buffer (if there is no command          -*
```

```
*-    in the command buffer a RC=Ø8 will be set).                    -*
*-    Write the command continuation to the command buffer.          -*
*-                                                                   -*
*- Any return code greater than 4 will cause all SYSIN records      -*
*- to be written to SYSPRINT without any further processing of      -*
*- commands.                                                        -*
*-------------------------------------------------------------------*
         ST    R14,A5ØØSR14            SAVE REGISTER 14
         XC    CURR_RC,CURR_RC         CLEAR CURRENT RC
         BAL   R14,Z2ØØ_WRITE_SYSPRINT BLANK LINE
         LA    R12,Ø                   SET COMMAND COUNTER TO ZERO
*-------------------------------------------------------------------*
*- Loop through the SYSIN records.                                  -*
*-------------------------------------------------------------------*
A5ØØLOOP BAL   R14,Z5ØØ_READ_SYSIN     READ INPUT RECORD
         CLI   CM419COM_SYSIN_EOF_F,CM419COM_SYSIN_EOF  IF EOF
         BE    A5ØØEOF                                    THEN LEAVE
*-------------------------------------------------------------------*
*- Check for a comment line.                                        -*
*-------------------------------------------------------------------*
A5ØØCMNT CLC   CM419COM_SYSIN_DATA(L'COMMENT),COMMENT  COMMENT LINE ?
         BNE   A5ØØBLNK                               NO  - NEXT CHECK
*                                                     YES - OUTPUT
         CLI   COMD_FLAG,FLAG_ON       COMMAND WAITING ?
         BNE   A5ØØOUTC                NO  - OUTPUT COMMENT LINE
         BAL   R14,Z6ØØ_LINK_CEDA      YES - PROCESS COMMAND FIRST
         CLC   HIGH_RC,=F'4'             IF RC > 4
         BH    A5ØØERRO                    THEN PROCESS ERROR
A5ØØOUTC MVC    CM419COM_SYSPRINT_DATA(L'CM419COM_SYSIN_DATA),CM419COM_SX
               YSIN_DATA               MOVE COMMENT LINE
         BAL   R14,Z2ØØ_WRITE_SYSPRINT  WRITE COMMENT TO SYSPRINT
         B     A5ØØLOOP                LOOP BACK FOR NEXT RECORD
*-------------------------------------------------------------------*
*- Check for a blank line.                                          -*
*-------------------------------------------------------------------*
A5ØØBLNK CLC   CM419COM_SYSIN_DATA(L'BLANKS),BLANKS    BLANK LINE ?
         BNE   A5ØØCOMD                               NO  - NEXT CHECK
*                                                     YES - OUTPUT
         CLI   COMD_FLAG,FLAG_ON       COMMAND WAITING ?
         BNE   A5ØØOUTB                NO  - OUTPUT BLANK LINE
         BAL   R14,Z6ØØ_LINK_CEDA      YES - PROCESS COMMAND FIRST
         CLC   HIGH_RC,=F'4'             IF RC > 4
         BH    A5ØØERRO                    THEN PROCESS ERROR
A5ØØOUTB BAL   R14,Z2ØØ_WRITE_SYSPRINT WRITE BLANK LINT TO SYSPRINT
         B     A5ØØLOOP                LOOP BACK FOR NEXT RECORD
*-------------------------------------------------------------------*
*- Check for a new command.                                         -*
*-------------------------------------------------------------------*
A5ØØCOMD CLC   CM419COM_SYSIN_DATA(1Ø),BLANKS          NEW COMMAND ?
         BE    A5ØØCONT                               NO  - NEXT CHECK
```

```
*                                                        YES - OUTPUT
         CLI   COMD_FLAG,FLAG_ON       COMMAND WAITING ?
         BNE   A500COUT                NO  - OUTPUT COMMAND LINE
         BAL   R14,Z600_LINK_CEDA      YES - PROCESS COMMAND FIRST
         CLC   HIGH_RC,=F'4'             IF RC > 4
         BH    A500ERRO                    THEN PROCESS ERROR
A500COUT MVI   COMD_FLAG,FLAG_ON       SET COMMAND FLAG ON
         A     R12,=F'1'               INCREMENT COMMAND COUNTER
         LA    R5,CM411COM_COMMAND     --> COMMAND FIELD
         LA    R6,0                    SET COMMAND LENGTH TO ZERO
*                                      MOVE COMMAND
         MVC   0(L'CM419COM_SYSIN_DATA,R5),CM419COM_SYSIN_DATA
         LA    R5,L'CM419COM_SYSIN_DATA(R5) INCREMENT COMMAND POINTER
         LA    R6,L'CM419COM_SYSIN_DATA(R6) INCREMENT COMMAND LENGTH
         MVC   CM419COM_SYSPRINT_DATA(L'CM419COM_SYSIN_DATA),CM419COM_SX
               YSIN_DATA               MOVE COMMAND LINE
         BAL   R14,Z200_WRITE_SYSPRINT  WRITE COMMAND TO SYSPRINT
         B     A500LOOP                LOOP BACK FOR NEXT RECORD
*-------------------------------------------------------------------*
*- Check for a command continuation.                               -*
*-------------------------------------------------------------------*
A500CONT CLI   COMD_FLAG,FLAG_ON       COMMAND CONTINUATION ?
         BNE   A500INER                NO  - INPUT ERROR
*                                      YES - OUTPUT
*                                      MOVE COMMAND CONTINUATION
         MVC   0(L'CM419COM_SYSIN_DATA,R5),CM419COM_SYSIN_DATA
         LA    R5,L'CM419COM_SYSIN_DATA(R5) INCREMENT COMMAND POINTER
         LA    R6,L'CM419COM_SYSIN_DATA(R6) INCREMENT COMMAND LENGTH
         MVC   CM419COM_SYSPRINT_DATA(L'CM419COM_SYSIN_DATA),CM419COM_SX
               YSIN_DATA               MOVE COMMAND CONTINUATION
         BAL   R14,Z200_WRITE_SYSPRINT  WRITE COMMAND TO SYSPRINT
         B     A500LOOP                LOOP BACK FOR NEXT RECORD
*-------------------------------------------------------------------*
*- Error record - all checks failed.                               -*
*-------------------------------------------------------------------*
A500INER MVC   CURR_RC,=F'8'           RETURN CODE 8
         BAL   R14,Z200_WRITE_SYSPRINT  BLANK LINE
         MVC   CM419COM_SYSPRINT_MSG,=C'CM412506E'     MOVE MSG. NO.
         MVC   CM419COM_SYSPRINT_DATA(L'CM412506E),CM412506E    & TEXT
         BAL   R14,Z200_WRITE_SYSPRINT  WRITE MSG TO SYSPRINT
         BAL   R14,Z200_WRITE_SYSPRINT  BLANK LINE
*-------------------------------------------------------------------*
*- Return code greater than 4 encountered. Write any remaining      -*
*- SYSIN records to SYSPRINT.                                       -*
*-------------------------------------------------------------------*
A500ERRO CLI   CM419COM_SYSIN_EOF_F,CM419COM_SYSIN_EOF  ALREADY EOF ?
         BE    A500EOF                                  YES - END
*                                                       NO - OUTPUT ALL
A500READ MVC   CM419COM_SYSPRINT_DATA(L'CM419COM_SYSIN_DATA),CM419COM_SX
               YSIN_DATA               MOVE RECORD
```

```
         BAL   R14,Z200_WRITE_SYSPRINT  WRITE RECORD TO SYSPRINT
         BAL   R14,Z500_READ_SYSIN     READ INPUT RECORD
         CLI   CM419COM_SYSIN_EOF_F,CM419COM_SYSIN_EOF  IF EOF
         BE    A500EOF                                    THEN LEAVE
         B     A500READ                LOOP BACK FOR NEXT RECORD
A500EOF  CLI   COMD_FLAG,FLAG_ON       COMMAND WAITING ?
         BNE   A500EOF2                NO  - CONTINUE EOF PROCESS
         BAL   R14,Z600_LINK_CEDA      YES - PROCESS COMMAND FIRST
A500EOF2 ST    R12,COMD_CNTR           STORE COMMAND COUNTER
         C     R12,=F'0'                 IF COMMAND COUNTER > ZERO
         BH    A500RET                     THEN RETURN
         MVC   CURR_RC,=F'8'               ELSE SET RC = 08
*-------------------------------------------------------------------*
*- Return to caller                                                 -*
*-------------------------------------------------------------------*
A500RET  BAL   R14,Z400_CHECK_RC       CHECK HIGHEST RC
         L     R14,A500SR14            RESTORE REGISTER 14
         BR    R14                     RETURN TO CALLER
         EJECT
*********************************************************************
*-------------------------------------------------------------------*
*- A 9 0 0 _ T E R M I N A T I O N  :  Perform Termination         -*
*-------------------------------------------------------------------*
*********************************************************************
*- R3   BASE CSECT CM412                                            -*
*- R4   Work Register                                               -*
*- R8   DSECT CM419COM                                              -*
*- R13  DYNREG DSECT DFHEISTG                                       -*
*- R14  Linkage                                                     -*
*********************************************************************
A900_TERMINATION  DS  0H
         ST    R14,A900SR14            SAVE REGISTER 14
         XC    CURR_RC,CURR_RC         CLEAR CURRENT RC
         BAL   R14,Z200_WRITE_SYSPRINT BLANK LINE
*-------------------------------------------------------------------*
*- Convert number of commands processed to displayable decimal      -*
*- characters and issue message.                                    -*
*-------------------------------------------------------------------*
         L     R4,COMD_CNTR            LOAD COMMAND COUNTER
         CVD   R4,WORKDW_1             CONVERT TO DECIMAL
         UNPK  WORKDW_2,WORKDW_1       CONVERT TO ...
         OI    WORKDW_2+7,X'F0'        ... DISPLAYABLE DECIMAL
         MVC   CM419COM_SYSPRINT_MSG,=C'CM412902I'     MOVE MSG. NO.
         MVC   CM419COM_SYSPRINT_DATA(L'CM412902I),CM412902I    & TEXT
         MVC   CM419COM_SYSPRINT_DATA+29(L'WORKDW_2),WORKDW_2   & RC
         BAL   R14,Z200_WRITE_SYSPRINT  WRITE MSG TO SYSPRINT
*-------------------------------------------------------------------*
*- Convert RC to displayable decimal characters and issue message.  -*
*-------------------------------------------------------------------*
         L     R4,HIGH_RC              LOAD RC
```

```
         CVD   R4,WORKDW_1             CONVERT TO DECIMAL
         UNPK  WORKDW_2,WORKDW_1       CONVERT TO ...
         OI    WORKDW_2+7,X'FØ'        ... DISPLAYABLE DECIMAL
         MVC   CM419COM_SYSPRINT_MSG,=C'CM412ØØ1I'     MOVE MSG. NO.
         MVC   CM419COM_SYSPRINT_DATA(L'CM412ØØ1I),CM412ØØ1I    & TEXT
         MVC   CM419COM_SYSPRINT_DATA+4Ø(L'WORKDW_2),WORKDW_2   & RC
         BAL   R14,Z2ØØ_WRITE_SYSPRINT  WRITE MSG TO SYSPRINT
*
         BAL   R14,Z2ØØ_WRITE_SYSPRINT  BLANK LINE
*--------------------------------------------------------------------*
*- Return to caller                                                  -*
*--------------------------------------------------------------------*
A9ØØRET  BAL   R14,Z4ØØ_CHECK_RC       CHECK HIGHEST RC
         L     R14,A9ØØSR14            RESTORE REGISTER 14
         BR    R14                     RETURN TO CALLER
         EJECT
**********************************************************************
*--------------------------------------------------------------------*
*- Z 2 Ø Ø _ W R I T E _ S Y S P R I N T :  Write message            -*
*--------------------------------------------------------------------*
**********************************************************************
*- R3   BASE CSECT CM412                                             -*
*- R8   DSECT CM419COM                                               -*
*- R13  DYNREG DSECT DFHEISTG                                        -*
*- R14  Linkage                                                      -*
**********************************************************************
Z2ØØ_WRITE_SYSPRINT  DS  ØH
*
         ST    R14,Z2ØØSR14            SAVE REGISTER 14
*
         MVI   CM419COM_FUNCTION,CM419COM_PUT    SET FUNCTION
*--------------------------------------------------------------------*
*- LINK to CM419 passing CM419COM. If the LINK doesn't work the      -*
*- program will abend.                                               -*
*--------------------------------------------------------------------*
Z2ØØLINK LINK EP=CM419,PARAM=((R8)),MF=(E,LINK_PL)
*--------------------------------------------------------------------*
*- The return code from CM419 is not saved as a current return       -*
*- code because this would have a negative affect on the current     -*
*- and high return codes logic. Any I/O errors will be covered by    -*
*- system abends.                                                    -*
*--------------------------------------------------------------------*
*        ST    R15,CURR_RC             SAVE RETURN CODE
*--------------------------------------------------------------------*
*- Return to caller                                                  -*
*--------------------------------------------------------------------*
Z2ØØRET  L     R14,Z2ØØSR14            RESTORE REGISTER 14
         BR    R14                     RETURN TO CALLER
         EJECT
**********************************************************************
```

```
*-------------------------------------------------------------------*
*- Z 3 Ø Ø _ E X C I _ D I A G N O S T I C S  :  Diagnostics      -*
*-------------------------------------------------------------------*
*********************************************************************
*- R3   BASE CSECT CM412                                           -*
*- R4   Work Register                                              -*
*- R8   DSECT CM419COM                                             -*
*- R1Ø  DSECT EXCI_RETURN_CODE                                     -*
*- R13  DYNREG DSECT DFHEISTG                                      -*
*- R14  Linkage                                                    -*
*********************************************************************
Z3ØØ_EXCI_DIAGNOSTICS  DS  ØH
*
         ST    R14,Z3ØØSR14            SAVE REGISTER 14
*-------------------------------------------------------------------*
*- Write EXCI diagnostics follow message.                          -*
*-------------------------------------------------------------------*
         MVC   CM419COM_SYSPRINT_MSG,=C'CM412Ø1ØE'     MOVE MSG. NO.
         MVC   CM419COM_SYSPRINT_DATA(L'CM412Ø1ØE),CM412Ø1ØE  & TEXT
         BAL   R14,Z2ØØ_WRITE_SYSPRINT  WRITE MSG TO SYSPRINT
*-------------------------------------------------------------------*
*- Convert EXCI_RESPONSE to displayable decimal characters         -*
*- and issue message.                                              -*
*-------------------------------------------------------------------*
         L     R4,EXCI_RESPONSE        LOAD RESPONSE
         CVD   R4,WORKDW_1             CONVERT TO DECIMAL
         UNPK  WORKDW_2,WORKDW_1       CONVERT TO ...
         OI    WORKDW_2+7,X'FØ'        ... DISPLAYABLE DECIMAL
         MVC   CM419COM_SYSPRINT_MSG,=C'CM412Ø1ØE'     MOVE MSG. NO.
         MVC   CM419COM_SYSPRINT_DATA+6(L'CM412Ø1ØA),CM412Ø1ØA  & TEXT
         MVC   CM419COM_SYSPRINT_DATA+28(L'WORKDW_2),WORKDW_2   & CODE
         BAL   R14,Z2ØØ_WRITE_SYSPRINT  WRITE MSG TO SYSPRINT
*-------------------------------------------------------------------*
*- Convert EXCI_REASON to displayable decimal characters           -*
*- and issue message.                                              -*
*-------------------------------------------------------------------*
         L     R4,EXCI_REASON          LOAD REASON
         CVD   R4,WORKDW_1             CONVERT TO DECIMAL
         UNPK  WORKDW_2,WORKDW_1       CONVERT TO ...
         OI    WORKDW_2+7,X'FØ'        ... DISPLAYABLE DECIMAL
         MVC   CM419COM_SYSPRINT_MSG,=C'CM412Ø1ØE'     MOVE MSG. NO.
         MVC   CM419COM_SYSPRINT_DATA+6(L'CM412Ø1ØB),CM412Ø1ØB  & TEXT
         MVC   CM419COM_SYSPRINT_DATA+28(L'WORKDW_2),WORKDW_2   & CODE
         BAL   R14,Z2ØØ_WRITE_SYSPRINT  WRITE MSG TO SYSPRINT
*-------------------------------------------------------------------*
*- Convert EXCI_SUB_REASON1 to displayable decimal characters      -*
*- and issue message.                                              -*
*-------------------------------------------------------------------*
         L     R4,EXCI_SUB_REASON1     LOAD SUB-REASON 1
         CVD   R4,WORKDW_1             CONVERT TO DECIMAL
```

```
         UNPK  WORKDW_2,WORKDW_1       CONVERT TO ...
         OI    WORKDW_2+7,X'FØ'        ... DISPLAYABLE DECIMAL
         MVC   CM419COM_SYSPRINT_MSG,=C'CM412Ø1ØE'     MOVE MSG. NO.
         MVC   CM419COM_SYSPRINT_DATA+6(L'CM412Ø1ØC),CM412Ø1ØC  & TEXT
         MVC   CM419COM_SYSPRINT_DATA+28(L'WORKDW_2),WORKDW_2   & CODE
         BAL   R14,Z2ØØ_WRITE_SYSPRINT  WRITE MSG TO SYSPRINT
*-------------------------------------------------------------------*
*- Convert EXCI_SUB_REASON2 to displayable decimal characters       -*
*- and issue message.                                               -*
*-------------------------------------------------------------------*
         L     R4,EXCI_SUB_REASON2     LOAD SUB-REASON 2
         CVD   R4,WORKDW_1             CONVERT TO DECIMAL
         UNPK  WORKDW_2,WORKDW_1       CONVERT TO ...
         OI    WORKDW_2+7,X'FØ'        ... DISPLAYABLE DECIMAL
         MVC   CM419COM_SYSPRINT_MSG,=C'CM412Ø1ØE'     MOVE MSG. NO.
         MVC   CM419COM_SYSPRINT_DATA+6(L'CM412Ø1ØD),CM412Ø1ØD  & TEXT
         MVC   CM419COM_SYSPRINT_DATA+28(L'WORKDW_2),WORKDW_2   & CODE
         BAL   R14,Z2ØØ_WRITE_SYSPRINT  WRITE MSG TO SYSPRINT
*-------------------------------------------------------------------*
*- Return to caller                                                 -*
*-------------------------------------------------------------------*
Z3ØØRET  L     R14,Z3ØØSR14            RESTORE REGISTER 14
         BR    R14                     RETURN TO CALLER
         EJECT
*********************************************************************
*-------------------------------------------------------------------*
*- Z 4 Ø Ø _ C H E C K _ R C  :  Check Return Codes                 -*
*-------------------------------------------------------------------*
*********************************************************************
*- R3   BASE CSECT CM412                                            -*
*- R13  DYNREG DSECT DFHEISTG                                       -*
*- R14  Linkage                                                     -*
*********************************************************************
Z4ØØ_CHECK_RC  DS  ØH
*
         ST    R14,Z4ØØSR14            SAVE REGISTER 14
*
         CLC   CURR_RC,HIGH_RC         IF CURRENT RC > HIGHEST RC
         BH    Z4ØØHIGH                   THEN RESET HIGHEST RC
         B     Z4ØØRET                    ELSE RETURN
Z4ØØHIGH MVC   HIGH_RC,CURR_RC         MOVE CURRENT RC TO HIGHEST RC
*-------------------------------------------------------------------*
*- Return to caller                                                 -*
*-------------------------------------------------------------------*
Z4ØØRET  L     R14,Z4ØØSR14            RESTORE REGISTER 14
         BR    R14                     RETURN TO CALLER
         EJECT
*********************************************************************
*-------------------------------------------------------------------*
*- Z 5 Ø Ø _ R E A D _ S Y S I N  :  Read SYSIN                     -*
```

```
*--------------------------------------------------------------------*
**********************************************************************
*- R3   BASE CSECT CM412                                            -*
*- R8   DSECT CM419COM                                              -*
*- R13  DYNREG DSECT DFHEISTG                                       -*
*- R14  Linkage                                                     -*
**********************************************************************
Z500_READ_SYSIN  DS  0H
*
         ST    R14,Z500SR14            SAVE REGISTER 14
*
         MVI   CM419COM_FUNCTION,CM419COM_GET    SET FUNCTION
*--------------------------------------------------------------------*
*- LINK to CM419 passing CM419COM. If the LINK doesn't work the      -*
*- program will abend.                                              -*
*--------------------------------------------------------------------*
Z500LINK LINK EP=CM419,PARAM=((R8)),MF=(E,LINK_PL)
*--------------------------------------------------------------------*
*- The return code from CM419 is not saved as a current return      -*
*- code because this would have a negative effect on the current    -*
*- and high return codes logic. Any I/O errors will be covered by   -*
*- system abends.                                                   -*
*--------------------------------------------------------------------*
*        ST    R15,CURR_RC             SAVE RETURN CODE
*--------------------------------------------------------------------*
*- Return to caller                                                 -*
*--------------------------------------------------------------------*
Z500RET  L     R14,Z500SR14            RESTORE REGISTER 14
         BR    R14                     RETURN TO CALLER
         EJECT
**********************************************************************
*--------------------------------------------------------------------*
*- Z 6 0 0 _ L I N K _ C E D A  :  DPL to CEDA Interface            -*
*--------------------------------------------------------------------*
**********************************************************************
*- R2   DSECT CM411COM                                              -*
*- R3   BASE CSECT CM412                                            -*
*- R4   Work Register                                               -*
*- R5   Work Register                                               -*
*- R6   CEDA Command Length / Work Register                         -*
*- R7   Work Register                                               -*
*- R8   DSECT CM419COM                                              -*
*- R9   DSECT CM420COM                                              -*
*- R10  DSECT EXCI_RETURN_CODE                                      -*
*- R11  DSECT EXCI_DPL_RETAREA                                      -*
*- R12  Work Register                                               -*
*- R13  DYNREG DSECT DFHEISTG                                       -*
*- R14  Linkage                                                     -*
**********************************************************************
Z600_LINK_CEDA  DS  0H
```

```
*
         ST    R14,Z600SR14            SAVE REGISTER 14
*-------------------------------------------------------------------*
*- Set the output from DFHEDAP to be written to a storage area.    -*
*-------------------------------------------------------------------*
         STH   R6,CM411COM_COM_LNG     STORE COMMAND LENGTH
         MVI   CM411COM_FLAG,CM411COM_FLAG_STOR  OUTPUT TO STORAGE
         LA    R4,L'CM411COM_STORAGE   LENGTH OF OUTPUT STORAGE
         STH   R4,CM411COM_STG_LEN     STORE LENGTH OF OUTPUT STORAGE
         BAL   R14,Z200_WRITE_SYSPRINT BLANK LINE
*-------------------------------------------------------------------*
*- Issue LINKing Message.                                          -*
*-------------------------------------------------------------------*
         MVC   CM419COM_SYSPRINT_MSG,=C'CM412501I'     MOVE MSG. NO.
         MVC   CM419COM_SYSPRINT_DATA(L'CM412501I),CM412501I  & TEXT
         BAL   R14,Z200_WRITE_SYSPRINT  WRITE MSG TO SYSPRINT
*-------------------------------------------------------------------*
*- DPL_CALL                                                        -*
*-------------------------------------------------------------------*
         MVC   CM420COM_CALL_TYPE,=AL4(DPL_REQUEST)      CALL TYPE
         MVC   CM420COM_PGMNAME,=CL8'CM411'              PGMNAME
         MVC   CM420COM_COMMAREA_LEN,=AL4(CM411COM_LENGTH) LENGTH
         MVC   CM420COM_DATA_LEN,=AL4(CM411COM_LENGTH)     LENGTH#
         MVC   CM420COM_TRANSID,=C'M420'                 TRAN ID
         MVC   CM420COM_OPTIONS,=AL1(SYNCONRETURN)       OPTIONS
*
Z600CALL CALL  DFHXCIS,                                            X
               (CM420COM_VERSION_NUM,                              X
               CM420COM_RETURN_AREA,                               X
               CM420COM_USER_TOKEN,                                X
               CM420COM_CALL_TYPE,                                 X
               CM420COM_PIPE_TOKEN,                                X
               CM420COM_PGMNAME,                                   X
               CM411COM,                                           X
               CM420COM_COMMAREA_LEN,                              X
               CM420COM_DATA_LEN,                                  X
               CM420COM_TRANSID,                                   X
               0,                                                  X
               0,                                                  X
               CM420COM_DPL_RETAREA,                               X
               CM420COM_OPTIONS),                                  X
               VL,                                                 X
               MF=(E,CM420COM_PL)
*
         MVI   COMD_FLAG,FLAG_OFF      SET COMMAND FLAG OFF
*-------------------------------------------------------------------*
*- Check the EXCI response and produce diagnostics if required.    -*
*-------------------------------------------------------------------*
Z600CHEX CLC   EXCI_RESPONSE,=AL4(EXCI_NORMAL)  IF EXCI NORMAL
         BE    Z600CHDP                          THEN CHECK DPL
```

```
         BAL   R14,Z3ØØ_EXCI_DIAGNOSTICS          ELSE DIAGNOSTICS
         MVC   CURR_RC,EXCI_RESPONSE                   SET RC
*-------------------------------------------------------------------*
*- Check the DPL response and produce diagnostics if required.      -*
*-------------------------------------------------------------------*
Z6ØØCHDP CLC   EXCI_DPL_RESP,=AL4(EXEC_NORMAL)  IF DPL NORMAL
         BE    Z6ØØCHRC                          THEN CHECK CM411
         BAL   R14,Z7ØØ_DPL_DIAGNOSTICS          ELSE DIAGNOSTICS
         CLC   CURR_RC,EXCI_DPL_RESP            IF RESP < CURR_RC
         BH    Z6ØØCHRC                          THEN CHECK CM411
         MVC   CURR_RC,EXCI_DPL_RESP             ELSE RESET RC
*-------------------------------------------------------------------*
*- Check the CEDA Diagnostics return code and issue message with    -*
*- the return code.                                                 -*
*-------------------------------------------------------------------*
Z6ØØCHRC L     R4,CM411COM_DIAG_RC     LOAD DIAGNOSTICS CODE
         CVD   R4,WORKDW_1             CONVERT TO DECIMAL
         UNPK  WORKDW_2,WORKDW_1       CONVERT TO ...
         OI    WORKDW_2+7,X'FØ'        ... DISPLAYABLE DECIMAL
         MVC   CM419COM_SYSPRINT_MSG,=C'CM412503I'     MOVE MSG. NO.
         MVC   CM419COM_SYSPRINT_DATA(L'CM412503I),CM412503I    & TEXT
         MVC   CM419COM_SYSPRINT_DATA+26(L'WORKDW_2),WORKDW_2   & CODE
         BAL   R14,Z2ØØ_WRITE_SYSPRINT  WRITE MSG TO SYSPRINT
         CLC   CM411COM_DIAG_RC,=F'Ø'    IF DIAG RC = ZERO
         BE    Z6ØØCHR2                    THEN CHECK EXEC RC
         CLC   CURR_RC,CM411COM_DIAG_RC  IF DIAG RC < CURR_RC
         BH    Z6ØØCHR2                    THEN CHECK EXEC RC
         MVC   CURR_RC,CM411COM_DIAG_RC    ELSE RESET RC
*-------------------------------------------------------------------*
*- Check the CEDA Execution return code and issue message with      -*
*- the return code.                                                 -*
*-------------------------------------------------------------------*
Z6ØØCHR2 L     R4,CM411COM_EXEC_RC     LOAD EXECUTION CODE
         CVD   R4,WORKDW_1             CONVERT TO DECIMAL
         UNPK  WORKDW_2,WORKDW_1       CONVERT TO ...
         OI    WORKDW_2+7,X'FØ'        ... DISPLAYABLE DECIMAL
         MVC   CM419COM_SYSPRINT_MSG,=C'CM412504I'     MOVE MSG. NO.
         MVC   CM419COM_SYSPRINT_DATA(L'CM412504I),CM412504I    & TEXT
         MVC   CM419COM_SYSPRINT_DATA+26(L'WORKDW_2),WORKDW_2   & CODE
         BAL   R14,Z2ØØ_WRITE_SYSPRINT  WRITE MSG TO SYSPRINT
         CLC   CM411COM_EXEC_RC,=F'Ø'    IF EXEC RC = ZERO
         BE    Z6ØØCHSY                    THEN CHECK CM411 RC
         CLC   CURR_RC,CM411COM_EXEC_RC  IF EXEC RC < CURR_RC
         BH    Z6ØØCHSY                    THEN CHECK CM411 RC
         MVC   CURR_RC,CM411COM_EXEC_RC    ELSE RESET RC
*-------------------------------------------------------------------*
*- Check the CM411 return code.                                     -*
*-------------------------------------------------------------------*
Z6ØØCHSY CLC   CM411COM_CM411_RC,=F'Ø'   IF CM411 RC = ZERO
         BE    Z6ØØCHCU                    THEN CHECK CURRENT
```

```
         CLC   CURR_RC,CM411COM_CM411_RC IF CM411 RC < CURR_RC
         BH    Z600CHCU                    THEN CHECK CURRENT
         MVC   CURR_RC,CM411COM_CM411_RC   ELSE RESET RC
*-------------------------------------------------------------------*
*- Write message to SYSPRINT containing the highest return code     -*
*- encountered during the processing of CM411 (and DFHEDAP).        -*
*-------------------------------------------------------------------*
Z600CHCU L     R4,CURR_RC              LOAD CURRENT RC
         CVD   R4,WORKDW_1             CONVERT TO DECIMAL
         UNPK  WORKDW_2,WORKDW_1       CONVERT TO ...
         OI    WORKDW_2+7,X'F0'        ... DISPLAYABLE DECIMAL
         MVC   CM419COM_SYSPRINT_MSG,=C'CM412502I'     MOVE MSG. NO.
         MVC   CM419COM_SYSPRINT_DATA(L'CM412502I),CM412502I    & TEXT
         MVC   CM419COM_SYSPRINT_DATA+40(L'WORKDW_2),WORKDW_2   & CODE
         BAL   R14,Z200_WRITE_SYSPRINT  WRITE MSG TO SYSPRINT
         BAL   R14,Z200_WRITE_SYSPRINT  BLANK LINE
*-------------------------------------------------------------------*
*- If a return code greater than 4 has been encountered, write a    -*
*- message to SYSPRINT stating that all subsequent commands will    -*
*- be ignored.                                                      -*
*-------------------------------------------------------------------*
Z600CHST CLC   CURR_RC,=F'4'             IF RC > 4
         BH    Z600IGNO                  THEN "IGNORING" MSG
         B     Z600CLBU                  ELSE CLEAR COMMAND BUFFER
Z600IGNO MVC   CM419COM_SYSPRINT_MSG,=C'CM412505I'     MOVE MSG. NO.
         MVC   CM419COM_SYSPRINT_DATA(L'CM412505I),CM412505I    & TEXT
         BAL   R14,Z200_WRITE_SYSPRINT  WRITE MSG TO SYSPRINT
         BAL   R14,Z200_WRITE_SYSPRINT  BLANK LINE
*-------------------------------------------------------------------*
*- Clear command buffer for next command                           -*
*-------------------------------------------------------------------*
Z600CLBU LA    R4,CM411COM_COMMAND     ADDRESS COMMAND BUFFER
         LA    R5,L'CM411COM_COMMAND   LENGTH OF COMMAND BUFFER
         XR    R6,R6                   FROM ADDRESS NOT REQUIRED
         XR    R7,R7                   SET LENGTH TO 0
         ICM   R7,8,=C' '              SET PADDING TO BLANKS
         MVCL  R4,R6                   CLEAR STORAGE TO BLANKS
*-------------------------------------------------------------------*
*- Return to caller                                                -*
*-------------------------------------------------------------------*
Z600RET  BAL   R14,Z400_CHECK_RC       CHECK HIGHEST RC
         L     R14,Z600SR14            RESTORE REGISTER 14
         BR    R14                     RETURN TO CALLER
         EJECT
*********************************************************************
*-------------------------------------------------------------------*
*- Z 7 0 0 _ D P L _ D I A G N O S T I C S  :  Diagnostics          -*
*-------------------------------------------------------------------*
*********************************************************************
*- R3   BASE CSECT CM412                                           -*
```

```
*- R4   Work Register                                                 -*
*- R8   DSECT CM419COM                                                -*
*- R11  DSECT EXCI_DPL_RETAREA                                        -*
*- R13  DYNREG DSECT DFHEISTG                                         -*
*- R14  Linkage                                                       -*
************************************************************************
Z700_DPL_DIAGNOSTICS  DS  0H
*
         ST    R14,Z700SR14            SAVE REGISTER 14
*---------------------------------------------------------------------*
*- Write DPL diagnostics follow message.                              -*
*---------------------------------------------------------------------*
         MVC   CM419COM_SYSPRINT_MSG,=C'CM412020E'     MOVE MSG. NO.
         MVC   CM419COM_SYSPRINT_DATA(L'CM412020E),CM412020E  & TEXT
         BAL   R14,Z200_WRITE_SYSPRINT  WRITE MSG TO SYSPRINT
*---------------------------------------------------------------------*
*- Convert EXCI_DPL_RESP to displayable decimal characters            -*
*- and issue message.                                                 -*
*---------------------------------------------------------------------*
         L     R4,EXCI_DPL_RESP        LOAD RESPONSE
         CVD   R4,WORKDW_1             CONVERT TO DECIMAL
         UNPK  WORKDW_2,WORKDW_1       CONVERT TO ...
         OI    WORKDW_2+7,X'F0'        ... DISPLAYABLE DECIMAL
         MVC   CM419COM_SYSPRINT_MSG,=C'CM412020E'     MOVE MSG. NO.
         MVC   CM419COM_SYSPRINT_DATA+6(L'CM412020A),CM412020A  & TEXT
         MVC   CM419COM_SYSPRINT_DATA+28(L'WORKDW_2),WORKDW_2   & CODE
         BAL   R14,Z200_WRITE_SYSPRINT  WRITE MSG TO SYSPRINT
*---------------------------------------------------------------------*
*- Convert EXCI_DPL_RESP2 to displayable decimal characters           -*
*- and issue message.                                                 -*
*---------------------------------------------------------------------*
         L     R4,EXCI_DPL_RESP2       LOAD REASON 2
         CVD   R4,WORKDW_1             CONVERT TO DECIMAL
         UNPK  WORKDW_2,WORKDW_1       CONVERT TO ...
         OI    WORKDW_2+7,X'F0'        ... DISPLAYABLE DECIMAL
         MVC   CM419COM_SYSPRINT_MSG,=C'CM412020E'     MOVE MSG. NO.
         MVC   CM419COM_SYSPRINT_DATA+6(L'CM412020B),CM412020B  & TEXT
         MVC   CM419COM_SYSPRINT_DATA+28(L'WORKDW_2),WORKDW_2   & CODE
         BAL   R14,Z200_WRITE_SYSPRINT  WRITE MSG TO SYSPRINT
*---------------------------------------------------------------------*
*- Issue Abend Code Message.                                          -*
*---------------------------------------------------------------------*
         MVC   CM419COM_SYSPRINT_MSG,=C'CM412020E'     MOVE MSG. NO.
         MVC   CM419COM_SYSPRINT_DATA+6(L'CM412020C),CM412020C  & TEXT
         MVC   CM419COM_SYSPRINT_DATA+28(L'EXCI_DPL_ABCODE),EXCI_DPL_ABX
               CODE
         BAL   R14,Z200_WRITE_SYSPRINT  WRITE MSG TO SYSPRINT
*---------------------------------------------------------------------*
*- Return to caller                                                   -*
*---------------------------------------------------------------------*
```

```
Z700RET  L     R14,Z700SR14            RESTORE REGISTER 14
         BR    R14                     RETURN TO CALLER
         EJECT
*********************************************************************
*-----------------------------------------------------------------*
*- M O D I F I A B L E   I N S T R U C T I O N S                 -*
*-----------------------------------------------------------------*
*********************************************************************
* None !!
         EJECT
*********************************************************************
*-----------------------------------------------------------------*
*- C O N S T A N T S                                             -*
*-----------------------------------------------------------------*
*********************************************************************
BLANKS   DC    CL(L'CM419COM_SYSIN_DATA)' '   BLANK LINE
COMMENT  DC    C'*'                           COMMENT INDICATOR
*
         EJECT
*********************************************************************
*-----------------------------------------------------------------*
*- T A B L E S                                                   -*
*-----------------------------------------------------------------*
*********************************************************************
TRANTAB0 DC    C'0123456789ABCDEF'     TRANSLATE HALF BYTES
         EJECT
*********************************************************************
*-----------------------------------------------------------------*
*- M E S S A G E S                                               -*
*-----------------------------------------------------------------*
*********************************************************************
CM412001I  DC  C'Loading program          ....'
CM412002I  DC  C'Program          loaded at address X''12345678''.'
CM412003I  DC  C'         storage address X''        '' length         X
                bytes.'
CM412010E  DC  C'Error processing EXCI call, diagnostics follow:-'
CM412010A  DC  C'EXCI Response . . . : '
CM412010B  DC  C'EXCI Reason . . . . : '
CM412010C  DC  C'EXCI Subreason-1  . : '
CM412010D  DC  C'EXCI Subreason-2  . : '
*
CM412020E  DC  C'Error processing DPL call, diagnostics follow:-'
CM412020A  DC  C'EXEC Response 1 . . : '
CM412020B  DC  C'EXEC Response 2 . . : '
CM412020C  DC  C'EXEC Abend Code . . : '
*
CM412501I  DC  C'DPL_CALL to CM411 processing....'
CM412502I  DC  C'DPL_CALL to CM411 complete. Return Code '
CM412503I  DC  C'CEDA Diagnostics Code . : '
CM412504I  DC  C'CEDA Execution Code . . : '
```

```
CM412505I  DC  C'Input ignored due to return code:-'
CM412506E  DC  C'Invalid command. Subsequent input will be ignored:-'
*
CM412901I  DC  C'Program Terminated. Highest Return Code '
CM412902I  DC  C'Number of commands processed '
         EJECT
*********************************************************************
*-------------------------------------------------------------------*
*- E N D   CM412                                                   -*
*-------------------------------------------------------------------*
*********************************************************************
         END   CM412
```

CM411

The EXCI client interface to DFHEDAP (CM412) cannot link directly to DFHEDAP. DFHEDAP must be linked to, passing a COMMAREA containing five 31-bit addresses, each contained in a fullword. That was the problem; DFHEDAP expected not data values to be passed, but addresses where data values could be found. When a DPL request passing a COMMAREA is processed, the COMMAREA is effectively copied into a storage area in the target CICS region and the server program is passed the address of that storage area. There was obviously no way of knowing the address where the storage would be copied and therefore the addresses required by DFHEDAP.

CM411 provides the solution to this problem by receiving the CM411COM COMMAREA and then setting the five 31-bit addresses within the CICS server region. It links to DFHEDAP, passing the addresses required by DFHEDAP in a COMMAREA.

DFHEDAP does provide detailed CEDA diagnostic (command validation) and execution messages. However, the format of the message fields is not guaranteed and could be changed by a PTF or in a new release or version of CICS. To avoid maintenance overhead, CM411 does not attempt to process the messages from DFHEDAP. CM411 processes only the return codes from DFHEDAP. If a return code other than zero is returned, CM411 writes a quick transaction dump, in which the messages could easily be read. This may appear somewhat

lazy, but I wanted CM411 to be as maintenance independent as possible.

It is important to note at this point that CM411 is a 'normal' CICS program and could be used (linked to) by any other CICS program wishing to process CEDA commands using the CM411COM COMMAREA. It is not specific to CM412 or EXCI.

CM411 should be assembled and link-edited using the same procedure you would use for any CICS program written in Assembler.

CM411 uses the subprograms CM401, CM402, and CM403, which can be obtained from the Xephon Web site.

```
*ASM XOPTS(CICS)
CM411    TITLE 'CM411 : CEDA INTERFACE'
*********************************************************************
*                 C A R L   W A D E   M C B U R N I E               *
*                   -  I T   C O N S U L T A N T  -                 *
*                                                                   *
*                            www.cwmit.com                          *
*********************************************************************
*                                                                   *
* MODULE NAME = CM411                                               *
* MODULE TYPE = CSECT  (Sub-Program)                                *
* DESCRIPTION = CICS/ESA CEDA Interface                             *
*                                                                   *
*               EXCI Server Program for the Programmable            *
*               Interface to CEDA (RDO).                            *
*                                                                   *
*               The program receives a communications area          *
*               CM411COM from the Batch Client Program CM412,       *
*               sets the required pointers, LINKs to DFHEDAP,       *
*               and returns the communications area to CM412,       *
*               including return codes and messages from            *
*               DFHEDAP.                                            *
*********************************************************************
         EJECT
*********************************************************************
*                                                                   *
* CHANGE HISTORY:                                                   *
* --------------                                                    *
*********************************************************************
         EJECT
*********************************************************************
* REGISTER  EQUATES                      USAGE                      *
* --------  --------  ------------------------------------------- *
```

```
*                                                                     *
* REG  Ø    RØ                                                       *
* REG  1    R1                                                       *
* REG  2    R2        DSECT - CM411COM                               *
* REG  3    BASE      Base Register for CSECT CM411                  *
* REG  4    R4        Work Register                                  *
* REG  5    R5        Work Register                                  *
* REG  6    R6        Work Register                                  *
* REG  7    R7        Work Register                                  *
* REG  8    R8        DSECT - CM4Ø1COM                               *
* REG  9    R9        DSECT - CM4Ø2COM                               *
* REG 1Ø    R1Ø       DSECT - CM4Ø3COM                               *
* REG 11    EIBREG    DSECT - DFHEIBLK                               *
* REG 12    R12       DSECT - CM411COM_STG_HEADER                    *
* REG 13    DYNREG    DSECT - DFHEISTG                               *
* REG 14    R14       Linkage                                        *
* REG 15    R15       Linkage                                        *
***********************************************************************
         EJECT
*---------------------------------------------------------------------*
*- Copybooks                                                         -*
*---------------------------------------------------------------------*
         EJECT
         COPY  CM4Ø1AØ1               DSECT - CM4Ø1COM  (TDQ MSG)
         EJECT
         COPY  CM4Ø2AØ1               DSECT - CM4Ø2COM  (EXEC DIAGS)
         EJECT
         COPY  CM4Ø3AØ1               DSECT - CM4Ø3COM  (WTO MSG)
         EJECT
         COPY  CM411AØ1               DSECT - CM411COM  (CEDA INTER.)
         EJECT
*---------------------------------------------------------------------*
*- Addressability to DFHEISTG will be established by CICS.          -*
*---------------------------------------------------------------------*
         DFHEISTG                     CICS DYNAMIC STORAGE
         EJECT
*---------------------------------------------------------------------*
*- CM411 Dynamic Storage - Start                                     -*
*---------------------------------------------------------------------*
DYNSTOR  DS    ØH                     CM411 DYNAMIC STORAGE
*
TRAN_ID  DS    CL4                    TRANSACTION IDENTIFIER
TERM_ID  DS    CL4                    TERMINAL IDENTIFIER
USER_ID  DS    CL8                    USER IDENTIFIER
*
ABNDCODE DS    CL4                    ABEND CODE
*
DUMP_REQD   DS  X                     DUMP REQUIRED FLAG
*
Z1ØØ_RESP1  DS  F                     RESP1 CODE
```

```
Z100_RESP2  DS  F                       RESP2 CODE
Z100_FUNC   DS  CL2                     FUNCTION CODE
*
A100SR14 DS    F                        SAVE REGISTER 14
A500SR14 DS    F                        SAVE REGISTER 14
A900SR14 DS    F                        SAVE REGISTER 14
Z100SR14 DS    F                        SAVE REGISTER 14
Z200SR14 DS    F                        SAVE REGISTER 14
Z300SR14 DS    F                        SAVE REGISTER 14
*
         DS    0D                       ALIGN STORAGE
CM401ST  DS    CL(CM401COM_LENGTH)      STORAGE FOR CM401COM
*
         DS    0D                       ALIGN STORAGE
CM402ST  DS    CL(CM402COM_LENGTH)      STORAGE FOR CM402COM
*
         DS    0D                       ALIGN STORAGE
CM403ST  DS    CL(CM403COM_LENGTH)      STORAGE FOR CM403COM
*
DYNSTORL EQU   *-DYNSTOR                LENGTH OF DYNAMIC STORAGE
*-------------------------------------------------------------------*
*- CM411 Dynamic Storage - End                                      -*
*-------------------------------------------------------------------*
         EJECT
*-------------------------------------------------------------------*
*- Register Equates                                                 -*
*-------------------------------------------------------------------*
         DFHREGS                        CICS STANDARD EQUATES
BASE     EQU   3                        BASE CODE REGISTER
EIBREG   EQU   11                       EXEC INTERFACE BLOCK REGISTER
DYNREG   EQU   13                       DYNAMIC STORAGE REGISTER
         EJECT
*********************************************************************
*===================================================================*
*=       E  N  T  R  Y          P  O  I  N  T                      =*
*===================================================================*
*********************************************************************
CM411    DFHEIENT CODEREG=(BASE),DATAREG=(DYNREG),EIBREG=(EIBREG)
CM411    AMODE 31
CM411    RMODE ANY
*-------------------------------------------------------------------*
*- Program Identification "Eye-Catchers"                            -*
*-------------------------------------------------------------------*
         B     A000_MAINLINE            BRANCH OVER EYE-CATCHERS
ASMEYE   DC    C'*'                     ASTERISK
ASMPROG  DC    C'CM411   '              PROGRAM NAME
         DC    C'-'                     HYPHEN
ASMLVL   DC    C'CWM00001'              PROGRAM LEVEL
         DC    C' '                     BLANK
ASMDATE  DC    C'&SYSDATE'              DATE OF ASSEMBLY
```

```
         DC    C' '                    BLANK
ASMTIME  DC    C'&SYSTIME'             TIME OF ASSEMBLY
         DC    C'*'                    ASTERISK
ASMEYEL  EQU   *-ASMEYE                LENGTH OF EYE-CATCHER
*
         EJECT
*********************************************************************
*-------------------------------------------------------------------*
*- A 0 0 0 _ M A I N L I N E :  Controls the flow of the program    -*
*-------------------------------------------------------------------*
*********************************************************************
*- R14  Linkage                                                     -*
*********************************************************************
A000_MAINLINE  DS  0H
         BAL   R14,A100_INITIALIZE     PERFORM INITIALIZATION
         BAL   R14,A500_LINK_DFHEDAP   PERFORM CEDA PROCESSING
         BAL   R14,A900_TERMINATION    PERFORM TERMINATION
*********************************************************************
*-------------------------------------------------------------------*
*- A 0 0 0 _ R E T U R N _ T O _ C I C S                           -*
*-------------------------------------------------------------------*
*********************************************************************
A000_RETURN_TO_CICS  DS  0H
*
RETURN   EXEC  CICS RETURN
*********************************************************************
*===================================================================*
*=       E  X  I  T            P  O  I  N  T                      =*
*===================================================================*
*********************************************************************
         EJECT
```

Editor's note: this article will be concluded next month.

Carl Wade McBurnie
IT Consultant (Germany)

.NET for CICS users

It all started when Micro Focus and Microsoft announced an alliance to enable the migration of critical proprietary mainframe systems onto Windows using Microsoft's .NET technology. The Micro Focus Enterprise Server with its Mainframe Transaction Option was designed to enable the migration and

deployment of CICS/COBOL mainframe applications onto the Windows platform. Why would anyone want to do that? The companies claimed that once an application had been re-hosted, it could be extended through the use of the .NET Framework, SQL Server 2000, XML, and Web services. The company also claimed that it helped customers reduce the cost of maintaining and modernizing their mainframe environments, suggesting that cheaper platforms saved time and money.

Shortly after that, Micro Focus announced Mainframe Express Enterprise Edition, a Windows-based environment for mainframe application development. The product combined a mainframe emulation and development environment with application analysis and automated program and component generation. Again, there was a promise of existing CICS legacy services being extended to Web services, .NET, or J2EE.

Fujitsu Software came out with Version 1.0 of NeoKicks. This provided a way for CICS applications to move to Microsoft .NET with ASP.NET pages, Windows Forms, and Visual Studio .NET using wizards. Fujitsu claimed that it lowered platform maintenance costs, gave interfaces new life as ASP.NET Web applications or Windows Forms client applications, and integrated with Visual Studio .NET for higher developer productivity.

So if we are thinking (even remotely) that we might do some CICS development in a .NET environment, what are we talking about?

It seems that .NET is a set of standards that define how software talks to other software, including Windows. The programming languages to use are Visual Basic .NET, C#, and J#. Visual Basic .NET is only slightly similar to its predecessor Visual Basic. It's different in that it's an object-oriented language and harder to learn. C# is similar to VB .NET. It has elements of C++ and Java in it. It's probably easier

to learn than VB .NET. J# is designed for Java programmers who want to convert existing Java applications to run under .NET.

Key to .NET is a component called the Common Language Run-time (CLR). It provides the same service to all the .NET programming languages in that it is a single system to execute all .NET applications. An application that is run under the supervision of the CLR is called managed code.

The .NET programs – ie written in VB .NET, C#, or J# – are taken by the .NET compiler and converted to what's called Microsoft Intermediate Language (MSIL) Assembler. This MSIL is converted into machine code at the last minute by the CLR before execution. This is Just-In-Time (JIT) compiling. It is used to allow the CLR to check that a program won't do anything dangerous or illegal. The performance level is still OK. This is meant to ensure that .NET applications are reliable and secure.

There is also what's called the .NET Framework, which is a large software library that allows programs to access the Windows operating system. This is still under development.

If you've heard of COM (Common Object Model) and ActiveX and, Automation Interfaces, forget them – they are all being replaced by .NET classes. If you thought you'd need to build a COM object library, and possibly add an ActiveX control or an Automation Interface, you don't. You write your program (using a .NET) language, and make your classes available as extensions to the .NET Framework. Although, if you migrate today, you will still get tied up with this older technology.

If you're thinking that the .NET strategy sounds similar to Java, you're not wrong. ActiveX components on Web pages are like Java applets, and desktop applications do the same job as Java beans. C# is a programming language similar to Java, and the CLR is similar to the Java Virtual Machine (JVM). The .NET Framework looks like Java class libraries. So how are they different? Well, .NET runs only on Windows

and Java runs on everything else (almost – it's platform independent).

Microsoft is quite keen that developing for the Web should be the same as developing for the desktop. This involves using Windows Web servers rather than Apache (or whatever). Their plan for server domination involves the use of ASP .NET. This is still some way in the future and requires a Microsoft SQL database as well. ASP .NET compiles Web pages into HTML – which everyone can read.

Microsoft is quite keen on Web services, and this is important for CICS sites because it allows a program on one computer to call an application running on another (ie an established CICS application). .NET Web services are quite easy to create. An application on a desktop computer calls a program over the Internet and the results appear back on the desktop machine. Web protocols that allow Web services include SOAP, WDSL, and XML. SOAP (Simple Object Access Protocol) is an XML-based protocol to implement Web services. WDSL (Web Services Description Language) is an XML format for describing Web services. XML (eXtensible Mark-up Language) is a mark-up language format to describe data. Visual Studio .NET can be used to create Web applications.

Nick Nourse
Independent Consultant (UK)

Why not share your expertise and earn money at the same time? Articles can be of any length and should be e-mailed to the editor, Trevor Eddolls, at trevore@xephon.com.

A free copy of our *Notes for Contributors*, which includes information about payment rates, is available from our Web site at www.xephon.com/nfc.

Compuware has announced new releases in its XPEDITER product line, including Program Analyzer, a new analysis tool to help developers understand enterprise applications.

Compuware also has new versions of XPEDITER/CICS and XPEDITER/Code Coverage, which are meant to enable organizations to deploy enterprise applications as they evolve toward Service-Oriented Architectures (SOA).

XPEDITER/CICS delivers intercept capabilities that target any function initiated in CICS, regardless of where it is initiated. This simplifies CICS test set-up and execution for middleware and/or SOA-enabled environments, including WebSphere MQ.

For further information contact:
Compuware, 1 Campus Martius, Detroit, MI 48226, USA.
Tel: (313) 227 7300.
URL: www.compuware.com/products/xpediter/1996_ENG_HTML.htm.

***

NEON Systems and Defywire have announced an alliance to co-market and co-sell their Shadow technology with the Defywire Mobility Suite. The joint solution will allow customers to extend mainframe data and applications to a variety of hand-held mobile devices.

NEON's Shadow product offers the ability to create mainframe-based Web services and capture real-time mainframe events. Shadow z/Services enables J2EE or .NET developers to use Web services to service-enable mainframe CICS, IMS, and Advantage CA-IDMS applications. Shadow z/Events allows developers to utilize real-time mainframe-based business events, occurring within DB2, Adabas, IMS/DB, VSAM, and CICS/TS environments, with distributed applications such as Defywire Mobility Suite.

Defywire's wireless middleware connects mobile workers with front and back office systems in real-time. With the Defywire Mobility Suite, application developers can build one design and then deliver it to mobile phones, PDAs, tablets, and laptops.

For further information contact:
NEON, 14100 Southwest Freeway, Suite 500, Sugar Land, TX 77478, USA.
Tel: (281) 491 4200.
URL: www.neonsys.com/newsroom/press_releases/2004/041207.asp.

***

IBM has announced Version 3.1 of CICS Transaction Server for z/OS, which provides an enhanced Web Services solution. IBM also announced Version 6 of CICS Transaction Gateway, which provides J2EE standards-based connectivity.

Traditional workloads are now able to participate in Service-Oriented Architectures (SOA) through CICS Transaction Server V3. This helps customers address specific business problems, such as extending current CICS applications, says IBM.

For further information contact your local IBM representative.
URL: www-306.ibm.com/software/htp/cics/tserver/v31/.