193 CICS

December 2001

In this issue

3 Background to CICS File Control logging and journalling

9 Migrating an existing home banking application from Microsoft IIS to IBM WebSphere

23 Description of the program CSCREATE

32 Support for the COBOL SORT verb in CICS – part 2

44 CICS news

update

CICS Update

Published by

Xephon
27-35 London Road
Newbury
Berkshire RG14 1JL
England
Telephone: 01635 38342
From USA: 01144 1635 38342
E-mail: trevore@xephon.com

North American office

Xephon
PO Box 350100
Westminster, CO 80035-0100
USA
Telephone: 303 410 9344

Subscriptions and back-issues

A year's subscription to *CICS Update*, comprising twelve monthly issues, costs £175.00 in the UK; $270.00 in the USA and Canada; £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 1998 issue, are available separately to subscribers for £16.00 ($24.00) 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.

Editor

Trevor Eddolls

Disclaimer

Readers are cautioned that, although the information in this journal is presented in good faith, neither Xephon nor the organizations or individuals that supplied information in this journal give any warranty or make any representations as to the accuracy of the material it contains. Neither Xephon nor the contributing organizations or individuals accept any liability of any kind howsoever arising out of the use of such material. Readers should satisfy themselves as to the correctness and relevance to their circumstances of all advice, information, code, JCL, and other contents of this journal before making any use of it.

Contributions

When Xephon is given copyright, articles published in *CICS Update* are paid for at the rate of £170 ($260) per 1000 words and £100 ($160) per 100 lines of code for the first 200 lines of original material. The remaining code is paid for at the rate of £50 ($80) per 100 lines. In addition, there is a flat fee of £30 ($50) per article. 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.

Background to CICS File Control logging and journalling

CICS Transaction Server for OS/390 supports a variety of logging and journalling options for files. These can result in some confusion when file definitions are being made or changed. This article describes the various options that may be specified, and explains the intention behind their use.

BACKWARD AND FORWARD RECOVERY

CICS provides two types of recovery processing for files. Backward recovery relates to the backout of recoverable operations against files by tasks that have abended, or by a CICS system that has failed. These operations are performed by dynamic transaction backout processing, or by CICS emergency restart processing, respectively. Forward recovery relates to the reconstruction of changes made to a file. This would typically be required following some form of device failure, such as a head crash to the DASD used for the VSAM dataset associated with the file.

Backout of recoverable resources relies on log data recorded by CICS during File Control request processing. Such log records are referred to as before-images, since they reflect the state of a file's record before it was modified by a particular operation. Before-images are recorded in the CICS system log (DFHLOG), which utilizes the CICS Log Manager and the MVS Logger subsystem to write the log records to a CICS system log stream on either a Coupling Facility or DASD.

CICS Log Manager actually splits the role of the system log into two physically separate log streams – these are called DFHLOG and DFHSHUNT. DFHLOG is used for the majority of system log data recording. DFHSHUNT is used to hold the log data for tasks whose Units Of Work (UOWs) have either not written to the log for some time, or else have failed between prepare and commit processing within a syncpoint operation. However, for the purposes of describing before-image logging by a CICS component such as File Control, it

is sufficient to regard the system log as a single entity.

Forward recovery logging preserves after-images of changes made by CICS File Control requests. These are written to a specified journal defined to CICS. Whereas system recovery writes before-images to the system log, forward recovery utilizes a general log. General logs are therefore the non-system logs handled by CICS, used for purposes such as user journalling, audit trails, etc.

IBM provides the CICS VSAM Recovery program CICSVR to recover lost VSAM data (say as the result of a DASD head crash). CICSVR works independently of CICS; it uses the log data held on forward recovery journals to rebuild VSAM datasets to their previous state. CICSVR helps minimize data recovery time for damaged VSAM spheres.

For more information on CICSVR, operational procedures for CICS recovery, and forward and backward recovery of CICS resources, please refer to the *CICS Recovery and Restart* guide.

DEFINING FILES TO USE BACKWARDS / FORWARDS RECOVERY

The Resource Definition Online (RDO) object FILE defines a file to CICS. The RECOVERY PARAMETERS settings allow the definition to specify what sort of recovery options relate to the file. These options are RECOVERY and FWDRECOVLOG.

RECOVERY can specify *None*, *Backoutonly*, or *All*.

None means that the file is non-recoverable. This means CICS will record neither before-images of changes (to the system log) nor after-images of changes (to the forward recovery journal). Any changes made to the file are therefore committed as they occur, and cannot be automatically backed-out or reinstated.

Backoutonly means that only before-image logging is performed. This means CICS can recover the file to a consistent state should a task or system abend occur after the file had been changed by a task but before that task could commit the changes at syncpoint time. The file is therefore deemed to be recoverable by CICS.

All means that (in addition to backward recovery recording), after-images of changes made to the file are written to the forward recovery journal.

Note that there is no option to have CICS perform forward recovery journalling but not backward recovery logging against the file. If forward recovery journalling is to be performed, the file must also be recoverable to CICS.

RECOVERY PARAMETERS option FWDRECOVLOG allows the specification of the journal to be used for forward recovery journalling of after-images to changes made to the file. If RECOVERY is set to *None* or *Backoutonly*, then FWDRECOVLOG is set to *No*, indicating that no forward recovery journal is used for this file. If RECOVERY is set to *All*, FWDRECOVLOG must contain a number in the range 1 through to 99. This number relates to the last two digits in the journal name to be used as the forward recovery destination for the file. For example 04 would refer to journal DFHJ04.

It should be noted that CICS/ESA 4.1.0 (and earlier releases) differed from CICS Transaction Server in the naming convention for the system log. Whereas CICS/ESA specified that the system log name was DFHJ01, this is now the name of a general log in CICS Transaction Server. As described earlier, system log data in CICS Transaction Server is recorded in DFHLOG (and possibly DFHSHUNT). Therefore, specifying 01 on the forward recovery option FWDRECOVLOG results in after-images of changes made to a file being recorded on general log DFHJ01.

CICS before-image recording means that sufficient information is hardened to the system log before any change to a file is made, to allow the undoing of that operation in a subsequent dynamic transaction backout or emergency restart. For example, an EXEC CICS DELETE request to remove a record on a recoverable file would result in the record being logged to the system log prior to the deletion, so that this image could be used to reinstate the record if the task that performed the deletion were to abend or issue an EXEC CICS SYNCPOINT ROLLBACK command. CICS syncpoint processing ensures that all before-image log records pertaining to the task are physically hardened to the system log's underlying log stream.

CICS after-image recording (to the forward recovery journal) includes the logging of a write-add-complete record when a new record is written to the file, a write-delete record when an existing record is deleted from the file, and a write-update record when a record is modified as the result of an EXEC CICS READ UPDATE / EXEC CICS REWRITE command. CICS ensures that such after-images for changes made by a task are physically hardened to the forward recovery journal's underlying log stream when the task syncpoints and commits the changes.

In addition to such after-image journalling, CICS also stores so-called tie-up records to the forward recovery journal, when files are opened and closed, and when files are being keypointed and backed up. These facilitate forward recovery processing.

A forward recovery of a corrupted dataset would apply the after-images from the forward recovery journal of the file onto an image copy of the dataset taken prior to the corruption. These could include after-images of changes made to the file by inflight tasks. A CICS emergency restart would then use the before-images recorded on the CICS system log to back out any such inflight task activity against the file and return it to a committed state once more.

AUTO JOURNALLING

File definitions also provide the ability to perform another form of journal recording when changes are made to files within CICS. The AUTO JOURNALLING parameters on the FILE definition provide the ability to have CICS record file operations to a user journal.

Automatic journalling support for CICS files predates forward recovery journalling. Unlike true forward recovery, automatic journalling is not related to the UOW performing the changes to the file. That is, information recorded on the CICS system log or forward recovery journal is synchronized with the completion of the UOW, at syncpoint or task detach time. CICS ensures that any such data held in an in-core log buffer is hardened to the physical log streams for the system log and forward recovery journal, at syncpoint time. This is not the case with auto-journalling data – ie no synchronization of data hardening

exists at syncpoint time.

The AUTO JOURNALLING option JOURNAL allows the specification of the journal to be used for auto-journalling operations against the file. As with FWDRECOVLOG, this can be either *No* or else a number in the range 1 through 99. Assuming that a number is specified, various other options can be set.

JNLREAD can be used to make CICS record the fact that the read operations are being performed against the file. This can be specified as *None*, *All*, *Readonly*, or *Updateonly*, depending on whether ordinary reads, read for updates, all reads, or no reads are to be recorded. JNLSYNCREAD is an associated option, specified by either *Yes* or *No*, which indicates whether such auto-journalling of read operations should be synchronous or asynchronous. This selects whether CICS Log Manager should call the MVS Logger subsystem to harden the journal record to the journal's underlying log stream as part of the log request, or whether it can be buffered within CICS and allowed to be flushed to the log stream at a later point in time.

JNLUPDATE controls the recording of rewrite or delete operations, if set to *Yes*.

JNLADD performs a similar role to JNLREAD, but for write operations against the file. This allows either *None*, *Before*, *After*, or *All* to be specified. As such, the file can auto-journal requests before and/or after the I/O request to write a new file record to VSAM. Lastly, JNLSYNCWRITE allows synchronization of auto-journalling for write requests, as per JNLSYNCREAD for read operations.

All the auto-journalling options default to *No* or *None*, as applicable.

THE DFHJ01 JOURNAL

System log data (for backward recovery) is always written to the CICS system log. You cannot redirect this to a journal mapped onto a general log stream. All system log data is recorded by CICS onto DFHLOG (or DFHSHUNT if necessary).

Forward recovery data is recorded onto a journal mapped to a general log stream, as specified by the FWDRECOVLOG option on the FILE definition. Similarly, auto-journalling data is recorded onto a journal as specified by the JOURNAL option.

CICS forward recovery processing is completely independent and unrelated to auto-journalling. However, you can specify that auto-journalled data is recorded to the same journal as forward recovery information, by specifying the same number on the appropriate options within the FILE definitions.

Auto-journalling records are not recommended to be used for VSAM forward recovery purposes. CICS provides additional synchronization and data hardening facilities for data written to forward recovery journals. This ensures data is hardened to the physical log stream for the forward recovery journal at the time that the task which updated the file is committed, by means of a syncpoint operation. Auto-journalling has no such association with CICS syncpointing. It is provided for user-defined purposes, such as for maintaining an audit trail of references and modifications to data on user files. It is not used for CICS recovery purposes.

Both forward recovery and auto-journalling allow the use of journal 1 for their data recording purposes. In CICS Transaction Server, the DFHJ01 journal is a user journal and no different from any other user journal in the range from 1 to 99. However, in CICS/ESA 4.1.0, DFHJ01 was the name used for the system log. BSAM datasets DFHJ01A and DFHJ01B were the dual extents used to record before-image data for logging of recoverable changes made to the system. In CICS/ESA 4.1.0, there was no policing of the use of DFHJ01 as a forward-recovery or auto-journalling destination. In other words, the CICS system log could have contained additional log records unrelated to before-image recovery purposes. This is not the case in CICS Transaction Server; File Control definitions do not allow the use of DFHLOG as a destination for either forward-recovery or auto-journalling.

Since system log data is hardened to its underlying dataset at syncpoint time, one side-effect of using DFHJ01 for auto-journalling purposes in CICS/ESA 4.1.0 was that such data would be hardened

to the dataset when the task that wrote the data committed any recoverable changes to CICS, by issuing an EXEC CICS SYNCPOINT or RETURN command. As stated above, this is not the case in CICS Transaction Server. True synchronization of general log data hardening is provided by CICS forward recovery processing, not auto-journalling.

I hope this article has helped explain the background to logging and journalling within CICS file control processing. Readers wishing to discuss the material in this article further are welcome to contact me via e-mail, at andy_wright@uk.ibm.com.

Andy Wright
CICS Change Team
IBM (UK)

Migrating an existing home banking application from Microsoft IIS to IBM WebSphere

We have a home banking application developed on Microsoft Internet Information Server. Our production data is on an IBM OS/390 mainframe, in DB2 tables and VSAM datasets. Dynamic HTML pages are filled by ASP files that get data from the host database using CICS HLAPI programs. An ActiveX component handles the connection between ASP and an HLAPI program and can be the source of a number of different problems. Generally, this solution has proved its value: in practice more than 30,000 active users per day visit our site and use the different services.

Figure 1 presents a schema of our software platform for that solution.

Our management considered the idea of migrating our home banking to an IBM platform because they expected a large growth in the number of users and wanted to supply them with the best service. I was given the task of migrating existing services to the new platform in the most efficient way. Figure 2 presents the target software platform.

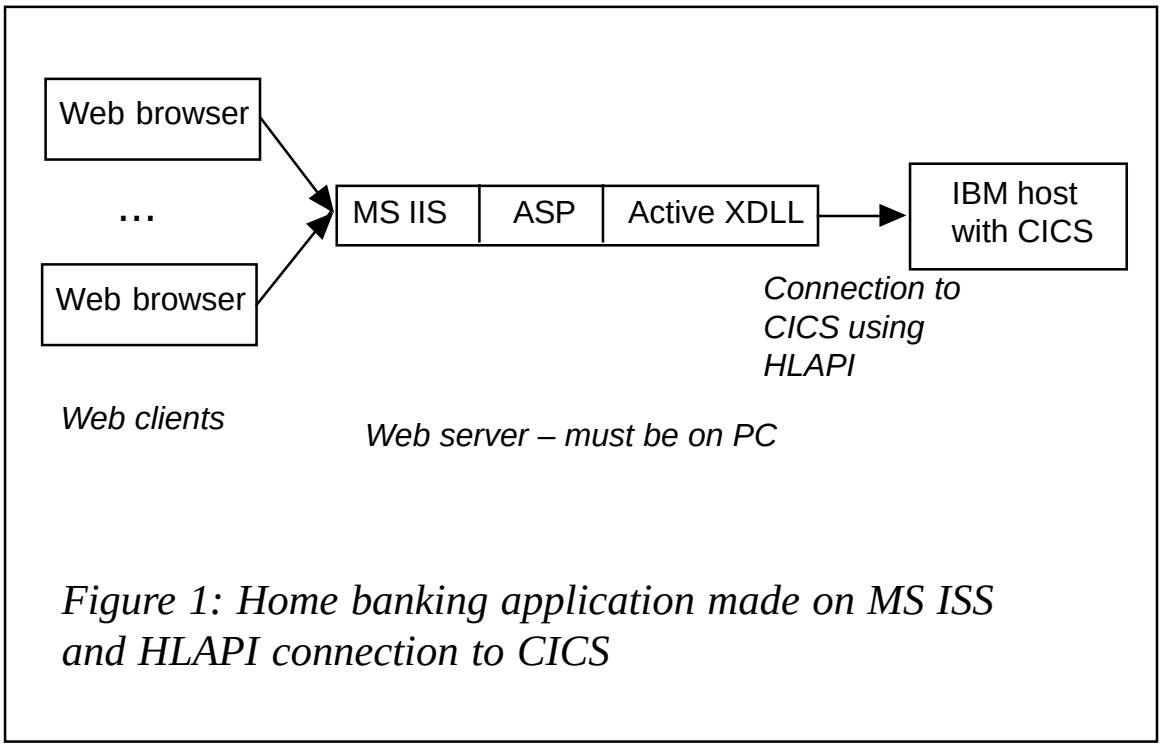

Figure 1: Home banking application made on MS ISS and HLAPI connection to CICS

This migration was carried out with the minimum of modification:

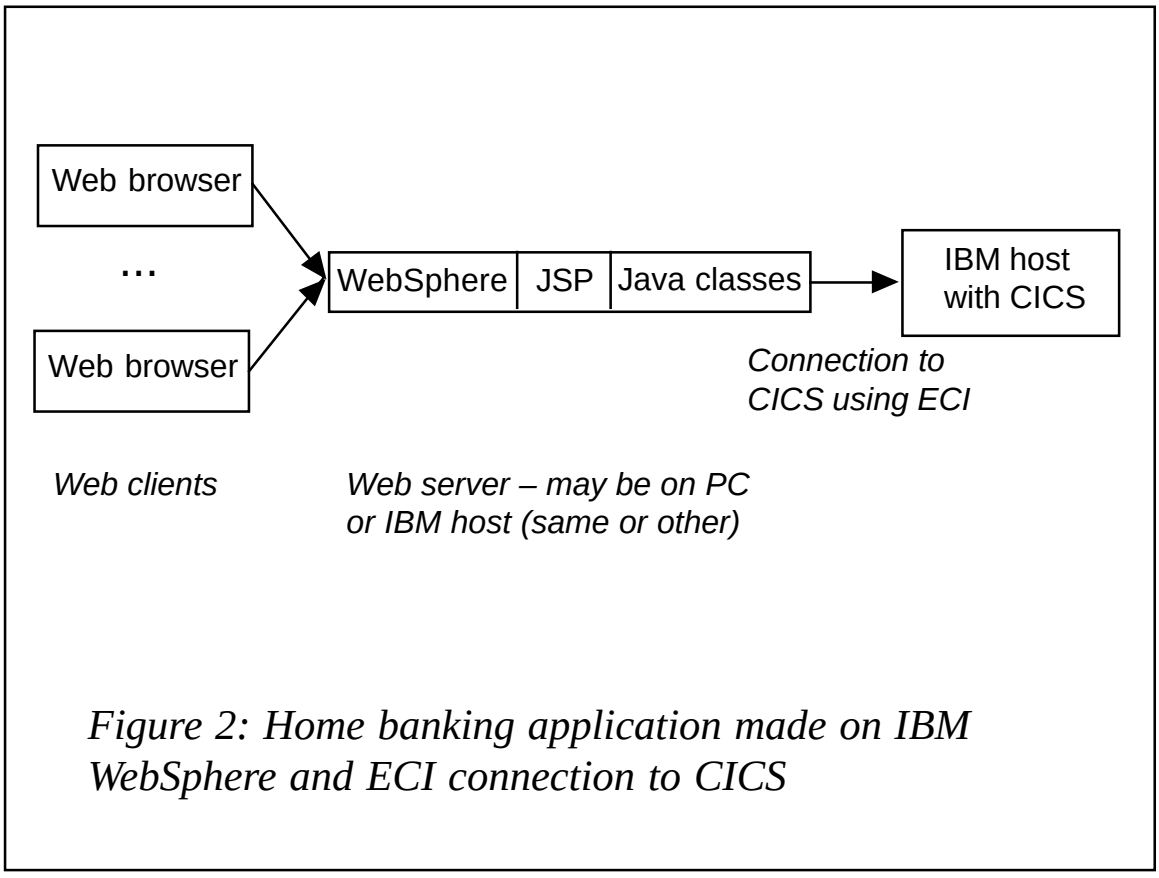

Figure 2: Home banking application made on IBM WebSphere and ECI connection to CICS

1 Existing ASP files were replaced with JSP files:

   - The HTML tags were left unchanged.
   - Any Visual Basic code in the ASP files was replaced with Visual Age for Java code. This task did not require too much effort because the main logic is in the CICS transactions.

2 We replaced existing terminal CICS transactions with CICS DPL programs that will be called from a CICS Universal Client using ECI.

3 Existing communication PC-CICS was via an ActiveX.dll (called GActiveX) and using HLAPI on PC-Web Server opened terminal emulation. We replaced this with new Java classes using IBM CICS Connector for Java and IBM CICS Transaction Gateway via TCP62.

I've selected one typical part of the application to illustrate the process of migrating to another Web platform. This part enables users to enter a username and password; it checks them, and takes the user to the next stage according to the checking results.

On the Microsoft platform, this part consists of:

- An ASP file, LOGIN.ASP, which calls CICS transaction I001 (program I001PLI).
- I001PLI, a program for checking username and password.

In the IBM WebSphere platform, we have the following pieces of code with the equivalent function:

- JSP file LOGIN.JSP, which calls class InetClient.
- InetClient.java, which checks input data and calls class I001.
- I001.java, which communicates with CICS program I001PLI.
- I001PLI, the program for checking username and password.

LOGIN.ASP

```
/*--------------------------------------------------------------------*/

<html><head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-2">
<title>Welcome</title>
</head>

<body bgcolor="#ffffcf">
<%
dim MyIBM
If Request.ServerVariables("REQUEST_METHOD") ="GET" Then
     ' FOR FIRST ATTEMPT SHOW WELCOME MESSAGE
     Msg="PLEASE, ENTER USERNAME AND PASSWORD."
     lUserName=""
     lPassword=""
Else
     lUserName = Trim(Request("fUserName"))
     lPassword = Trim(Request("fPassword"))
     InputOK = False

     ' FOR NONE FIRST ATTEMPT TEST ENTERED DATA
     If lUserName = "" Then
          Msg = "USERNAME IS EMPTY."
     Elseif Len(lUserName) < 5  Then
          Msg = "USERNAME MUST CONTAIN 5 OR MORE CHARACTERS."
     Elseif lPassword = "" Then
          Msg = "PASSWORD IS EMPTY."
     Elseif Len(lPassword) < 5  Then
          Msg = "PASSWORD MUST CONTAIN 5 OR MORE CHARACTERS."
     Else
          InputOK = True
     End If

     if InputOK Then
          ' ENTERED DATA IS CORRECT

                ' VARIABLES MUST BE 15 CHARACTERS
                ' LENGTH FOR CICS TRANSCATION IØØ1
          lUserName15 = String(15 - Len(lUserName)," ")
          lPassword15 = String(15 - Len(lPassword)," ")

          ' COMMUNICATION WITH CICS
          ' VIA SOME ACTIVEX.DLL, WE CALLED THAT "GACTIVEX",
          ' AND HLAPI ON PC-WEB SERVER OPENED TERMINAL EMULATION
          Set MyIBM = Server.CreateObject("GActiveX.Transaction")
          lRequest = "IØØ1" & lUserName15 & lPassword15
          gaOut = MyIBM.Transact(lRequest)

          code = Mid(CStr(gaOut),1,2)
```

```
          If  code = "ØØ" Then
               Msg = "LOGIN SUCCESS. YOU MAY CHOOSE ANY SERVICE."
               Session("UserState") = "Ø"
               Session("UserName")= lUserName
               korisnik = ""
               lozinka = ""
          Else
               Msg = Mid(gaout,3)
          End if
     End If
End if ' Request.ServerVariables("REQUEST_METHOD")
%>
<table border="2" width="1ØØ%">
  <tr>
    <td align="middle" width="755" bgcolor="#cØcØcØ" nowrap>
    <strong><font size="2" face="Arial CE"><bold>
    <% =Msg %></BOLD></font>
    </strong>
    </td>
  </tr>
</table>
<form method="post">
<table border="Ø" cellpadding="Ø" cellspacing="Ø">
<tr>
<td><p align="right"><font face="Arial" size="1">
Username:</font></p>
</td>
<td><input maxlength="15" name="fUserName" size="15"
value="<% =lUserName %>">
</td>
</tr>
<tr>
<td><p align="right"><font face="Arial" size="1">
Password:</font></p>
</td>
<td><input type="password" maxlength="15" name="fPassword"
size="15"
value="<% =lPassword %>">
</td>
</tr>
<tr><td></td><td><input type="submit" value="LOGIN"></td></tr>
</table>
</form>
</body>
</html>
```

I001PLI

```
 /*-------------------------------------------------------------------*/
```

```
I00 1PLI: PROC OPTIONS(MAIN);
/*****************************************************************/
/*                                                               */
/* FUNCTION: CHECK USERNAME AND PASSWORD                         */
/*                                                               */
/* PROGRAM I00 1PLI CONNECTED WITH I00 1 TRANSACTION             */
/*                                                               */
/* INPUT: USERNAME AND PASSWORD                                  */
/*                                                               */
/* OUTPUT: 00 - OK                                               */
/*         01<MESSAGE> - NOT OK                                  */
/*                                                               */
/*****************************************************************/

EXEC SQL INCLUDE TBXE001;    /*TABLE OF USERNAMES*/
EXEC SQL INCLUDE TBXE008;    /*TABLE OF LOGS*/

DCL EVUP316 ENTRY;   /*CHECK CONNECTION CICS-DB2*/

DCL (SELECT, ADDR, BASED) BUILTIN;
DCL (NULL, STG, CSTG) BUILTIN;

DCL 1 I001IN,
      2 TRANID   CHAR(4),
      2 USERNAME CHAR(15),
      2 PASSWORD CHAR(15);
DCL 1 I001OUT CHAR(180);

DCL INDDB2 BIN FIXED(31); /*INDICATOR FOR CONNECTION CICS-DB2*/
DCL INPUTL BIN FIXED(15);

EXEC SQL INCLUDE SQLCA;
EXEC SQL WHENEVER SQLERROR   GO TO SQL_FAIL;
EXEC SQL WHENEVER SQLWARNING GO TO SQL_FAIL;
EXEC SQL WHENEVER NOT FOUND  CONTINUE;
EXEC CICS IGNORE CONDITION LENGERR;

/***************************************************************/

  CALL EVUP316(INDDB2); /* CHECK CONNECTION CICS-DB2 */
  IF INDDB2 = 0 THEN DO;
    I001OUT = '01DATABASE CLOSED. PLEASE, TRY AGAIN LATER.';
    EXEC CICS RETURN;
  END;

  INPUTL=1920;  /* SCREEN SIZE = 1920(80X24) */
  EXEC CICS RECEIVE INTO(I001IN) LENGTH(INPUTL);
  IF INPUTL<34 THEN DO
    I001OUT='02INPUT DATA MUST BE 34 CHARACTERS LENGTH.';
```

```
    EXEC CICS SEND FROM(I001OUT) ERASE;
    EXEC CICS RETURN;
  END;

  DCLTBXE001.INETUSER = I001IN.USERNAME;
  EXEC SQL SELECT INETPASS
           INTO :DCLTBXE001.INETPASS
           FROM INETP.TBXE001
           WHERE INETUSER = :DCLTBXE001.INETUSER;
  IF SQLCODE = 100
  THEN DO;
    /* NOT FOUND USERNAME */
    DCLTBXE008.LOGDESC = 101; /* CODE FOR UNKNOWN USERNAME */
    DCLTBXE008.LOGRESP = 1;
    CALL WRITELOG;
    I001OUT='01UNKNOWN USERNAME.';
    EXEC CICS SEND FROM(I001OUT) ERASE;
    EXEC CICS RETURN;
  END; /* IF SQLCODE=100 THEN */

  IF DCLTBXE001.INETPASS ¬= I001IN.PASSWORD THEN DO;
    DCLTBXE008.LOGDESC = 102; /* CODE FOR INVALID PASSWORD' */
    DCLTBXE008.LOGRESP = 1;
    CALL WRITELOG;
    I001OUT='01INVALID PASSWORD.';
    EXEC CICS SEND FROM(I001OUT) ERASE;
    EXEC CICS RETURN;
  END;

   /* OK */

    DCLTBXE008.LOGDESC = 0;
    DCLTBXE008.LOGRESP = 0;
    CALL WRITELOG;
    I001OUT='00';
    EXEC CICS SEND FROM(I001OUT) ERASE;
    EXEC CICS RETURN;

SQL_FAIL:
  I001OUT = '01DATABASE CLOSED. PLEASE, TRY AGAIN LATER.';
  EXEC CICS SEND FROM(I001OUT) ERASE;
  EXEC CICS RETURN;

WRITELOG: PROC;
  DCLTBXE008.INETUSER = I001IN.USERNAME;
  DCLTBXE008.TERMINAL = EIBTRMID;
  DCLTBXE008.INETPRG  = 'I001';

  EXEC SQL INSERT INTO INETP.TBXE008
          (INETUSER,LOGRESP,LOGDESC,TERMINAL,INETPRG)
```

```
   VALUES(:DCLTBXE008.INETUSER,:DCLTBXE008.LOGRESP,
          :DCLTBXE008.LOGDESC,:DCLTBXE008.TERMINAL,:DCLTBXE008.INETPRG);

 END WRITELOG;

 END I001PLI;

 /*-------------------------------------------------------------------*/

 CREATE TABLE INETP.TBXE001
          (
            INETUSER                        CHAR(15) NOT NULL,
            INETPASS                        CHAR(15) NOT NULL
            ...
          );

 /*-------------------------------------------------------------------*/

 CREATE TABLE INETP.TBXE008
          (
            INETUSER                        CHAR(15) NOT NULL,
            LOGRESP                         INTEGER NOT NULL,
            LOGDESC                         INTEGER NOT NULL,
            TERMINAL                        CHAR(4) NOT NULL,
            INETPRG                         CHAR(4) NOT NULL,
            ...
            LOGDATE                         TIMESTAMP DEFAULT
          );
```

LOGIN.JSP

```
/*-------------------------------------------------------------------*/

<html><head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-2">
<title>Welcome</title>
</head>

<body bgcolor="#ffffcf">
<%
     // FOR FIRST ATTEMPT SHOW WELCOME MESSAGE
     String msg="PLEASE, ENTER USERNAME AND PASSWORD.";
     String lUserName=new String("");
     String lPassword=new String("");
     if(request.getMethod().equals("POST"))
     {
          // FOR NONE FIRST ATTEMPT TEST ENTERED DATA
          lPassword=request.getParameter("fPassword").trim();
          lUserName=request.getParameter("fUserName").trim();
```

```
          postedweb.InetClient ic = new postedweb.InetClient();
          if(ic.loginRequest(request,lUserName,lPassword))
          {
               lUserName="";
               lPassword="";
          }
          msg=ic.messageForClient;
     }
%>
<table border="2" width="100%">
  <tr>
    <td align="middle" width="755" bgcolor="#c0c0c0" nowrap>
    <strong>
        <font size="2" face="Arial CE"><bold> <%= msg %></BOLD></font>
    </strong>  
    </td>
  </tr>
</table>
<form method="post">
<table border="0" cellpadding="0" cellspacing="0">
<tr>
<td><p align="right"><font face="Arial" size="1">
Username:</font></p>
</td>
<td><input maxlength="15" name="fUserName" size="15"
value=" <%= lUserName %>">
</td>
</tr>
<tr>
<td><p align="right"><font face="Arial" size="1">
Password:</font></p>
</td>
<td><input type="password" maxlength="15" name="fPassword"
size="15" value="<%= lPassword %>">
</td>
</tr>
<tr><td></td><td><input type="submit" value="LOGIN"></td></tr>
</table>
</form>
</body>
</html>
```

INETCLIENT.JAVA

```
/*--------------------------------------------------------------------*/

package postedweb;

public class InetClient {
```

```
    public java.lang.String messageForClient;

 public InetClient() {
      super();
 }

 public boolean loginRequest
      (javax.servlet.http.HttpServletRequest request,
       String lUserName,String lPassword)
 {
      short cicsCode;
      boolean result=false;
      if(lUserName.length()==Ø)
           messageForClient="USERNAME IS EMPTY.";
      else if (lUserName.length()<5)
           messageForClient =
                 "USERNAME MUST CONTAIN 5 OR MORE CHARACTERS.";
      else if (lPassword.length()==Ø)
           messageForClient= "PASSWORD IS EMPTY.";
      else if (lPassword.length()<5)
           messageForClient =
                 "PASSWORD MUST CONTAIN 5 OR MORE CHARACTERS.";
      else
      {
            postedweb.IØØ1  t = new postedweb.IØØ1();
            cicsCode=t.run(lUserName,lPassword);
            if(cicsCode==(short)Ø)
            {
               javax.servlet.http.HttpSession ses1=request.getSession();
             ses1.putValue("UserState","ØØ");
               ses1.putValue("UserName",lUserName);
             messageForClient=
                        "LOGIN SUCCESS. YOU MAY CHOOSE ANY SERVICE.";
             result=true;
            }
            else
              messageForClient=t.cicsMessage;
      }
      return result;
 }
}
```

I001.JAVA

```
/*-----------------------------------------------------------------*/

package postedweb;

public class IØØ1 {
```

```
  public java.lang.String cicsMessage;
  public I001() {
      super();
  }
  public short run(String lUserName, String lPassword)
  {
      try
      {
           I001Command com=new I001Command();
           com.setUsername(lUserName);
           com.setPassword(lPassword);
           com.execute();
           cicsMessage=com.getI001msg();
           return com.getI001code();
      }
      catch (Exception e)
      {
           cicsMessage=e.toString();
           return (short)1;
      }
  }
}
```

I001PLI (DPL)

```
 /*-----------------------------------------------------------------*/

 I001PLI: PROC(POINT) OPTIONS(MAIN);
 /*****************************************************************/
 /*                                                               */
 /* FUNCTION: CHECK USERNAME AND PASSWORD                         */
 /*                                                               */
 /* INPUT: USERNAME AND PASSWORD                                  */
 /*                                                               */
 /* OUTPUT: 00 - OK                                               */
 /*         01<MESSAGE> - NOT OK                                  */
 /*                                                               */
 /*****************************************************************/

 EXEC SQL INCLUDE TBXE001;    /*TABLE OF USERNAMES*/
 EXEC SQL INCLUDE TBXE008;    /*TABLE OF LOGS*/

 DCL EVUP316 ENTRY;   /*CHECK CONNECTION CICS-DB2*/

 DCL (SELECT, ADDR, BASED) BUILTIN;
 DCL (NULL, STG, CSTG) BUILTIN;

 DCL POINT PTR;
 DCL 1 ZONE BASED(POINT),
```

```
      2 USERNAME CHAR(15),
      2 PASSWORD CHAR(15),
      2 I001OUT CHAR(180);

DCL INDDB2 BIN FIXED(31); /*INDICATOR FOR CONNECTION CICS-DB2*/

EXEC SQL INCLUDE SQLCA;
EXEC SQL WHENEVER SQLERROR   GO TO SQL_FAIL;
EXEC SQL WHENEVER SQLWARNING GO TO SQL_FAIL;
EXEC SQL WHENEVER NOT FOUND  CONTINUE;

/*************************************************************/

  CALL EVUP316(INDDB2); /* CHECK CONNECTION CICS-DB2 */
  IF INDDB2 = 0 THEN DO;
    I001OUT = '01DATABASE CLOSED. PLEASE, TRY AGAIN LATER.';
    EXEC CICS RETURN;
  END;

  DCLTBXE001.INETUSER = ZONE.USERNAME;
  EXEC SQL SELECT INETPASS
           INTO :DCLTBXE001.INETPASS
           FROM INETP.TBXE001
           WHERE INETUSER = :DCLTBXE001.INETUSER;
  IF SQLCODE = 100
  THEN DO;
    /* NOT FOUND USERNAME */
    DCLTBXE008.LOGDESC = 101; /* CODE FOR UNKNOWN USERNAME */
    DCLTBXE008.LOGRESP = 1;
    CALL WRITELOG;
    I001OUT='01UNKNOWN USERNAME.';
    EXEC CICS RETURN;
  END; /* IF SQLCODE=100 THEN */

  IF DCLTBXE001.INETPASS ¬= ZONE.PASSWORD THEN DO;
    DCLTBXE008.LOGDESC = 102; /* CODE FOR INVALID PASSWORD' */
    DCLTBXE008.LOGRESP = 1;
    CALL WRITELOG;
    I001OUT='01INVALID PASSWORD.';
    EXEC CICS RETURN;
  END;

   /* OK */

    DCLTBXE008.LOGDESC = 0;
    DCLTBXE008.LOGRESP = 0;
    CALL WRITELOG;
    I001OUT='00';
    EXEC CICS RETURN;
```

```
 SQL_FAIL:
   I00101OUT = '01DATABASE CLOSED. PLEASE, TRY AGAIN LATER.';
   EXEC CICS RETURN;

 WRITELOG: PROC;
   DCLTBXE008.INETUSER = ZONE.USERNAME;
   DCLTBXE008.TERMINAL = EIBTRMID;
   DCLTBXE008.INETPRG  = 'I001';

   EXEC SQL INSERT INTO INETP.TBXE008
           (INETUSER,LOGRESP,LOGDESC,TERMINAL,INETPRG)
   VALUES(:DCLTBXE008.INETUSER,:DCLTBXE008.LOGRESP,
          :DCLTBXE008.LOGDESC,:DCLTBXE008.TERMINAL,:DCLTBXE008.INETPRG);

 END WRITELOG;

 END I001PLI;
```

I001COB

```
/*-------------------------------------------------------------------*/

/* Our primary programming language on IBM host is PL/I.             */
/* For work with Enterprise Access Builder in VAJ, we must           */
/* translate only the common area in the small COBOL program.        */

         PROGRAM-ID. I001.
         WORKING-STORAGE SECTION.
         LINKAGE SECTION.
         01 DFHCOMMAREA.
            02 USERNAME PIC X(15).
            02 PASSWORD PIC X(15).
            02 I001CODE PIC 9(2).
            02 I001MSG  PIC X(178).
         PROCEDURE DIVISION.

/*-------------------------------------------------------------------*/
```

Before making class I001, we had to carry out the following steps in Visual Age for Java (using Tools/Enterprise Access Builder):

1 Create I001RecordType importing from COBOL program I001COB.

2 Create I001Record from I001RecordType.

3 Create I001Command using CICSConectionSpec, ECIInteractionSpec, and I001Record.

For more information, see IBM's Redbooks *CICS Connector in VAJ*.

We had to modify the existing CICS transactions in the CICS programs, deleting all terminal statements (like SEND or RECEIVE) and use the common area for any data communication. We cannot use CEDF (or CEDX) for testing these CICS programs, so we made a simple CICS transaction for testing this in CEDF.

FILE: I001TST

```
/*-----------------------------------------------------------------*/

 I00ØTST: PROC OPTIONS(MAIN);
  DCL 1 ZONE,
        2 USERNAME CHAR(15),
        2 PASSWORD CHAR(15),
        2 I00ØRES  CHAR(18Ø);

  USERNAME='DEKI1';
  PASSWORD='12DEKI';
  EXEC CICS LINK PROGRAM('I00ØPLI') COMMAREA(ZONE);
  EXEC CICS RETURN;
 END I00ØTST;

/*-----------------------------------------------------------------*/
```

The WebSphere solution for our home banking application is now in the test phase. We use Web Application Server and CICS Transaction Gateway on the NT platform. We plan to establish a production environment on an OS/390 mainframe without any change to the application. Our first experiences from testing proved that reliability is much higher than on the previous platform.

Our business logic is based on the CICS transactions that use miscellaneous external procedures, the CICS link programs, and work with DB2 tables and VSAM files. Therefore, this platform provides for the simple usage of existing modules that support business logic.

We plan to improve security by implementing client certificates and storing them in LDAP.

Dejan Jelic
Programmer
Postal Savings Bank (Yugoslavia)

Description of the program CSCREATE

One of the many possibilities of using CICS directly via an intranet or the Internet is the ability to call 3270 transactions via a Web browser. From the BMS masks of the called applications, HTML templates must be created, which can be deposited in a file with the DDname DFHHTML. You must insert an appropriate jobcard into your CICS JCL.

An example of the generation of these HTML templates is shown at the end of this article.

To activate this in CICS you need, among other things, the so-called TCPIPSERVICE(s).

These can be defined and administered by using the CEDA transaction. They can also be created via the program CSCREATE, which you can start via PLT processing. You need only the SYSID of the respective CICS, the port this CICS was assigned to, and the name of the TCPIPSERVICE.

Don’t forget to set the parameter TCPIP=YES in your SIT!

```
*ASM XOPTS(CICS SP)
CSCREATE TITLE 'DEFINE TCPIPSERVICES FOR THE WEB-BRIDGE'
         SPACE 1
*---------------------------------------------------------------------*
*                                                                     *
*        MODULE NAME : CSCREATE                                       *
*        AUTHOR      : CLAUS REIS                                     *
*        WRITTEN     : 14.9.2001                                      *
*                                                                     *
*        FUNCTION    : DEFINE TCPIPSERVICES FOR THE WEB-BRIDGE        *
*                                                                     *
*---------------------------------------------------------------------*
*                                                                     *
*        CHANGE ACTIVITY -                                            *
*                                                                     *
*        00-01 ??.??.??  xx     xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx *
*                                                                     *
*---------------------------------------------------------------------*
             EJECT 1
DFHEISTG     DSECT
MESSAGE      DS CL100
```

```
MSGLEN       DC F'Ø'
TCPIPSER     DS CL8
RESPONSE     DS CL4
SYID         DS CL4
             SPACE 1
ATTRIBUTES   DS ØCL(ATTRIBUTES_E-ATTRIBUTES_A)
ATTRIBUTES_A EQU *
ATTR1_ARG    DS CL1Ø
ATTR1_VAL    DS CLØ8
ATTR2_ARG    DS CL1Ø
ATTR2_VAL    DS CLØ5
ATTR3_ARG    DS CL1Ø
ATTR3_VAL    DS CLØ4
ATTR4_ARG    DS CLØ1
ATTRIBUTES_E EQU *
             EJECT 1
*-------------------------------------------------------------------*
*        MAIN PROGRAM                                               *
*-------------------------------------------------------------------*
         SPACE 1
CSCREATE DFHEIENT CODEREG=CODEREG,DATAREG=DATAREG,EIBREG=EIBREG
         SPACE 1
CSCREATE AMODE 31
CSCREATE RMODE ANY
         SPACE 1
         MVC   MESSAGE,BLANK                  SEND
**
         MVC   MESSAGE(MSGØØØ_L),MSGØØØ       START
**
         BAS   SUBREG,SENDMSG                 MESSAGE
**
         SPACE 1
**
         EXEC  CICS ASSIGN SYSID(SYID)                              *
                           RESP(RESPONSE)

         SPACE 1
         CLC   RESPONSE,DFHRESP(NORMAL)
         BE    CHETAB
         MVC   MESSAGE,BLANK                  SYSID CAN'T
         MVC   MESSAGE(MSGØØ1_L),MSGØØ1       BE ASSIGNED
**
         BAS   SUBREG,SENDMSG                 SEND MESSAGE
         B     RETURN
         SPACE 1
CHETAB   DS    ØH'Ø'
         LA    R6,PORTTAB                     FIRST ENTRY
         SPACE 1
CHETABØ  DS    ØH'Ø'
         CLI   Ø(R6),C'*'                     END OF TABLE?
```

```
         BNE   CHETAB1
         MVC   MESSAGE,BLANK                  NO VALID PORT CAN
         MVC   MESSAGE(MSGØØ2_L),MSGØØ2       BE ASSIGNED
**
         BAS   SUBREG,SENDMSG                 SEND MESSAGE
         B     RETURN
         SPACE 1
CHETAB1  DS    ØH'Ø'
         CLC   SYID,Ø(R6)                     VALID CICS?
         BE    FILLARG
         LA    R6,L'PORTTAB(,R6)              POINT TO NEXT SYSID-ENTRY
         B     CHETABØ
         SPACE 1
FILLARG  DS    ØH'Ø'
         MVC   TCPIPSER,BLANK                 FORMAT NAME !!!
         MVC   TCPIPSER(4),9(R6)              NAME OF TCPIPSERVICE
         MVC   ATTR1_ARG,=CL(L'ATTR1_ARG)'U('
         MVC   ATTR2_ARG,=CL(L'ATTR2_ARG)')P('
         MVC   ATTR3_ARG,=CL(L'ATTR3_ARG)')TR('
         MVC   ATTR4_ARG,=CL(L'ATTR4_ARG)')'
         MVC   ATTR1_VAL,=CL8'DFHWBADX'
         MVC   ATTR2_VAL,4(R6)                PORT NBR.
         MVC   ATTR3_VAL,=CL4'CWXN'
         SPACE 1
CREATE   DS    ØH'Ø'
         EXEC  CICS CREATE TCPIPSERVICE(TCPIPSER)                      *
                        ATTRIBUTES(ATTRIBUTES)                         *
                        ATTRLEN(ATTRLEN)                               *
                        RESP(RESPONSE)
         SPACE 1
         CLC   RESPONSE,DFHRESP(NORMAL)
         BE    RETURN
*        CLC   RESPONSE,DFHRESP(INVREQ)                 ALREADY OPEN ?
*        BE    RETURN
         MVC   MESSAGE,BLANK
         MVC   MESSAGE(MSGØØ3_L),MSGØØ3
**
         MVC   MESSAGE+35(L'TCPIPSER-4),TCPIPSER
         BAS   SUBREG,SENDMSG                           SEND MESSAGE
         SPACE 1
RETURN   DC    ØH'Ø'
         EXEC  CICS RETURN
         SPACE 1
ABEND    DC    ØH'Ø'
         EXEC  CICS ABEND ABCODE('IPCO')
*-------------------------------------------------------------------*
*        MESSAGE ROUTINE                                            *
*-------------------------------------------------------------------*
         SPACE 1
SENDMSG  DC    ØH'Ø'
```

```
         MVC  MSGLEN,=A(L'MESSAGE)
         SPACE 1
         EXEC CICS WRITE OPERATOR TEXT (MESSAGE)                       *
                                  TEXTLENGTH (MSGLEN)                  *
                                  ROUTECODES (ELEVEN) NUMROUTES(ONE)
*        (EXPIRED,INVREQ,LENGERR)
         SPACE 1
MESSAGE9 DC    Ø H'Ø'
         BR    SUBREG                 RETURN TO CALLER
         EJECT
*-------------------------------------------------------------------*
         DFHEIRET
         SPACE 1
*-------------------------------------------------------------------*
*        EQUATES  /  CONSTANTS                                      *
*-------------------------------------------------------------------*
         SPACE 1
BLANK    DC    CL1ØØ' '
ATTRLEN  DC    AL2(ATTRIBUTES_E-ATTRIBUTES)
ELEVEN   DC    B'ØØØØ1Ø11'            ROUTECODE 11
ONE      DC    F'1'
         SPACE 1
PORTTAB  DS    ØCL13
*
*              SYSID   PORT     NAME OF TCPIPSERVICE
*
         DC    C'SSDØ',C'Ø555Ø',C'SYST'  SYSTEMCICS
         DC    C'SSTØ',C'Ø5551',C'SYST'
         DC    C'SSØØ',C'Ø5552',C'SYST'
         DC    C'SSØ1',C'Ø5553',C'SYST'
         DC    C'SVTØ',C'Ø5554',C'VPRD'  VPRDCICS
         DC    C'SVVØ',C'Ø5555',C'VPRD'
         DC    C'SVØØ',C'Ø5556',C'VPRD'
         DC    C'SVØ1',C'Ø5557',C'VPRD'
         DC    C'SPIN',C'Ø5558',C'PROD'  PRODCICS OHNE PAISY
         DC    C'SPOD',C'Ø5559',C'PROD'
         DC    C'SPTØ',C'Ø556Ø',C'PROD'
         DC    C'SPVØ',C'Ø5561',C'PROD'
         DC    C'SPØØ',C'Ø5562',C'PROD'
         DC    C'SPØ1',C'Ø5563',C'PROD'
         DC    C'SPØ2',C'Ø5564',C'PROD'
         DC    C'SPØ3',C'Ø5565',C'PROD'
         DC    C'SPØ4',C'Ø5566',C'PROD'
         DC    C'SP9Ø',C'Ø5567',C'PROD'
         DC    C'STOD',C'Ø5568',C'TEST'  TESTCICS
         DC    C'STTØ',C'Ø5569',C'TEST'
         DC    C'STVØ',C'Ø557Ø',C'TEST'
         DC    C'STØØ',C'Ø5571',C'TEST'
         DC    C'STØ1',C'Ø5572',C'TEST'
         DC    C'ST1Ø',C'Ø5573',C'TEST'
```

```
         DC    C'ST11',C'Ø5574',C'TEST'
         DC    C'ST12',C'Ø5575',C'TEST'
         DC    C'ST13',C'Ø5576',C'TEST'
         DC    C'ST2Ø',C'Ø5577',C'TEST'
         DC    C'ST21',C'Ø5578',C'TEST'
         DC    C'ST22',C'Ø5579',C'TEST'
         DC    C'ST23',C'Ø558Ø',C'TEST'
         DC    C'ST24',C'Ø5581',C'TEST'
         DC    C'ST25',C'Ø5582',C'TEST'
         DC    C'ST26',C'Ø5583',C'TEST'
         DC    C'ST27',C'Ø5584',C'TEST'
         DC    C'ST3Ø',C'Ø5585',C'TEST'
         DC    C'ST31',C'Ø5586',C'TEST'
         DC    C'ST32',C'Ø5587',C'TEST'
         DC    C'*'
         SPACE 1
*-------------------------------------------------------------------*
*        MESSAGES                                                   *
*-------------------------------------------------------------------*
         SPACE 1
MSGØØØ   DC    C'CSCREATE-ØØØ Create HTTP/HTTPS - services'
**
MSGØØØ_L EQU   *-MSGØØØ
**
         SPACE 1
MSGØØ1   DC    C'CSCREATE-ØØ1 SYSID can not be assigned - please contac*
               t the CICS Systems Programmer'
**
MSGØØ1_L EQU   *-MSGØØ1
**
         SPACE 1
**
MSGØØ2   DC    C'CSCREATE-ØØ2 No valid port found - please contact the *
               CICS Systems Programmer'
**
MSGØØ2_L EQU   *-MSGØØ2
**3
         SPACE 1
MSGØØ3   DC    C'CSCREATE-ØØ3 Create for tcpipservice ???? failed - see*
                DFH-message'
**
MSGØØ3_L EQU   *-MSGØØ3
**
         SPACE 1
         LTORG
         DFHEJECT
*-------------------------------------------------------------------*
*        REGISTER USAGE                                             *
*-------------------------------------------------------------------*
         SPACE 1
```

```
EIBREG   EQU   R7            EIB POINTER (DEFAULT)
SUBREG   EQU   R8            SUBROUTINE REGISTER
CODEREG  EQU   R11           BASE REGISTER
DATAREG  EQU   R12           EISTG REGISTER
         SPACE 1
         DFHREGS
         SPACE 1
         END   CSCREATE
```

JCL

```
//YUSERIDX JOB Ø02665,'DEVELOPERS NAME',NOTIFY=YUSERID,
//*-------------------------------------------------------------
//*   JOB  SUBMITTED FROM YUSERID.MAIN.JCL(CRHTMLTE)
//*   DOC: GENERATE HTML-TEMPLATES
//*   GRP: YOUR ARCHIVE GROUP
//*   DATE: 14.Ø9.Ø1, TIME: 2Ø:ØØ
//*-------------------------------------------------------------
//             CLASS=T,USER=YUSERID,MSGCLASS=X,REGION=4M,RESTART=*
//*-------------------------------------------------------------
//DFHMAPT PROC INDEX='CICSTS13.IBM', FOR SDFHMAC
//             MAPLIB='CICS.SYST.PPLOAD',       TARGET FOR MAP
//             DSCTLIB='YUSERID.MAIN.SOURCE',   TARGET FOR DSECT
//             TEMPLIB='CICSTS13.SYST.DFHHTML', TARGET FOR TEMPLATES
//             MAPNAME=CMCSGM,               NAME OF MAPSET - REQUIRED
//             A=,                           A=A FOR ALIGNED MAP
//             RMODE=24,                     24/ANY
//             ASMBLR=ASMA9Ø,                ASSEMBLER PROGRAM NAME
//             REG=2Ø48K,                    REGION FOR ASSEMBLY
//             OUTC=*,                       PRINT SYSOUT CLASS
//             WORK=SYSDA                    WORK FILE UNIT
//COPY     EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=&OUTC
//SYSUT2   DD DSN=&&TEMPM,UNIT=&WORK,DISP=(,PASS),
//            DCB=(RECFM=FB,LRECL=8Ø,BLKSIZE=4ØØ),
//            SPACE=(4ØØ,(5Ø,5Ø))
//SYSIN    DD DUMMY
//*SYSUT1   DD * NEEDED FOR THE MAP SOURCE
//ASMMAP   EXEC PGM=&ASMBLR,REGION=&REG,
//  PARM='SYSPARM(&A.MAP),DECK,NOOBJECT'
//SYSPRINT DD SYSOUT=&OUTC
//SYSLIB   DD DSN=&INDEX..SDFHMAC,DISP=SHR
//         DD DSN=SYS1.MACLIB,DISP=SHR
//SYSUT1   DD UNIT=&WORK,SPACE=(CYL,(5,5))
//SYSUT2   DD UNIT=&WORK,SPACE=(CYL,(5,5))
//SYSUT3   DD UNIT=&WORK,SPACE=(CYL,(5,5))
//SYSPUNCH DD DSN=&&MAP,DISP=(,PASS),UNIT=&WORK,
//            DCB=(RECFM=FB,LRECL=8Ø,BLKSIZE=4ØØ),
//            SPACE=(4ØØ,(5Ø,5Ø))
```

```
//SYSIN    DD DSN=&&TEMPM,DISP=(OLD,PASS)
//LINKMAP  EXEC PGM=IEWL,PARM='LIST,LET,XREF,RMODE(&RMODE)'
//SYSPRINT DD SYSOUT=&OUTC
//SYSLMOD  DD DSN=&MAPLIB(&MAPNAME),DISP=SHR
//SYSUT1   DD UNIT=&WORK,SPACE=(1Ø24,(2Ø,2Ø))
//SYSLIN   DD DSN=&&MAP,DISP=(OLD,DELETE)
//ASMDSECT EXEC PGM=&ASMBLR,REGION=&REG,
//  PARM='SYSPARM(&A.DSECT),DECK,NOOBJECT'
//SYSPRINT DD SYSOUT=&OUTC
//SYSLIB   DD DSN=&INDEX..SDFHMAC,DISP=SHR
//         DD DSN=SYS1.MACLIB,DISP=SHR
//SYSUT1   DD UNIT=&WORK,SPACE=(CYL,(5,5))
//SYSUT2   DD UNIT=&WORK,SPACE=(CYL,(5,5))
//SYSUT3   DD UNIT=&WORK,SPACE=(CYL,(5,5))
//SYSPUNCH DD DSN=&DSCTLIB(&MAPNAME),DISP=OLD
//SYSIN    DD DSN=&&TEMPM,DISP=(OLD,PASS)
//ASMTEMPL EXEC PGM=&ASMBLR,REGION=&REG,
//  PARM='SYSPARM(TEMPLATE,TCSGM),DECK,NOOBJECT'
//SYSPRINT DD SYSOUT=&OUTC
//SYSLIB   DD DSN=&INDEX..SDFHMAC,DISP=SHR
//         DD DSN=SYS1.MACLIB,DISP=SHR
//SYSUT1   DD UNIT=&WORK,SPACE=(CYL,(5,5))
//SYSUT2   DD UNIT=&WORK,SPACE=(CYL,(5,5))
//SYSUT3   DD UNIT=&WORK,SPACE=(CYL,(5,5))
//SYSPUNCH DD UNIT=&WORK,SPACE=(CYL,(5,5)),DISP=(,PASS)
//SYSIN    DD DSN=&&TEMPM,DISP=(OLD,DELETE)
//UPDTEMPL EXEC PGM=IEBUPDTE,REGION=&REG,PARM=NEW
//SYSPRINT DD SYSOUT=&OUTC
//SYSIN    DD DSN=*.ASMTEMPL.SYSPUNCH,DISP=(OLD,DELETE)
//SYSUT2   DD DSN=&TEMPLIB,DISP=SHR
//DFHMAPT  PEND
//DOIT     EXEC DFHMAPT
//COPY.SYSUT1 DD * INSERT BMS-DECK
*
*              !!!!!!!!THIS IS ONLY A SAMPLE !!!!!!!!
*
         MACRO
         TCSGM
         DFHMDX MAPSET=*,MAP=*,                                        *
               TITLE='327Ø TRANSACTION (CSGM) IM INTRA- BZW. INTERNET  *
               BMS MAP (CMCSGM) CONVERTED TO HTML',                    *
               BGCOLOR=BLUE,                                           *
               TEXT=BLACK,                                             *
               RESET=NO
         MEND
*
CMCSGM   TITLE 'CMCSGM    -   NLV   G O O D M O R N I N G   MESSAGE'
CMCSGM   DFHMSD TYPE=&SYSPARM,MODE=OUT,LANG=ASM,                       *
               STORAGE=AUTO,TIOAPFX=YES
         SPACE 3
```

```
MCSGM    DFHMDI SIZE=(24,8Ø),LINE=1,COLUMN=1,TIOAPFX=YES,              *
               HILIGHT=OFF,COLOR=DEFAULT,                            *
               MAPATTS=(COLOR,HILIGHT,OUTLINE)
         DFHMDF POS=(1,1),LENGTH=4,INITIAL='ZSSN',ATTRB=(FSET,ASKIP)
         DFHMDF POS=(1,7),LENGTH=1
         DFHMDF POS=(1,24),LENGTH=17,INITIAL='CICS/ESA  Version'
Z1VERS   DFHMDF POS=(1,42),LENGTH=3
Z1DATE   DFHMDF POS=(1,57),LENGTH=8
Z1TIME   DFHMDF POS=(1,66),LENGTH=5
Z1TERM   DFHMDF POS=(1,72),LENGTH=4
         DFHMDF POS=(1,77),LENGTH=3,INITIAL='SYS'
Z2LEFT   DFHMDF POS=(Ø2,Ø1),LENGTH=2Ø,ATTRB=(ASKIP),COLOR=YELLOW
Z2RIGHT  DFHMDF POS=(Ø2,57),LENGTH=23,ATTRB=(ASKIP),COLOR=TURQUOISE
Z3LEFT   DFHMDF POS=(Ø3,Ø1),LENGTH=2Ø,ATTRB=(ASKIP),COLOR=YELLOW
Z3RIGHT  DFHMDF POS=(Ø3,57),LENGTH=23,ATTRB=(ASKIP),COLOR=TURQUOISE
Z4LEFT   DFHMDF POS=(Ø4,Ø1),LENGTH=2Ø,ATTRB=(ASKIP),COLOR=YELLOW
Z4RIGHT  DFHMDF POS=(Ø4,57),LENGTH=23,ATTRB=(ASKIP),COLOR=TURQUOISE
Z5LEFT   DFHMDF POS=(Ø5,Ø1),LENGTH=2Ø,ATTRB=(ASKIP),COLOR=YELLOW
         DFHMDF POS=(Ø5,38),LENGTH=4,INITIAL='NLV>',                 *
               COLOR=BLUE,HILIGHT=REVERSE
         DFHMDF POS=(Ø5,43),LENGTH=1
Z5RIGHT  DFHMDF POS=(Ø5,57),LENGTH=23,ATTRB=(ASKIP),COLOR=TURQUOISE
Z6LEFT   DFHMDF POS=(Ø6,Ø1),LENGTH=2Ø,ATTRB=(ASKIP),COLOR=YELLOW
         DFHMDF POS=(Ø6,41),LENGTH=1,INITIAL='I',                    *
               COLOR=BLUE,HILIGHT=REVERSE
         DFHMDF POS=(Ø6,43),LENGTH=1
Z6RIGHT  DFHMDF POS=(Ø6,57),LENGTH=23,ATTRB=(ASKIP),COLOR=NEUTRAL
         DFHMDF POS=(Ø7,4Ø),LENGTH=3,INITIAL='ANA'
Z7RIGHT  DFHMDF POS=(Ø7,57),LENGTH=23,ATTRB=(ASKIP),COLOR=NEUTRAL
         DFHMDF POS=(Ø8,39),LENGTH=5,INITIAL='ANNNA'
         DFHMDF POS=(Ø8,52),LENGTH=4,INITIAL='NAV>',                 *
               COLOR=RED,HILIGHT=BLINK
Z8RIGHT  DFHMDF POS=(Ø8,57),LENGTH=23,ATTRB=(ASKIP),COLOR=NEUTRAL
         DFHMDF POS=(Ø9,39),LENGTH=5,INITIAL='NNNNN'
         DFHMDF POS=(Ø9,55),LENGTH=1,INITIAL='I'
         DFHMDF POS=(1Ø,4Ø),LENGTH=3,INITIAL='NNN'
         DFHMDF POS=(1Ø,54),LENGTH=3,INITIAL='ANA'
         DFHMDF POS=(11,2Ø),LENGTH=3,INITIAL='NNA'
         DFHMDF POS=(11,33),LENGTH=11,INITIAL='ANNNNNNNNNN'
         DFHMDF POS=(11,52),LENGTH=7,INITIAL='I NNN I'
         DFHMDF POS=(12,2Ø),LENGTH=4,INITIAL='NNNA'
         DFHMDF POS=(12,33),LENGTH=12,INITIAL='ANNNNNNNNNN'
         DFHMDF POS=(12,52),LENGTH=7,INITIAL='NNNNNNN'
         DFHMDF POS=(13,2Ø),LENGTH=4Ø,                               *
               INITIAL='NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN'
         DFHMDF POS=(14,2Ø),LENGTH=4Ø,                               *
               INITIAL='NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN'
         DFHMDF POS=(15,2Ø),LENGTH=1Ø,INITIAL='NNNNNNNNNN'
         DFHMDF POS=(15,4Ø),LENGTH=5,INITIAL='NNNNN'
         DFHMDF POS=(15,5Ø),LENGTH=1Ø,INITIAL='NNNNNNNNNN'
```

```
         DFHMDF POS=(16,20),LENGTH=10,INITIAL='NNNNNNNNNN'
         DFHMDF POS=(16,35),LENGTH=2,INITIAL='NN'
         DFHMDF POS=(16,41),LENGTH=4,INITIAL='NNNN'
         DFHMDF POS=(16,50),LENGTH=10,INITIAL='NNNNNNNNNN'
         DFHMDF POS=(17,20),LENGTH=10,INITIAL='NNNNNNNNNN'
         DFHMDF POS=(17,35),LENGTH=3,INITIAL='NNN'
         DFHMDF POS=(17,42),LENGTH=3,INITIAL='NNN'
         DFHMDF POS=(17,50),LENGTH=10,INITIAL='NNNNNNNNNN'
         DFHMDF POS=(18,20),LENGTH=10,INITIAL='NNNNNNNNNN'
         DFHMDF POS=(18,35),LENGTH=4,INITIAL='NNNN'
         DFHMDF POS=(18,43),LENGTH=2,INITIAL='NN'
         DFHMDF POS=(18,50),LENGTH=10,INITIAL='NNNNNNNNNN'
         DFHMDF POS=(19,20),LENGTH=10,INITIAL='NNNNNNNNNN'
         DFHMDF POS=(19,35),LENGTH=5,INITIAL='NNNNN'
         DFHMDF POS=(19,50),LENGTH=10,INITIAL='NNNNNNNNNN'
         DFHMDF POS=(20,20),LENGTH=40,                                 *
               INITIAL='NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN'
         DFHMDF POS=(21,20),LENGTH=40,                                 *
               INITIAL='NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN'
         DFHMDF POS=(23,01),LENGTH=79,ATTRB=(BRT,ASKIP),               *
               COLOR=PINK,HILIGHT=BLINK,                               *
               INITIAL='>> Durch betätigen der Datenfreigabe-Taste erha*
               lten Sie das ZSS-Anmeldebild <<'
         DFHMDF POS=(24,01),LENGTH=79,INITIAL='Ihre Benutzer-Service-St*
               elle erreichen Sie unter der Telefon-Nummer 7777'
         DFHMSD TYPE=FINAL
         END
/*
//
```

Claus Reis
CICS Systems Programmer
Nuernberger Lebensversicherung AG (Germany)

Why not share your expertise and earn money at the same time? *CICS Update* is looking for technical articles and hints and tips that experienced CICS users have written to make their life, or the lives of their users, easier. We would also be interested in articles about performance and tuning of CICS TS.

Articles can be of any length and can be sent to any of the addresses shown on page 2 or e-mailed to Trevor Eddolls at trevore@xephon.com. A copy of our *Notes for contributors* is available from our Web site at www.xephon.com/nfc.

Support for the COBOL SORT verb in CICS – part 2

This month we conclude the article that describes a CICS compatible SORT routine, callable from general-purpose COBOL programs via the SORT verb.

```
*
         CLC   =C'FILSZ=',87(R4)     Is it FILSZ
         BNE   PARS98Ø               No - nothing else supported
PARS94Ø  EQU   *                     Process the FILSZ= data
*        *     *
         CLI   93(R4),C'E'           Is it Estimate?     If not...
         BNE   PARS98Ø               Give up, 'cos we can't handle it
*        *     *
         PACK  WORKPACK,94(8,R4)     Pack the length value
         CVB   R9,WORKPACK           Make it binary
         STH   R9,FILESIZE           Save the FILESIZE (number of recs)
         CH    R9,=AL2(MAXRECS)      Too many records?
         BNH   PARS99Ø               No, so we are done
*                                    Yes, too many records to sort!
         EXEC CICS WRITEQ TD                                          X
               QUEUE(NBUG)                                            X
               FROM(SRTERRØ9)                                         X
               LENGTH(=AL2(L'SRTERRØ9))                               X
               NOHANDLE
*
         B     RETURN                Give up immediately.
*
PARS98Ø  EQU   *
         SR    R7,R7                 Set parse length to zero
*        *     *
*
PARS99Ø  EQU   *
         CLC   MAINSIZE,=F'Ø'        MAINSIZE set?
         BE    PARS995               No, so no good - don't set good
         CLC   FILESIZE,=H'Ø'        FILESIZE set?
         BE    PARS995               No, so no good - don't set good
         MVI   OPTIONKW,C'Y'         Everything Ok. Flag good parms
*
PARS995  EQU   *
         L     R14,PROCRET3          Restore return addres
         BR    R14                   Go back.
         DROP  R6
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

```
*********************************************************************
*
*        GETSTG ensures that the SORT task queue is deleted so that
*        we can use it to store the records, and acquires storage
*        needed for processing. The storage needed is:
*         (KEYLEN+2)*(FILSZ+1) for table and swap storage
*         Storage for all records in a MAIN TS queue as above
*
*********************************************************************
GETSTG   EQU   *                    Acquire storage for the sort
         ST    R14,PROCRET1         Save the subroutine return address
*        *     *
         LH    R8,KEYLEN            Get key length
         LA    R8,2(R8)             Add 2 for TSQ item number
         STH   R8,KEYLENA2          Save it away
*
         LH    R9,FILESIZE          Get number of records
         MH    R9,KEYLENA2          Calculate storage required
         ST    R9,MAINSIZE          Ignore what was passed
*
         EXEC CICS GETMAIN                                          X
               SET(R9)                                              X
               FLENGTH(MAINSIZE)                                    X
               NOSUSPEND                                            X
               NOHANDLE
         CLC   EIBRESP,DFHRESP(NORMAL) Good Getmain?
         BE    GETS100              Yes, so keep going
*
         EXEC CICS WRITEQ TD                                        X
               QUEUE(NBUG)                                          X
               FROM(SRTERR13)                                       X
               LENGTH(=AL2(L'SRTERR13))                             X
               NOHANDLE
*
         B     RETURN               Give up immediately.
*
GETS100  EQU   *
         ST    R9,TABPTR            Sort table address (FILSZ entries)
         EXEC CICS DELETEQ TS                                       X
               QUEUE(SORTQNM)                                       X
               NOHANDLE
*
         L     R14,PROCRET1         Restore return address
         BR    R14                  Return to caller
*
*********************************************************************
GETRECS  EQU   *                    Get the records from E15
         ST    R14,PROCRET1         Save the subroutine return address
*        *     *
         LA    R6,100               Set up 100 point counter
```

```
GETR1ØØ  EQU   *                    Set up the pass area for E15
         SR    R8,R8                 Zero value
         ST    R8,E15NEWA            No new records.
*
*        User Exit Address Constant is already set up.
*        *     *
GETR2ØØ  EQU   *                    Loop until RC=12
*                                    Give CICS a chance to reschedule
         BCT   R6,GETR21Ø            1ØØ done? Don't suspend unless so
         EXEC CICS SUSPEND NOHANDLE
         LA    R6,1ØØ                Skip another 1ØØ times
GETR21Ø  EQU   *
         L     R15,SPE15A            Get address of E15 exit
         LA    R1,E15PARM            Set up parameter block address
         BALR  R14,R15               Execute the E15 exit to get a rec
         C     R15,=F'Ø'             No action:
         BE    GETR2ØØ
         C     R15,=F'4'             Delete - > No action:
         BE    GETR2ØØ
         C     R15,=F'8'             End of input: stop calling
         BE    GETR99Ø
         C     R15,=F'12'            Insert a record:
         BE    GETR3ØØ
         EXEC CICS WRITEQ TD                                          X
               QUEUE(NBUG)                                            X
               FROM(SRTERR14)                                         X
               LENGTH(=AL2(L'SRTERR14))                               X
               NOHANDLE
*
         LA    R8,16                 Set return code
         ST    R8,RETCODE
         B     RETURN                Give up with prejudice
*
GETR3ØØ  EQU   *                    We have a record to insert
         LTR   R7,R1                 Get the address of the record
         BZ    GETR2ØØ               No record -> get another one
         MVC   QRECLEN,LRECL         Set up length for WRITEQ
         CLI   RECFM,C'F'            Is it fixed length?
         BE    GETR31Ø               Yes, so write it to the queue
*                                    No.. SO
*                                    Variable length, so get from RDW
         MVC   QRECLEN,Ø(R7)         RDW is first 2 bytes of record
GETR31Ø  EQU   *                    We know the length.. write it
         EXEC CICS WRITEQ TS                                          X
               NOSUSPEND                                              X
               QUEUE(SORTQNM)                                         X
               FROM(Ø(R7))                                            X
               LENGTH(QRECLEN)                                        X
               NUMITEMS(QCOUNT)                                       X
               NOHANDLE
```

```
*
         CLC   EIBRESP,DFHRESP(NORMAL) Result OK
         BE    GETR32Ø                 Yes, keep going
*
         EXEC CICS WRITEQ TD                                          X
               QUEUE(NBUG)                                            X
               FROM(SRTERR15)                                         X
               LENGTH(=AL2(L'SRTERR15))                               X
               NOHANDLE
*
         LA    R8,16                   Set return code
         ST    R8,RETCODE
         B     RETURN                  Give up with prejudice
*
GETR32Ø  EQU   *                       Data has been saved in TSQ
         LH    R9,QCOUNT               How much has been saved?
         CH    R9,FILESIZE             Have we exceeded our limit?
         BNH   GETR33Ø                 No, so keep going
         EXEC CICS WRITEQ TD                                          X
               QUEUE(NBUG)                                            X
               FROM(SRTERR16)                                         X
               LENGTH(=AL2(L'SRTERR16))                               X
               NOHANDLE
*
         LA    R8,16                   Set return code
         ST    R8,RETCODE
         B     RETURN                  Give up with prejudice
*
GETR33Ø  EQU   *                       We have not exceeded our limits
         L     R8,TABPTR               Address the keys table
         BCTR  R9,Ø                    Decrement so we calculate offsets
         MH    R9,KEYLENA2             Calculate offset
         AR    R8,R9                   R8 points to new entry in table
         MVC   Ø(2,R8),QCOUNT          Save the item number
         LH    R9,KEYOFFST             Get the key offset
         AR    R7,R9                   Point R7 to the record key
         LH    R1Ø,KEYLEN              Get the key length
         BCTR  R1Ø,Ø                   Decrement the key length for EX
         EX    R1Ø,MVCKEY              Move the key into the table
         B     GETR2ØØ                 Get the next record
*
MVCKEY   MVC   2(Ø,R8),Ø(R7)           Copy the key to the SORT table
*
GETR99Ø  EQU   *
         L     R14,PROCRET1            Restore return address
         BR    R14                     Return to caller
*
*********************************************************************
*
* This entire Quicksort routine was lifted from XEPHON CICS UPDATE.
```

```
* Only the names have been changed to protect the guilty.
*
* The code originates from the November 199Ø CICS Update, and
* is ascribed to
*
* Safran Menachem
* Systems Programmer
* Mivtachim Computers (Israel)                  c Xephon 199Ø
*
*
*
*********************************************************************
DOSORT   EQU   *                     Do the sort
         DROP  R5
         ST    R14,PROCRET1          Save the subroutine return address
*
         SR    R9,R9
         LH    R9,QCOUNT             Number of items to sort
         MH    R9,=H'2'              Amount of storage needed for stack
         ST    R9,STCKLEN
         EXEC CICS GETMAIN                                          X
               SET(R12)                                             X
               FLENGTH(STCKLEN)                                     X
               NOSUSPEND                                            X
               NOHANDLE
         CLC   EIBRESP,DFHRESP(NORMAL)
         BE    SORTØ1Ø
*
         EXEC CICS WRITEQ TD                                        X
               QUEUE(NBUG)                                          X
               FROM(SRTERR19)                                       X
               LENGTH(=AL2(L'SRTERR19))                             X
               NOHANDLE
*
         LA    R8,16                 Set return code
         ST    R8,RETCODE
         B     RETURN                Give up with prejudice
*
SORTØ1Ø  EQU   *
         ST    R12,STCKPTR           Save the stack ptr address
         MVC   LEFT,=H'1'            At the start of the sort: LEFT=1
         MVC   RIGHT,QCOUNT          and RIGHT = number of entries
         LH    R4,KEYLEN             Get length of key
         BCTR  R4,Ø                  Decrement ready for EX instruction
         LA    R14,1ØØ               Suspend every 1ØØth main loop.
*        *     *
MAINLOOP EQU   *
*                                    Don't monopolize the processor:
*                                    Give CICS a chance to reschedule
         BCT   R14,SORTØ2Ø           Decrement, if not done then skip
```

```
         EXEC CICS SUSPEND NOHANDLE
         LA    R14,100               Skip suspend 100 times
SORT020  EQU   *
         CLC   RIGHT,LEFT            RIGHT > LEFT
         BH    CHANGE                YES - GO CHANGE
         C     R12,STCKPTR           Is R12 at Stack beginning?
         BE    ENDSORT               YES - End of sort
         SH    R12,=H'2'
* POPPING THE CURRENT STACK ENTRY - TO RIGHT
         MVC   RIGHT,0(R12)
         MVC   0(2,R12),=2X'00'      'CLEANING' THE ENTRY
         SH    R12,=H'2'
* AND POPPING THE NEXT ENTRY - TO LEFT
         MVC   LEFT,0(R12)
         MVC   0(2,R12),=2X'00'      'CLEANING' THE ENTRY
         B     MAINLOOP
CHANGE   EQU   *
         SR    R5,R5
         LH    R5,RIGHT
         SH    R5,=H'1'
         MH    R5,KEYLENA2  R5 - OFFSET  OF RIGHT ENTRY
         A     R5,TABPTR    R5 - ADDRESS OF RIGHT ENTRY
         EX    R4,EX2MVC
         SR    R9,R9
         LH    R9,LEFT      R9 - CURRENT-LEFT
         SR    R8,R8
         LH    R8,RIGHT
         SH    R8,=H'1'     R8 - CURRENT-RIGHT (RIGHT-1)
         LR    R10,R9
         SH    R10,=H'1'
         MH    R10,KEYLENA2   R10 - OFFSET  OF R9 ENTRY
         A     R10,TABPTR     R10 - ADDRESS OF R9 ENTRY
         LR    R7,R8
         SH    R7,=H'1'
         MH    R7,KEYLENA2  R7 - OFFSET  OF R8 ENTRY
         A     R7,TABPTR    R7 - ADDRESS OF R8 ENTRY
*
BACKCHNG EQU   *
         CR    R9,R8            CURRENT-LEFT < CURRENT-RIGHT
         BL    CHNGLOOP          YES - GO CHANGE
         CLC   SORTDIR,=C'A'     ASCENDING SORT ?
         BE    ASCEND3
DESCND3  EQU   *
         EX    R4,EX3CLC         DESCENDING SORT
         BNH   GOON
         B     CONT3
ASCEND3  EQU   *
         EX    R4,EX3CLC
         BNL   GOON
CONT3    EQU   *
```

```
         AH    R4,=H'2'       ADDING THE ENTRY NUMBER LENGTH - 2 BYTES
         EX    R4,EX3MVC      EXCHANGING THE
         EX    R4,EX6MVC      RIGHT WITH
         EX    R4,EX7MVC      THE CURRENT-LEFT ENTRIES
         SH    R4,=H'2'       SUBTRACTING 2 FROM R4
*
GOON     EQU   *
         MVC   AIDLEFT,LEFT   SAVING THE PREVIOUS LEFT
         SR    R6,R6
         LR    R6,R9
         AH    R6,=H'1'
         STH   R6,LEFT      LEFT = CURRENT-LEFT + 1
         SH    R6,=H'2'     R2 - PREVIOUS LEFT - 1
         CH    R6,AIDLEFT
         BNH   OUTCHNG
         MVC   Ø(2,R12),AIDLEFT  PUSHING SAVED LEFT TO STACK
         AH    R12,=H'2'   ADVANCING TO NEXT STACK ENTRY
         STH   R6,Ø(R12)   AND PUSHING R2(RIGHT PART) TO STACK
         AH    R12,=H'2'   ADVANCING TO NEXT STACK ENTRY
OUTCHNG  EQU   *
         B     MAINLOOP
*
CHNGLOOP EQU   *
LOOP1    EQU   *
         CLC   SORTDIR,=C'A'    ASCENDING SORT ?
         BE    ASCEND1
DESCND1  EQU   *
         EX    R4,EX1CLC        DESCENDING SORT
         BL    LOOP2
         B     CONT1
ASCEND1  EQU   *
         EX    R4,EX1CLC
         BH    LOOP2
CONT1    EQU   *
         AH    R9,=H'1'         INCREASING R9  BY 1
         AH    R1Ø,KEYLENA2     INCREASING R1Ø BY TABENTRY LENGTH
         CH    R9,RIGHT         R9 REACHED THE CURRENT RIGHT ?
         BL    LOOP1            NO - BACK TO LOOP1
LOOP2    EQU   *
         CLC   SORTDIR,=C'A'    ASCENDING SORT ?
         BE    ASCEND2
DESCND2  EQU   *
         EX    R4,EX2CLC        DESCENDING SORT
         BH    OUTLOOP2
         B     CONT2
ASCEND2  EQU   *
         EX    R4,EX2CLC
         BL    OUTLOOP2
CONT2    EQU   *
         SH    R8,=H'1'         DECREASING R8 BY 1
```

```
         SH    R7,KEYLENA2      DECREASING R8 BY TABENTRY LENGTH
         CH    R8,LEFT          R8 REACHED THE CURRENT LEFT ?
         BH    LOOP2            NO - BACK TO LOOP2
*
OUTLOOP2 EQU   *
         CR    R9,R8            CURRENT-LEFT < CURRENT-RIGHT
         BNL   BACKCHNG         NO - DO NOT EXCHANGE ENTRIES
         AH    R4,=H'2'      ADDING THE ENTRY NUMBER LENGTH - 2 BYTES
         EX    R4,EX3MVC     EXCHANGING THE
         EX    R4,EX4MVC     CURRENT-LEFT
         EX    R4,EX5MVC     WITH CURRENT-RIGHT ENTRIES
         SH    R4,=H'2'      SUBTRACTING 2 FROM R4
         B     BACKCHNG
*
ENDSORT  EQU   *
         L     R5,PARMLIST           Address the sort parm list
         USING SORTPARM,R5           Map the sort parameters
         L     R14,PROCRET1
         BR    R14                   Return to caller
*
EX1CLC   CLC   2(Ø,R1Ø),CURVALUE  COMPARING CURRENT-LEFT  WITH CURVALUE
EX2CLC   CLC   2(Ø,R7),CURVALUE   COMPARING CURRENT-RIGHT WITH CURVALUE
EX3CLC   CLC   2(Ø,R5),2(R1Ø)     COMPARING RIGHT WITH CURRENT-LEFT
EX2MVC   MVC   CURVALUE(Ø),2(R5)   MOVING THE RIGHT KEY TO CURVALUE
EX3MVC   MVC   TEMP(Ø),Ø(R1Ø)      MOVING CURRENT-LEFT  TO TEMP
EX4MVC   MVC   Ø(Ø,R1Ø),Ø(R7)      MOVING CURRENT-RIGHT TO CURRENT-LEFT
EX5MVC   MVC   Ø(Ø,R7),TEMP        MOVING TEMP  TO CURRENT-RIGHT
EX6MVC   MVC   Ø(Ø,R1Ø),Ø(R5)      MOVING RIGHT TO CURRENT-LEFT
EX7MVC   MVC   Ø(Ø,R5),TEMP        MOVING TEMP  TO RIGHT
*
*********************************************************************
PUTRECS  EQU   *                     Give the records back to E35
         ST    R14,PROCRET1          Save the subroutine return address
*        *     *
         L     R7,TABPTR             Get address of table
         LH    R6,QCOUNT             Number of records to export
         LTR   R6,R6                 Any records at all?
         BZ    PUTR99Ø               No, so stop
*                                    Build the passarea for E35
         MVC   E35UEAC,E15UEAC       Set up address constant
         SR    R8,R8                 Create a zero
         ST    R8,E35NEWA            Clear value
         ST    R8,E35OUTA            No output area (no output file)
*
         LA    R12,1ØØ               Suspend every 1ØØth time
*
PUTR1ØØ  EQU   *
         SR    R1Ø,R1Ø               Make a zero
         ST    R1Ø,E35NEWA           No record until we get it
         LH    R1Ø,LRECL
```

```
         STH   R1Ø,QRECLEN
         EXEC CICS READQ TS                                          X
               QUEUE(SORTQNM)                                        X
               ITEM(Ø(R7))                                           X
               SET(R1Ø)                                              X
               LENGTH(QRECLEN)                                       X
               NOHANDLE
         ST    R1Ø,E35NEWA          Put it in the parm list
PUTR11Ø  EQU   *
*                                   Give CICS a chance to reschedule
         BCT   R12,PUTR12Ø          Don't suspend unless 1ØØ done
         EXEC CICS SUSPEND NOHANDLE
         LA    R12,1ØØ              Skip another 1ØØ loops
PUTR12Ø  EQU   *
         L     R15,SPE35A           Address of the output exit
         LA    R1,E35PARM           Pointer to parm list
         BALR  R14,R15              Call the E35 exit for rec output
         C     R15,=F'Ø'            Not supported - Can't write file
         BE    PUTR5ØØ              Just ignore it
         C     R15,=F'4'            Delete the record. Branch back
         BE    PUTR5ØØ              Just ignore it
         C     R15,=F'8'            End of processing. All done
         BE    PUTR99Ø              Just ignore it
         C     R15,=F'12'           Insert - not supported
         BE    PUTR5ØØ              Just ignore it
*        If we get here, we abandon the sort
         EXEC CICS WRITEQ TD                                         X
               QUEUE(NBUG)                                           X
               FROM(SRTERR17)                                        X
               LENGTH(=AL2(L'SRTERR17))                              X
               NOHANDLE
*
         LA    R8,16                Set return code
         ST    R8,RETCODE
         B     RETURN               Give up with prejudice
*
PUTR5ØØ  EQU   *                    Bottom of the loop
         SR    R8,R8                Create a zero
         ST    R8,E35NEWA           No records left (just in case)
         LTR   R6,R6                More records left?
         BZ    PUTR11Ø              No, but keep calling till RC=8
         AH    R7,KEYLENA2          Skip the table entry
         BCT   R6,PUTR1ØØ           Go get the next one.
         B     PUTR11Ø              Keep going until RC=8
*
PUTR99Ø  EQU   *
         L     R14,PROCRET1         Restore return address
         BR    R14                  Return to caller
*
*********************************************************************
```

```
RELSTG   EQU   *                    Release storage use by sort
         ST    R14,PROCRET1         Save the subroutine return address
*        *     *
         EXEC CICS DELETEQ TS                                          X
               QUEUE(SORTQNM)                                          X
               NOHANDLE
*
         L     R1Ø,TABPTR
         EXEC CICS FREEMAIN                                            X
               DATAPOINTER(R1Ø)                                        X
               NOHANDLE
*
         L     R1Ø,STCKPTR
         EXEC CICS FREEMAIN                                            X
               DATAPOINTER(R1Ø)                                        X
               NOHANDLE
*
         L     R14,PROCRET1         Restore return address
         BR    R14                  Return to caller
*
*
SRTERRØØ DC    C'NABSORTØØE - Sort extended parameter list missing'
SRTERRØ1 DC    C'NABSORTØ1E - Sort Control Statements are required'
SRTERRØ2 DC    C'NABSORTØ2E - Sort E15 exit is required'
SRTERRØ3 DC    C'NABSORTØ3E - Sort E35 exit is required'
SRTERRØ4 DC    C'NABSORTØ4E - Alternate sequence table not supported'
SRTERRØ5 DC    C'NABSORTØ5E - Sort E18 exit is not supported'
SRTERRØ6 DC    C'NABSORTØ6E - Sort E39 exit is not supported'
SRTERRØ7 DC    C'NABSORTØ7E - Sort parm list end indicator missing'
SRTERRØ8 DC    C'NABSORTØ8E - Sort control statement not supported'
SRTERRØ9 DC    C'NABSORTØ9E - Too many records to sort. Max 5ØØØ'
SRTERR1Ø DC    C'NABSORT1ØE - Key length is larger than 25Ø bytes'
SRTERR11 DC    C'NABSORT11E - Key occurs outside record'
SRTERR12 DC    C'NABSORT12E - Record length is too large (>2Ø48)'
SRTERR13 DC    C'NABSORT13E - Could not acquire key table storage'
SRTERR14 DC    C'NABSORT14E - Sort abandoned by E15 exit RC=16'
SRTERR15 DC    C'NABSORT15E - Could not fit data in SORT TS queue'
SRTERR16 DC    C'NABSORT16E - More than FILSZ records passed to SORT'
SRTERR17 DC    C'NABSORT17E - Sort abandoned by E35 exit RC=16'
SRTERR18 DC    C'NABSORT18E - Transaction: xxxx Terminal: xxxx'
SRTERR19 DC    C'NABSORT19E - Could not acquire stack storage'
CSMT     DC    C'CSMT'
NBUG     DC    C'NBUG'
EXITPT   DFHEIRET
         END
```

SORT JCL

```
//NWCASSEM JOB ACCT-CLASS,'COMPILE PGM',
```

```
//             MSGCLASS=X,
//*            TYPRUN=SCAN,
//             CLASS=M,REGION=4M
//***
//***  ASSEMBLE PROGRAMS
//***    SORT
//***
//TRN    EXEC PGM=DFHEAP1$,
//            REGION=2M,
//            PARM='SP,NOPROLOG'
//STEPLIB  DD DSN=SYS1.SDFHLOAD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSPUNCH DD DSN=&&SYSCIN,
//            DISP=(,PASS),UNIT=SYSDA,
//            DCB=BLKSIZE=400,
//            SPACE=(400,(400,100))
//*
//SYSIN   DD DISP=SHR,DSN=NWC.SOURCE.ASM(SORT)
//*
//*********************************************************************
//*
//ASM    EXEC PGM=ASMA90,
//            REGION=2M,
//            PARM='DECK,NOOBJECT,LIST'
//SYSLIB   DD DSN=NWC.SOURCE.ASM,DISP=SHR
//         DD DSN=SYS1.SDFHMAC,DISP=SHR
//         DD DSN=SYS1.SDFHSAMP,DISP=SHR
//         DD DSN=SYS1.MACLIB,DISP=SHR
//         DD DSN=SYS1.MODGEN,DISP=SHR
//SYSUT1   DD UNIT=SYSDA,SPACE=(1700,(400,400))
//SYSUT2   DD UNIT=SYSDA,SPACE=(1700,(400,400))
//SYSUT3   DD UNIT=SYSDA,SPACE=(1700,(400,400))
//SYSPUNCH DD DSN=&&LOADSET,
//            UNIT=SYSDA,DISP=(,PASS),
//            SPACE=(400,(100,100))
//SYSPRINT DD SYSOUT=*
//SYSIN    DD DSN=&&SYSCIN,DISP=(OLD,DELETE)
//*
//*********************************************************************
//*
//COPYLINK EXEC PGM=IEBGENER,COND=(7,LT,ASM)
//SYSUT1   DD DSN=SYS1.SDFHMAC(DFHEILIA),DISP=SHR
//SYSUT2   DD DSN=&&COPYLINK,DISP=(NEW,PASS),
//            DCB=(LRECL=80,BLKSIZE=400,RECFM=FB),
//            UNIT=SYSDA,SPACE=(400,(20,20))
//SYSPRINT DD SYSOUT=*
//SYSIN    DD DUMMY
//*
//*********************************************************************
//*
```

```
//LKED   EXEC PGM=IEWL,REGION=2M,
//            PARM='LIST,XREF',COND=(7,LT,ASM)
//SYSLIB   DD DSN=SYS1.SDFHLOAD,DISP=SHR
//SYSLMOD  DD DSN=NWC.LOADLIB,DISP=SHR
//SYSUT1   DD UNIT=SYSDA,DCB=BLKSIZE=1024,
//            SPACE=(1024,(200,20))
//SYSPRINT DD SYSOUT=*
//SYSLIN   DD DSN=&&LOADSET,DISP=(OLD,DELETE)
//         DD DSN=&&COPYLINK,DISP=(OLD,DELETE)
//         DD DDNAME=SYSIN
//SYSIN    DD *
           ENTRY SORT
           MODE AMODE(31),RMODE(ANY)
           NAME SORT(R)
/*
```

Neil Casey
Senior Systems Programmer (CICS and MQSeries)
National Australia Bank (Australia)

CICS news

IBM has announced WebSphere Application Server V4.0.1 for z/OS and OS/390, with J2EE-compliant deployment platform for servlets, JavaServer Pages, and Enterprise JavaBeans, plus Java Messaging Service and Web services.

The deployment environment is delivered as Web services with SOAP and UDDI support. It continues to support connector access to CICS, IMS, and DB2 as well as connections to applications that run on Web servers and back-end systems such as DB2, CICS, and IMS.

Included with the application server is Software Development Kit 1.3, providing the base support for applications at the Java 2 API level.

For further information contact your local IBM representative.
URL: http://www.ibm.com/software/webservers.

* * *

H&W Computer Systems has announced WebTek Tools for CICS, which is a suite of tools for developing, deploying, maintaining, and securing CICS Web applications.

The product addresses the infrastructure requirements of extending CICS applications to the Web, including security, file management, limitations of a 32KB COMMAREA, and restricted concurrent connections.

WebTek also provides flexibility in application development and maintenance, including separate corporate presentation Web elements from CICS application logic, merging data from other sources in a CICS Web page, and deploying CICS as a single server solution.

For further information contact:
H&W Computer Systems, 6154 N Meeker Place, Suite 100, Boise, ID 83713-1533, USA.
Tel: (208) 377 0336.
URL: http://www.hwcs.com/html/cicsweb.html.

* * *

IBM has begun a promotion by which users get a free one-day CICS Migration Consultancy, worth about US$2,800, when they upgrade to CICS TS for OS/390 V1.3. The deal is sweetened if you upgrade and also purchase WebSphere V4.0 for any platform: you get the migration consultancy plus CICS Transaction Gateway V4 for the same platform as the WebSphere licence.

For further information contact your local IBM representative.
URL: http://www.software.ibm.com.

* * *

Xephon is holding a one-day conference entitled *CICS Update 2001* at the Radisson SAS Portman hotel, London, on 11 December 2001.

For further details about *CICS Update 2001*, or information about Xephon's complete range of seminars and conferences, call 01635 33823 or browse www.xephon.com/events.

* * *