



**Legac-E Education**

# **Exploiting the CICS/Batch Interface (EXCI)**



Legac-E Education

## Exploiting the CICS/Batch Interface (EXCI)

Topics	Page
<i>Introduction</i> _____	<i>1</i>
<i>Automating CEMT NEWCOPY</i> _____	<i>2</i>
Batch Client program – EXCLN03 _____	2
CICS Client program – EXSRV03 _____	4
<i>Automate CEDA Install</i> _____	<i>8</i>
Batch Client Program - EXCCLN01 _____	8
CICS Client Program – EXSRV01 _____	10
<i>Automate CEMT P SHUT from Batch</i> _____	<i>12</i>
Batch Server Program – EXCLN02 _____	12
CICS Server program – EXSRV02 _____	14
Actual SHUTDOWN program _____	16



## **Exploiting the CICS/Batch Interface (EXCI)**

### **Introduction**

When running first the FLEX-ES System (2003); followed by the zPDT Version from 2010 onwards there was a requirement to automate various tasks to ease the administrative burden.

Without access to Independent Software Vendor (ISV) Products, the solution was to code our own principally using Assembler but with some REXX routines included.

The CICS Batch Interface (EXCI) provided scope for assisting with some CICS related tasks so the following three were engineered using COBOL as the primary coding language:

- Automate CEMT NEWCOPY after compiling amended versions of programs or maps.
- Automate CEDA INSTALL after running DFHCSDUP to add new resources
- Provide a batch mechanism for performing clean SHUTDOWN of CICS, i.e. CEMT P SHUT.

It should be noted that the CICS Batch Interface utilises a Client/Server relationship with CICS as the Server, therefore exploitation typically requires two programs, one in Batch and the other in CICS.

The examples included here were introduced into a CICS/TS 4.1 system and were still active when operations ceased at CICS/TS 5.1.



## Exploiting the CICS/Batch Interface (EXCI)

### Automating CEMT NEWCOPY

#### Batch Client program – EXCLN03

```
IDENTIFICATION DIVISION.
PROGRAM-ID.                                EXCCLN03.
ENVIRONMENT DIVISION.
INPUT-OUTPUT SECTION.
DATA DIVISION.
FILE SECTION.
WORKING-STORAGE SECTION.
*-----*
*   Declare Call level,DPL, and EXEC level Return Code areas. *
*-----*
COPY DFHXCPLO.
*-----*
*   Initialise Target information variables. *
*-----*
01  TARGET-PROGRAM          PIC X(8)  VALUE 'EXCSRV03'.
01  TARGET-TRANSID         PIC X(4)  VALUE 'EXCI'.
01  TARGET-SYSTEM          PIC X(8)  VALUE SPACES.
*-----*
*   Initialise Call level specific variables. *
*-----*
01  LINK-COM-LEN           PIC S9(4) COMP VALUE 80.
01  LINK-DAT-LEN           PIC S9(4) COMP VALUE 80.
01  USER-TOKEN             PIC S9(8) COMP VALUE ZERO.
01  PIPE-TOKEN             PIC S9(8) COMP VALUE ZERO.
*-----*
*   Initialise Commarea length and Data length(in bytes). *
*-----*
01  CEMT-COMMAREA.
    05  CEMT-NEW-PGM        PIC X(8)  VALUE SPACES.
    05                      PIC X(72) VALUE SPACES.
01  MSG1.
    05                      PIC X(26) VALUE
        'EXCCLN03- About to issue '.
    05                      PIC X(17) VALUE
        'CEMT SET PROGRAM('.
    05  MSG-PGM1           PIC X(8)  VALUE SPACES.
    05                      PIC X(10) VALUE ') NEWCOPY '.
01  MSG-OK                 PIC X(49) VALUE
        'EXCCLN03 - CICS indicates NEWCOPY was successful.'.
01  MSG-NO-PPT             PIC X(57) VALUE
        'EXCCLN03 - CICS used PHASEIN instead of NEWCOPY'.
01  MSG-NO-MODULE          PIC X(48) VALUE
        'EXCCLN03 - CICS indicates module not in library.'.
LINKAGE SECTION.
01  PARM-DATA.
    03  PARM-LENGTH        PIC 9(4)  COMP.
    03  REAL-PARM-DATA.
        05  PARM-REGION    PIC X(8)  .
```



## Exploiting the CICS/Batch Interface (EXCI)

```
      05                                PIC X.
      05 LS-NEW-PGM                      PIC X(8).
PROCEDURE DIVISION                      USING PARM-DATA.
*-----*
*   Extract the PARM information from the JCL EXEC statement.*
*-----*
      IF PARM-LENGTH = 0                 DISPLAY 'EXCCLN03 No Parm'
                                         UPON SYSOUT
                                         GOBACK.
      UNSTRING REAL-PARM-DATA            DELIMITED BY ','
                                         INTO TARGET-SYSTEM
                                         CEMT-NEW-PGM
                                         END-UNSTRING.
      MOVE PARM-REGION                   TO TARGET-SYSTEM.
      MOVE CEMT-NEW-PGM                  TO MSG-PGM1.
*-----*
*   Connect to the relevant CICS Region in a single pass.*
*-----*
      DISPLAY MSG1                       UPON SYSOUT.
      EXEC CICS LINK                      PROGRAM(TARGET-PROGRAM)
                                         TRANSID(TARGET-TRANSID)
                                         APPLID(TARGET-SYSTEM)
                                         COMMAREA(CEMT-COMMAREA)
                                         LENGTH(LINK-COM-LEN)
                                         DATALENGTH(LINK-DAT-LEN)
                                         RETCODE(EXCI-EXEC-RETURN-CODE)
                                         SYNCONRETURN
      END-EXEC.
      EVALUATE CEMT-NEW-PGM
      WHEN 'NOTFND '                     DISPLAY MSG-NO-PPT
                                         UPON SYSOUT
      WHEN 'PHASEIN '                   DISPLAY MSG-NO-MODULE
                                         UPON SYSOUT
      WHEN OTHER                         DISPLAY MSG-OK
                                         UPON SYSOUT.
*-----*
*   Logical end of program, return to z/OS
*-----*
      MOVE ZERO                          TO RETURN-CODE.
      GOBACK.
*-----*
*   Physical end of source for program - EXCCLN03.
*-----*
```



## Exploiting the CICS/Batch Interface (EXCI)

### CICS Client program – EXSRV03

```
CBL CICS('SP')
IDENTIFICATION DIVISION.
PROGRAM-ID.                EXCSR03.
ENVIRONMENT DIVISION.
DATA DIVISION.
WORKING-STORAGE SECTION.
*-----*
*  Server program to execute CEMT NEW PGM issued from a batch*
*  job via EXCI.                                           *
*-----*
01  DUMP-MARKER                PIC X(43) VALUE
    'EXSRV03 - WORKING STORAGE STARTS HERE'.
*-----*
*  Common area                                           *
*-----*
01  WS-COMMAREA.
    05  COMMAND-ADDR          USAGE IS POINTER.
    05  COM-LEN-ADDR          USAGE IS POINTER.
    05  FLAG-ADDR             USAGE IS POINTER.
    05  OUTPUT-ADDR           USAGE IS POINTER.
    05  OUTP-LEN-ADDR         USAGE IS POINTER.
01  REQUIRED-FIELDS.
    05  PGM-SIZE               PIC S9(8) COMP VALUE +0.
    05  LINK-COM-LEN           PIC S9(4) COMP VALUE +8.
    05  LINK-DAT-LEN          PIC S9(4) COMP VALUE +8.
    05  MSG1-L                 PIC S9(4) COMP VALUE +61.
    05  MSG2-L                 PIC S9(4) COMP VALUE +37.
    05  MSG3-L                 PIC S9(4) COMP VALUE +57.
    05  MSG4-L                 PIC S9(4) COMP VALUE +47.
    05  FLAG-AREA              PIC S9(4) COMP VALUE +0.
    05  REAL-FLAGS             REDEFINES FLAG-AREA.
        10                      PIC X.
        10  RESP-FLAG-D        PIC X.
    05  CEMT-COMMAND           PIC X(80) VALUE SPACES.
    05  NO-LOAD.
        10                      PIC X(7) VALUE 'NOTFND '.
        10                      PIC X(73) VALUE SPACES.
    05  USED-PHASEIN.
        10                      PIC X(7) VALUE 'PHASEIN'.
        10                      PIC X(73) VALUE SPACES.
01  MSG1.
    05                      PIC X(26) VALUE
        'EXCSR03 - About to issue '.
    05                      PIC X(17) VALUE
        'CEMT SET PROGRAM('.
    05  MSG-PGM1              PIC X(8) VALUE SPACES.
    05                      PIC X(10) VALUE ') NEWCOPY '.
01  MSG2.
    05                      PIC X(19) VALUE
```



## Exploiting the CICS/Batch Interface (EXCI)

```
      'EXCSRVO3 - Program '.
05  MSG-PGM2                PIC X(8)  VALUE SPACES.
05                          PIC X(10) VALUE ' not found'.
01  MSG3.
05                          PIC X(22) VALUE
      'EXCSRVO3 - Now trying '.
05                          PIC X(17) VALUE
      'CEMT SET PROGRAM(' .
05  MSG-PGM3                PIC X(8)  VALUE SPACES.
05                          PIC X(10) VALUE ') PHASEIN '.
01  MSG4.
05                          PIC X(19) VALUE
      'EXCSRVO3 - Program '.
05  MSG-PGM4                PIC X(8)  VALUE SPACES.
05                          PIC X(20) VALUE
      ' not in load library'.
01  OUTPUT-LENGTH          PIC S9(4) COMP VALUE 160.
01  CEMT-NEW-PGM           PIC X(8)  VALUE SPACES.
01  OUTPUT-AREA            PIC X(160) VALUE SPACES.
01  RESP-FLAG              PIC X.
      COPY DFHAID.
LINKAGE SECTION.
01  DFHCOMMAREA.
      05  LS-NEW-COPY        PIC X(8).
      05                          PIC X(72).
*-----*
*   Program Logic starts here.                                     *
*-----*
PROCEDURE DIVISION.
MAIN.
      MOVE LS-NEW-COPY                TO CEMT-NEW-PGM, MSG-PGM1,
                                      MSG-PGM2, MSG-PGM3,
                                      MSG-PGM4.
      EXEC CICS WRITEQ TD              QUEUE('CSML')
                                      FROM(MSG1)
                                      LENGTH(MSG1-L)
                                      END-EXEC.
      MOVE RESP-FLAG-D                TO RESP-FLAG.
      SET COMMAND-ADDR                TO ADDRESS OF CEMT-COMMAND.
      SET COM-LEN-ADDR                TO ADDRESS OF LINK-COM-LEN.
      SET FLAG-ADDR                   TO ADDRESS OF RESP-FLAG.
      SET OUTPUT-ADDR                 TO ADDRESS OF OUTPUT-AREA.
      SET OUTP-LEN-ADDR               TO ADDRESS OF OUTPUT-LENGTH.
      EXEC CICS HANDLE                 CONDITION
                                      PGMIDERR(DO-PHASEIN)
                                      END-EXEC.
      EXEC CICS HANDLE                 CONDITION
                                      PGMIDERR(DO-PHASEIN)
                                      IOERR(NO-LOAD-MODULE)
                                      END-EXEC.
```



## Exploiting the CICS/Batch Interface (EXCI)

```
EXEC CICS SET                                PROGRAM(CEMT-NEW-PGM)
                                              NEWCOPY
                                              END-EXEC.

*-----*
*   Logical end of the program, so return control to CICS   *
*-----*

EXEC CICS RETURN                                END-EXEC.
MAIN-EXIT.
EXIT.
DO-PHASEIN.

*-----*
*   This section is entered if the initial NEWCOPY attempt *
*   fails to locate designated program to be refreshed.    *
*-----*

EXEC CICS WRITEQ TD                            QUEUE('CSML')
                                              FROM(MSG2)
                                              LENGTH(MSG2-L)
                                              END-EXEC.

EXEC CICS WRITEQ TD                            QUEUE('CSML')
                                              FROM(MSG3)
                                              LENGTH(MSG3-L)
                                              END-EXEC.

EXEC CICS IGNORE                              CONDITION
                                              PGMIDERR
                                              END-EXEC.

EXEC CICS LOAD                                PROGRAM(CEMT-NEW-PGM)
                                              FLENGTH(PGM-SIZE)
                                              END-EXEC.

EXEC CICS SET                                PROGRAM(CEMT-NEW-PGM)
                                              PHASEIN
                                              END-EXEC.

MOVE USED-PHASEIN                            TO DFHCOMMAREA.
EXEC CICS RETURN                                END-EXEC.
DO-PHASEIN-EXIT.
EXIT.
NO-LOAD-MODULE.

*-----*
*   This section is entered if the second attempt fails    *
*   IOERR which typically means the load module ain't in the *
*   library.                                               *
*-----*

EXEC CICS WRITEQ TD                            QUEUE('CSML')
                                              FROM(MSG4)
                                              LENGTH(MSG4-L)
                                              END-EXEC.

MOVE USED-PHASEIN                            TO DFHCOMMAREA.
EXEC CICS RETURN                                END-EXEC.
NO-LOAD-MODULE-EXIT.
EXIT.

*-----*
*   The physical end of program EXCSR03.                  *
*-----*
```





**Legac-E Education**

## **Exploiting the CICS/Batch Interface (EXCI)**

*This page reserved.*



## Exploiting the CICS/Batch Interface (EXCI)

### Automate CEDA Install

#### Batch Client Program - EXCCLN01

```
IDENTIFICATION DIVISION.
PROGRAM-ID.                                EXCCLN01.
ENVIRONMENT DIVISION.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
DATA DIVISION.
FILE SECTION.
WORKING-STORAGE SECTION.
*-----*
*   Declare Call level,DPL, and EXEC level Return Code areas *
*-----*
COPY DFHXCPLO.
*-----*
*   Initialise Target information variables. *
*-----*
01  SERVER-DATA.
    03  TARGET-PROGRAM          PIC X(8)  VALUE 'EXCSRVO1'.
    03  TARGET-TRANSID         PIC X(4)  VALUE 'EXCI'.
    03  TARGET-SYSTEM          PIC X(8)  VALUE SPACES.
    03  TARGET-GROUP           PIC X(8)  VALUE SPACES.
01  MESSAGE-LIST.
    03  NO-PARM                 PIC X(42) VALUE
        'EXCCLN01 - No PARM field on EXEC statement'.
    03  FAIL-MSG                PIC X(45) VALUE
        'EXCCLN01 - The INSTALL failed, missing group '.
    03  OK-MSG                  PIC X(48) VALUE
        'EXCCLN01 - The INSTALL was successful for group '.
*-----*
*   Initialise Call level specific variables. *
*-----*
01  BINARY-DATA.
    03  USER-TOKEN              PIC S9(8) COMP VALUE ZERO.
    03  PIPE-TOKEN              PIC S9(8) COMP VALUE ZERO.
    03  LINK-COM-LEN            PIC S9(4) COMP VALUE +80.
    03  LINK-DAT-LEN            PIC S9(4) COMP VALUE +80.
LINKAGE SECTION.
01  PARM-DATA.
    03  PARM-LENGTH              PIC 9(4) COMP.
    03  REAL-PARM-DATA.
        05  PARM-REGION          PIC X(8).
        05  PARM-REGION          PIC X.
        05  PARM-GROUP           PIC X(8).
```



## Exploiting the CICS/Batch Interface (EXCI)

```
PROCEDURE DIVISION              USING PARM-DATA.
*-----*
*   Extract the PARM information from the JCL EXEC statement*
*-----*
      IF PARM-LENGTH = 0          DISPLAY NO-PARM UPON CONSOLE
                                  GOBACK.
      UNSTRING REAL-PARM-DATA     DELIMITED BY ', '
                                  INTO TARGET-SYSTEM
                                  TARGET-GROUP
                                  END-UNSTRING.
*-----*
*   Connect to the relevant CICS Region in a single pass.   *
*-----*
      DISPLAY 'EXCCLN01 - Attempting to INSTALL group '
              TARGET-GROUP UPON SYSOUT.
      EXEC CICS LINK              PROGRAM(TARGET-PROGRAM)
                                  TRANSID(TARGET-TRANSID)
                                  APPLID(TARGET-SYSTEM)
                                  COMMAREA(TARGET-GROUP)
                                  LENGTH(LINK-COM-LEN)
                                  DATALENGTH(LINK-DAT-LEN)
                                  RETCODE(EXCI-EXEC-RETURN-CODE)
                                  SYNCONRETURN
                                  END-EXEC.
      IF TARGET-GROUP = 'NOTFND ' DISPLAY FAIL-MSG PARM-GROUP
                                  UPON SYSOUT
      ELSE                        DISPLAY OK-MSG PARM-GROUP
                                  UPON SYSOUT.
*-----*
*   Logical end of EXCCLN01, return to z/OS.                 *
*-----*
      MOVE ZERO                    TO RETURN-CODE.
      GOBACK.
*-----*
*   Physical end of source code for program EXCCLN01.       *
*-----*
```



## Exploiting the CICS/Batch Interface (EXCI)

### CICS Client Program – EXSRV01

```
CBL CICS('SP')
IDENTIFICATION DIVISION.
PROGRAM-ID.                                EXCSRV01.
ENVIRONMENT DIVISION.
DATA DIVISION.
WORKING-STORAGE SECTION.
*-----*
*  Server program to execute CEDA INSTALL commands issued  *
*  from a batch program via EXCI. (INSTALL is not part of  *
*  the DFHCSDUP utility portfolio of commands.)           *
*-----*
01  DUMP-MARKER                                PIC X(43) VALUE
    'EXCSRV01 - WORKING STORAGE STARTS HERE'.
*-----*
*  COMMON AREA                                           *
*-----*
01  WS-COMMAREA.
    05  COMMAND-ADDR                                USAGE IS POINTER.
    05  COM-LEN-ADDR                               USAGE IS POINTER.
    05  FLAG-ADDR                                  USAGE IS POINTER.
    05  OUTPUT-ADDR                               USAGE IS POINTER.
    05  OUTP-LEN-ADDR                              USAGE IS POINTER.
01  REQUIRED-FIELDS.
    05  LINK-COM-LEN                               PIC S9(4) COMP VALUE +80.
    05  LINK-DAT-LEN                               PIC S9(4) COMP VALUE +80.
    05  MSG1-L                                     PIC S9(4) COMP VALUE +42.
    05  MSG2-L                                     PIC S9(4) COMP VALUE +54.
    05  FLAG-AREA                                  PIC S9(4) COMP VALUE 0.
    05  REAL-FLAGS                                 REDEFINES FLAG-AREA.
        10                                         PIC X.
        10  RESP-FLAG-D                             PIC X.
    05  GROUP-MISSING                              PIC X(8) VALUE 'NOTFND '.
    05  MSG1.
        10                                         PIC X(34) VALUE
            'EXCSRV01 - About to INSTALL group '.
        10  NEW-GROUP                               PIC X(8) VALUE SPACES.
    05  MSG2.
        10                                         PIC X(46) VALUE
            'EXCSRV01 - the following group was not in CSD '.
        10  DUFF-GROUP                              PIC X(8) VALUE SPACES.
01  OUTPUT-LENGTH                                PIC S9(4) COMP VALUE 160.
01  CEDA-COMMAND                                 PIC X(80) VALUE SPACES.
01  OUTPUT-AREA                                  PIC X(160) VALUE SPACES.
01  RESP-FLAG                                     PIC X.
    COPY DFHAID.
LINKAGE SECTION.
01  DFHCOMMAREA.
    05  BATCH-GROUP                                PIC X(8).
    05                                         PIC X(72).
```



## Exploiting the CICS/Batch Interface (EXCI)

```
PROCEDURE DIVISION.
  MOVE BATCH-GROUP          TO NEW-GROUP, DUFF-GROUP.
  EXEC CICS WRITEQ TD       QUEUE('CSML')
                             FROM(MSG1)
                             LENGTH(MSG1-L)
                             END-EXEC.

  MOVE RESP-FLAG-D         TO RESP-FLAG.
  SET COM-LEN-ADDR         TO ADDRESS OF LINK-COM-LEN.
  SET FLAG-ADDR            TO ADDRESS OF RESP-FLAG.
  SET OUTPUT-ADDR          TO ADDRESS OF OUTPUT-AREA.
  SET OUTP-LEN-ADDR        TO ADDRESS OF OUTPUT-LENGTH.
  EXEC CICS HANDLE          CONDITION
                             NOTFND(NO-GROUP)
                             END-EXEC.

  EXEC CICS CSD INSTALL     GROUP(NEW-GROUP)
                             END-EXEC.

*-----*
*   Logical end of program, return to CICS.   *
*-----*
  EXEC CICS RETURN          END-EXEC.
  EXIT.
NO-GROUP.

*-----*
*   This section is entered if the second attempt fails *
*   IOERR which typically means the load module ain't in the *
*   library.                                           *
*-----*
  EXEC CICS WRITEQ TD       QUEUE('CSML')
                             FROM(MSG2)
                             LENGTH(MSG2-L)
                             END-EXEC.

  MOVE GROUP-MISSING        TO BATCH-GROUP.
  EXEC CICS RETURN          END-EXEC.
NO-GROUP-EXIT.
  EXIT.

*-----*
*   The physical end of program EXCSR01.           *
*-----*
```



## Exploiting the CICS/Batch Interface (EXCI)

### Automate CEMT P SHUT from Batch

#### Batch Server Program – EXCLN02

```
IDENTIFICATION DIVISION.
PROGRAM-ID.                                EXCCLN02.
ENVIRONMENT DIVISION.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
    SELECT CEMT-IN                          ASSIGN TO SYSIN.
DATA DIVISION.
FILE SECTION.
FD CEMT-IN                                  BLOCK CONTAINS 0 CHARACTERS
                                           RECORD CONTAINS 80 CHARACTERS
                                           RECORDING MODE IS F.
                                           PIC X(80).

01 COMMAND-IN
WORKING-STORAGE SECTION.
*-----*
*   Declare Call level,DPL, and EXEC level Return Code areas. *
*-----*
COPY DFHXCPL0.
*-----*
*   Initialise Target information variables. *
*-----*
01 CHARACTER-FIELDS.
   03 TARGET-PROGRAM                        PIC X(8)  VALUE 'EXCSRV02'.
   03 TARGET-TRANSID                       PIC X(4)  VALUE 'EXCI'.
   03 TARGET-SYSTEM.
       05 T-SYSC                            PIC X OCCURS 8 TIMES.
01 BINARY-FIELDS.
   03 USER-TOKEN                           PIC S9(8) COMP VALUE ZERO.
   03 PIPE-TOKEN                           PIC S9(8) COMP VALUE ZERO.
   03 LINK-COM-LEN                         PIC S9(4) COMP VALUE +80.
   03 LINK-DAT-LEN                        PIC S9(4) COMP VALUE +80.
   03 SUB                                  PIC S9(4) COMP VALUE +1.
01 CEMT-COMMAND                            PIC X(80) VALUE SPACES.
01 MESSAGE-LIST.
   03 MSG1                                  PIC X(42) VALUE
       'EXCCLN02 - No PARM field on EXEC statement'.
   03 MSG2                                  PIC X(49) VALUE
       'EXCCLN02 - CICS APPLID value exceeds 8 characters'.
LINKAGE SECTION.
01 PARM-DATA.
   05 PARM-LENGTH                          PIC 9(4) COMP.
   05 PARM-REGION                          PIC X OCCURS 8 TIMES.
```



## Exploiting the CICS/Batch Interface (EXCI)

```
PROCEDURE DIVISION                                USING PARM-DATA.
*-----*
*   Extract the PARM information from the JCL EXEC statement.*
*-----*
      IF PARM-LENGTH = 0                          DISPLAY MSG1  UPON SYSOUT
                                                GOBACK.
      IF PARM-LENGTH > 8                          DISPLAY MSG2 UPON SYSOUT
                                                GOBACK.
      PERFORM TEST BEFORE                          VARYING SUB FROM 1 BY 1
                                                UNTIL SUB > PARM-LENGTH
                                                MOVE PARM-REGION (SUB) TO
                                                T-SYSC (SUB)
                                                END-PERFORM.
*-----*
*   Obtain Pseudo-command - it is actually a comment.      *
*-----*
      OPEN INPUT                                  CEMT-IN.
      READ CEMT-IN                                INTO CEMT-COMMAND.
      CLOSE CEMT-IN.
*-----*
*   Connect to the relevant CICS Region in a single pass.  *
*-----*
      DISPLAY 'EXCCLN02 - ' CEMT-COMMAND UPON SYSOUT.
      EXEC CICS LINK                               PROGRAM(TARGET-PROGRAM)
                                                TRANSID(TARGET-TRANSID)
                                                APPLID(TARGET-SYSTEM)
                                                COMMAREA(CEMT-COMMAND)
                                                LENGTH(LINK-COM-LEN)
                                                DATALENGTH(LINK-DAT-LEN)
                                                RETCODE(EXCI-EXEC-RETURN-CODE)
                                                SYNCONRETURN
                                                END-EXEC.
*-----*
*   Logical end of program, return to z/OS                  *
*-----*
      MOVE ZERO                                  TO RETURN-CODE.
      GOBACK.
*-----*
*   Physical end of source for program - EXCCLN02.         *
*-----*
```



## Exploiting the CICS/Batch Interface (EXCI)

### CICS Server program – EXSRV02

```
CBL CICS('SP')
IDENTIFICATION DIVISION.
PROGRAM-ID.                                EXCSR02.
ENVIRONMENT DIVISION.
DATA DIVISION.
WORKING-STORAGE SECTION.
*-----*
*   Server program to execute CEMT SHUTDOWN issued from a   *
*   batch job. This does not issue the command itself, but   *
*   STARTS another transaction, in order that the EXCI links *
*   can be closed prior to shutdown.                         *
*-----*
01  DUMP-MARKER                                PIC X(43) VALUE
    'EXCSR02 - WORKING STORAGE STARTS HERE'.
*-----*
*   COMMON AREA                                           *
*-----*
01  WS-COMMAREA.
    05  COMMAND-ADDR                                USAGE IS POINTER.
    05  COM-LEN-ADDR                               USAGE IS POINTER.
    05  FLAG-ADDR                                  USAGE IS POINTER.
    05  OUTPUT-ADDR                                USAGE IS POINTER.
    05  OUTP-LEN-ADDR                              USAGE IS POINTER.
01  REQUIRED-FIELDS.
    05  LINK-COM-LEN                               PIC S9(4) COMP VALUE 80.
    05  LINK-DAT-LEN                               PIC S9(4) COMP VALUE 80.
    05  FLAG-AREA                                  PIC S9(4) COMP VALUE 0.
    05  REAL-FLAGS                                REDEFINES FLAG-AREA.
        10
        10  RESP-FLAG-D                            PIC X.
01  OUTPUT-LENGTH                                PIC S9(4) COMP VALUE 160.
01  OUTPUT-AREA                                  PIC X(160) VALUE SPACES.
01  RESP-FLAG                                    PIC X.
    COPY DFHAID.
LINKAGE SECTION.
01  DFHCOMMAREA.
    05  COMMAND-IN                                PIC X(80).
*-----*
*   Program Logic starts here.                             *
*-----*
```





## Exploiting the CICS/Batch Interface (EXCI)

```
PROCEDURE DIVISION.  
    MOVE RESP-FLAG-D          TO RESP-FLAG.  
*   SET COMMAND-ADDR         TO ADDRESS OF CEMT-COMMAND.  
    SET COMMAND-ADDR         TO ADDRESS OF COMMAND-IN.  
    SET COM-LEN-ADDR         TO ADDRESS OF LINK-COM-LEN.  
    SET FLAG-ADDR           TO ADDRESS OF RESP-FLAG.  
    SET OUTPUT-ADDR         TO ADDRESS OF OUTPUT-AREA.  
    SET OUTP-LEN-ADDR       TO ADDRESS OF OUTPUT-LENGTH.  
    EXEC CICS START          TRANSID('QUIT')  
                                END-EXEC.  
*-----*  
*   Logical end of the program, so return control to CICS *  
*-----*  
    EXEC CICS RETURN          END-EXEC.  
*-----*  
*   The physical end of program EXCSRVO2. *  
*-----*
```



## Exploiting the CICS/Batch Interface (EXCI)

### Actual SHUTDOWN program

```
CBL CICS('SP')
IDENTIFICATION DIVISION.
PROGRAM-ID.                                EXCSRV99.
ENVIRONMENT DIVISION.
DATA DIVISION.
WORKING-STORAGE SECTION.
*-----*
*   Server program to execute CEMT SHUTDOWN issued. This is   *
*   entered via a START TRAN command from within EXCSRV02 so   *
*   as to allow the EXCI connections to be closed prior to   *
*   shutdown.                                                 *
*-----*
01 DUMP-MARKER                                PIC X(43) VALUE
   'EXCSRV99 - WORKING STORAGE STARTS HERE'.
01 ADHOC-FIELDS.
   05 PAUSE-10                                PIC S9(8) COMP VALUE +10.
   05 MSG1-L                                  PIC S9(4) COMP VALUE +38.
   05 MSG1.
       10                                      PIC X(26) VALUE
         'EXCSRV99 - About to issue '.
       10 MSG-COMM                            PIC X(12) VALUE
         'CEMT P SHUT '.
   COPY DFHAID.
LINKAGE SECTION.
01 DFHCOMMAREA                                PIC X(80).
*-----*
*   Program Logic starts here.                               *
*-----*
PROCEDURE DIVISION.
   EXEC CICS WRITEQ TD                          QUEUE('CSML')
                                           FROM(MSG1)
                                           LENGTH(MSG1-L)
                                           END-EXEC.

   EXEC CICS DELAY                              FOR SECONDS(PAUSE-10)
                                           END-EXEC.

   EXEC CICS PERFORM                            SHUTDOWN
                                           END-EXEC.
*-----*
*   Logical end of the program, so return control to CICS     *
*-----*
   EXEC CICS RETURN                            END-EXEC.
   GOBACK.
*-----*
*   The physical end of program EXCSRV99.                   *
*-----*
```

*This is the end of "Exploiting the CICS/Batch Interface (EXCI)".*