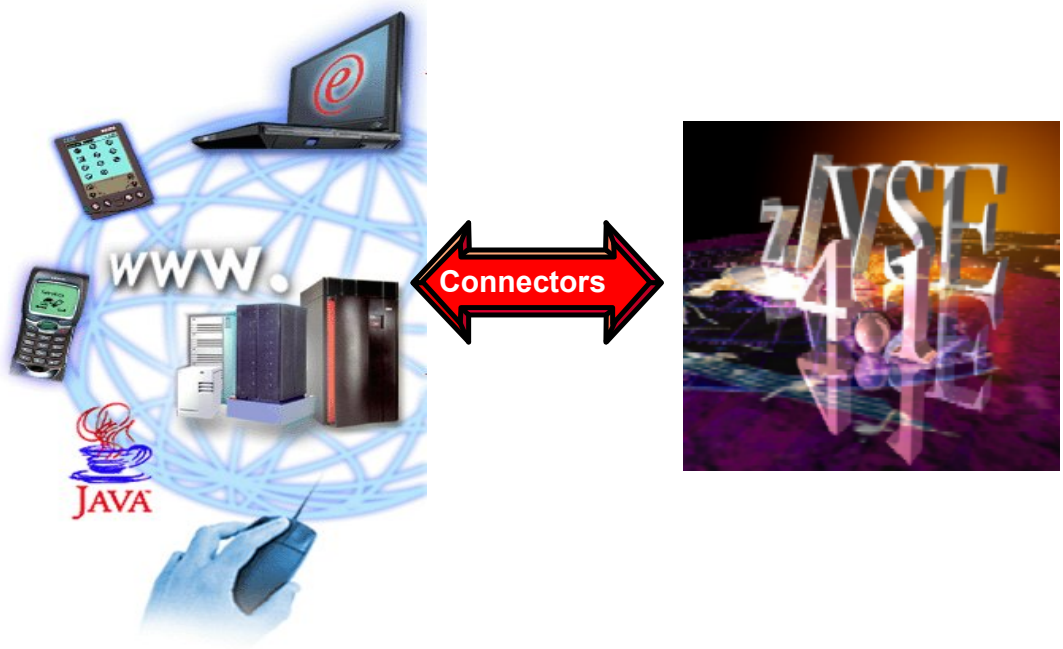
VSE Connectors Workshop

Setup of
Connections to VSE CICS TS
from Windows



Wilhelm Mild, IBM
Ingo Franzki, IBM

Contact: zVSE@de.ibm.com

Trademarks

References in this publication to IBM products or services do not imply that IBM intends to make them available in every country in which IBM operates. Consult your local IBM business contact for information on the products, features, and services available in your area.

AIX*, APPN*, CICS*, CICS/VSE*, CICS, DB2*, DB2 Connect, DB2 Universal Database, DFSORT, DRDA*, e-business logo*, Enterprise Storage Server,FlashCopy, HiperSockets, IBM*, IBM logo*, IBM eServer, iSeries, Language Environment*, MQSeries*, Multiprise*, pSeries, S/390*, S/390, System z9Parallel Enterprise Server, TotalStorage, VSE/ESA, z/VSE, VTAM*, WebSphere*, xSeries, z/OS, z/VM, zSeries and Distributed Relational Database Architecture are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

Linux is a registered trademark of Linus Torvalds.
Java and all Java-related trademarks and logos are trademarks or registered trademark of Sun Microsystems, Inc.

UNIX is a registered trademark in the United States and other countries, licensed exclusively through The Open Group.

Microsoft, Windows, Windows NT, Visual Basic and the Windows flat logo are Trademarks of Microsoft Corporation.

Other trademarks and registered trademarks are the properties of their respective companies.

IBM hardware products are manufactured from new parts, or new and used parts. Regardless, our warranty terms apply. This equipment is subject to all applicable FCC rules and will comply with them upon delivery.Information concerning non-IBM products was obtained from the suppliers of those products. Questions concerning those products should be directed to those suppliers.

All statements regarding IBM’s future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Workshop objectives

Implementation of different ways for CICS access:

1. Access the sample CICS application FFST via 3270 terminal:

The first part of the workshop introduces the sample CICS application FFST. We will use the sample transaction in the traditional way using 3270 terminal. In the following chapters, we will modernize the access to this application in 3 different ways.

2. Access to CICS applications via Web Browser:

This part implements access to the sample application via CICS Web Support and the 3270 Bridge using a web browser.

3. Integration of CICS applications in distributed processes:

This part implements a solution with CICS Transaction Gateway. This time, the sample application is invoked from a distributed system.

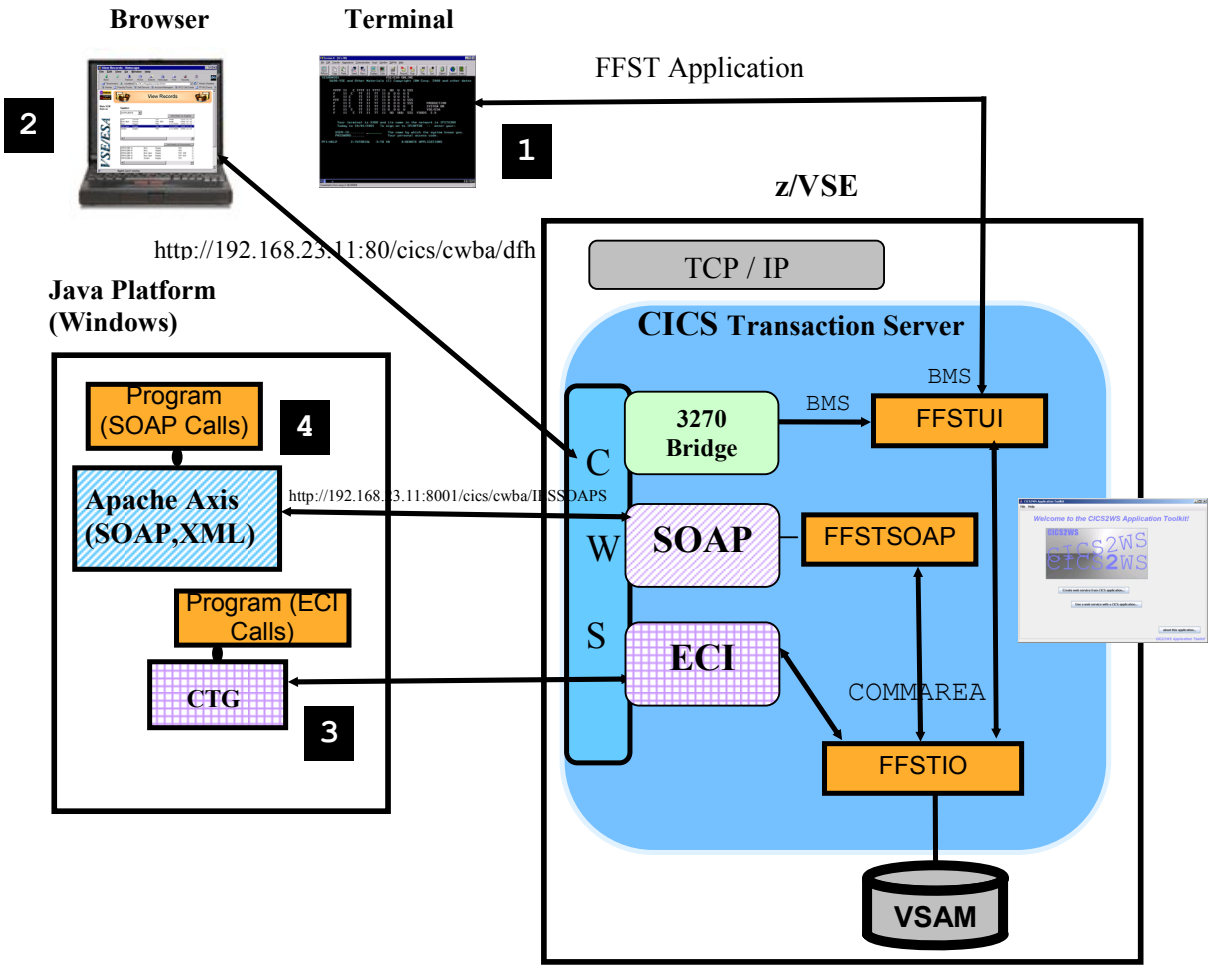
4. Access a CICS application as Web service:

This part implements VSE Web services using SOAP and XML. We will service enable the sample application and invoke it from a distributed system as a web service.

Structure of the sample application FFST

The following picture shows the structure of the sample application FFST that we are using in the workshop.

The workshop will guide you through the implementation of the scenarios 1 to 4.



Chapter 1.) Software prerequisites for Windows

Step 1.1: Verification if Java environment installed

To install the VSE Connector Client, a Java Virtual Machine (JVM) must be installed in Windows.

The JVM can be installed in different flavors:

- To just run Java programs, the JRE 1.4.x or later is needed (Java Runtime Environment)
- To develop/compile Java programs, JDK 1.4.x or higher is needed (Java Developer Kit, which includes the JRE).

To verify if a Java Virtual Machine is installed, open a Command prompt and enter the following command:

```
java -version
```

You should see something like:

```
Java version "1.4.2"
Java(TM) 2 Runtime Environment, Standard Edition
```

If you see messages like above, please continue with Chapter 2.).

Step 1.2: Install a Java Environment

If following message (or similar) is shown:

```
java' is not recognized as an internal or external command,
operable program or batch file.
```

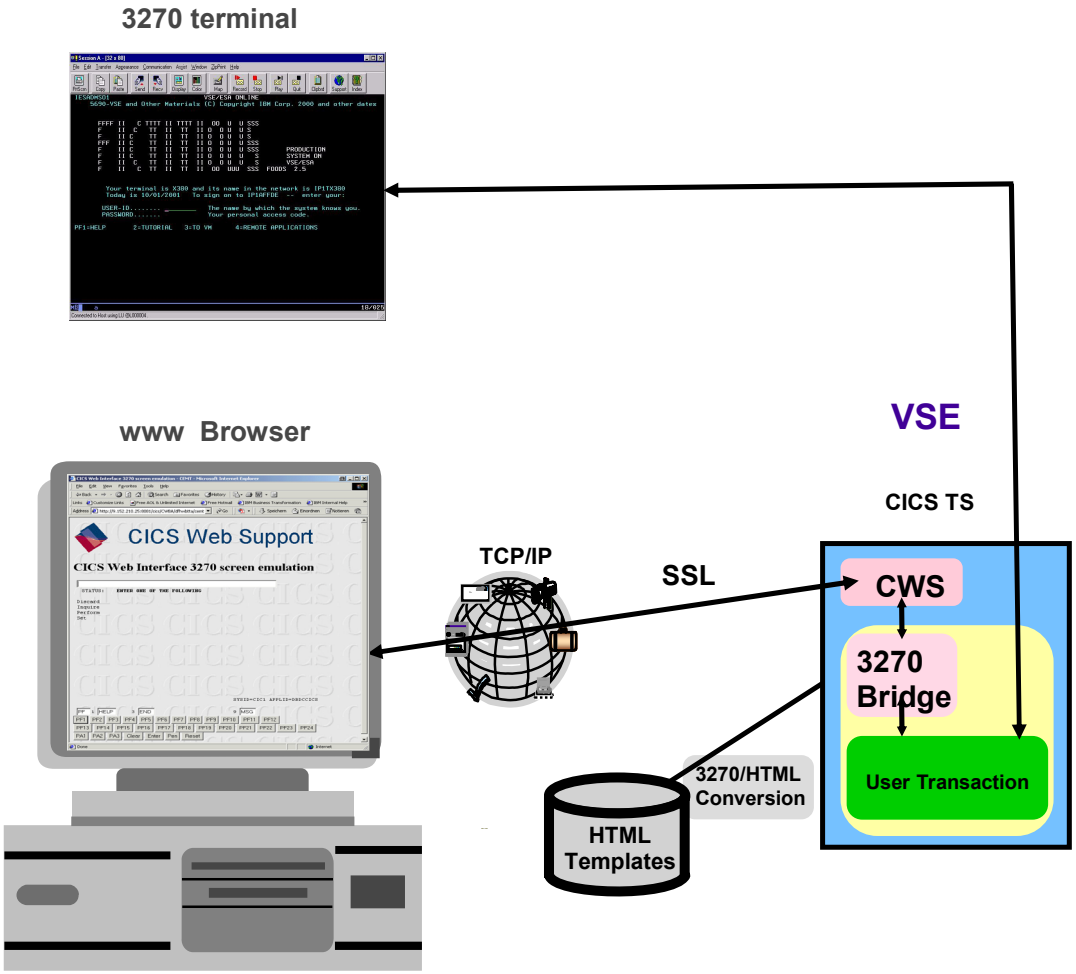
then your system has no Java virtual machine (Runtime Environment) installed or it can not be found in the path.

To install a Java Virtual Machine download the code from IBM: http://www.ibm.com/developerworks/java/ or download a SUN Version from http://www.sun.com.

After downloading, you need to install the downloaded JDK 1.4.x. or later.

Chapter 2.) Setup CICS Web support

Access to z/VSE transactions via terminal and browser:



Goal of this chapter:

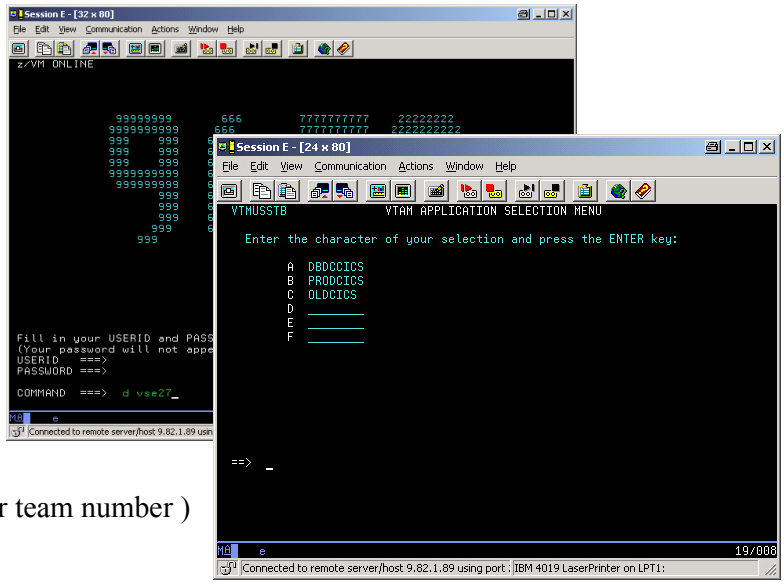
- Traditional way to access a CICS applications via 3270 terminal
- Direct access to z/VSE applications via web Browser
- Without the need of a web server on VSE

Software requirements:

- z/VSE 3.1 and newer
- CICS Transaction Server
- TCP/IP for VSE

Step 2.1: Access FFST sample application via Terminal

Logon to your VSE system using the 3270 icon on your desktop:



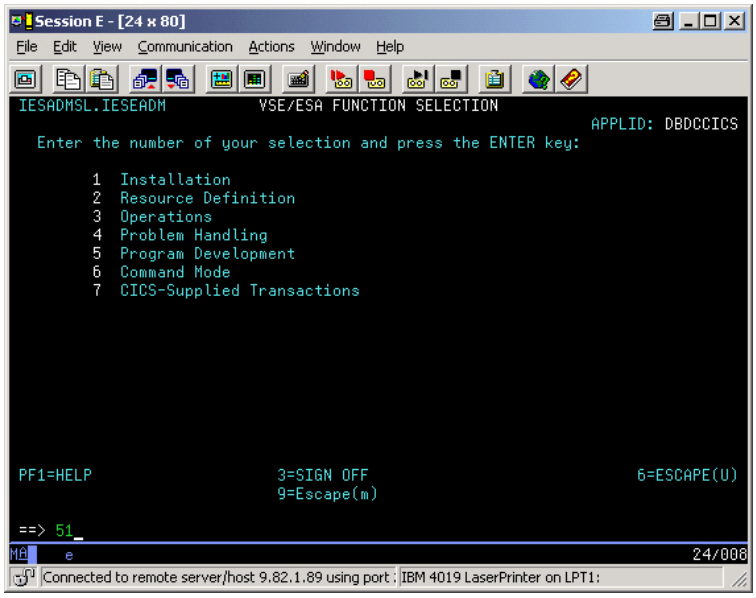
On Command line enter:

D VSE410 (on z/VM only)
A

On the logon screen enter:

User: **TExx** (xx- is your team number)
Password: **TEAMxx**

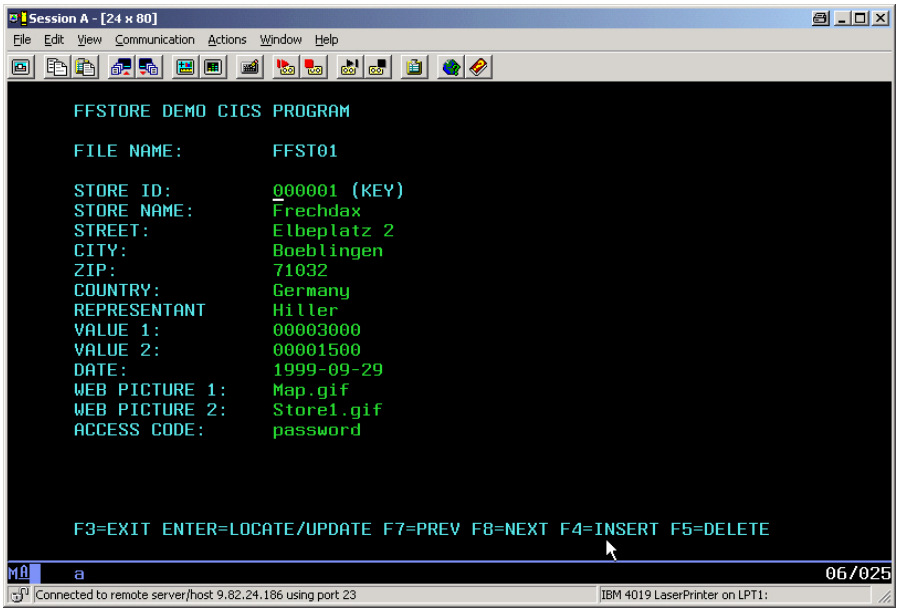
You are now in the Interactive Interface main panel of VSE.



To start the sample application we are using the transaction FFST:

Hit: **PF9** (to go into CICS mode – mixed case)
Enter: **FFST FFSTxx** (where FFSTxx is the VSAM file for team xx)

Now you have traditional access to VSAM data via a 3270 terminal emulation.



The sample application allows you to browse through a VSAM file. Every team uses its own VSAM file named FFSTxx, where xx is the team number.

Now, play around with this application, to get used to it. We will use the same application later on in this workshop.

- Press **PF8** to display the next record, or **PF7** to display the previous record.
- To display a specific store, enter the store id and press **enter**.
- To insert a new record, enter the data and press **PF4**.
- To delete a record, navigate to the record you want to delete and press **PF5**.
- To update a record, navigate to the record you want to update, enter the data you want to update and press **enter**.
- To leave the application and return to the CICS screen, press **PF3**. To return to the Interactive Interface selection panel, press PF3 once more.

Step 2.2: Setup CICS Web Support

In order to access a CICS application via a web browser, you need to setup CICS Web Support. CICS Web Support is a function of CICS Transaction Server in VSE. It allows to access CICS applications via a TCP/IP service that has to be defined in CICS. A TCP/IP service is a listener in CICS. It can communicate via HTTP with a web browser. For each separate CICS region in your system, CICS Web Support can be enabled.

The following steps have to be done in order to enable CICS Web Support. We have **already done these steps** for the VSE system we are using in this workshop:

- The following parameters have been changed in DFHSITSP for the CICS TS region DBDCCICS to enable CWS:
    - **ISC=YES** Intersystem communication enabled
    - **TCPIP=YES** TCP/IP protocol enabled
- Configure and enable codepage conversion in CICS. The IBM provided skeleton **DFHCNV** (ICCF library 59) has been adapted and submitted.
- Configure and enable the CICS Web Error program. The IBM provided skeleton **DFHWBEP** (ICCF library 59) has been adapted and submitted.
- The BMS map for application FFST was compiled with option **`SYSPARM='TEMPLATE'`**. This produces a HTML template that was adapted and stored in PRD2.DFHDOC.
- The CICS startup job has been adapted to include the library PRD2.DFHDOC in the **LIBDEF**. This is required to allow CICS to find the HTM templates.

More details of these changes are described in Appendix A on page 28.

As already mentioned, these configuration steps have already been done by us prior to the workshop. You do not need to perform any of these steps now, but you will have to perform them on your VSE system in your shop to enable CICS Web Support.

The next steps will guide you to enable CWS and browser access to the CICS application FFST. This includes the following steps:

- Define and install a TCP/ IP service
- Verify if the TCP/IP service is open
- Access the FFST application via a web browser

Step 2.3: Define a TCP/IP service for CWS

From the interactive interface main panel (as described in Step 2.1:)

Hit: **PF6** (to go into the CICS mode)
Enter: **CEDA DEF TCPIPS(CWSxx)** were xx is your team number.

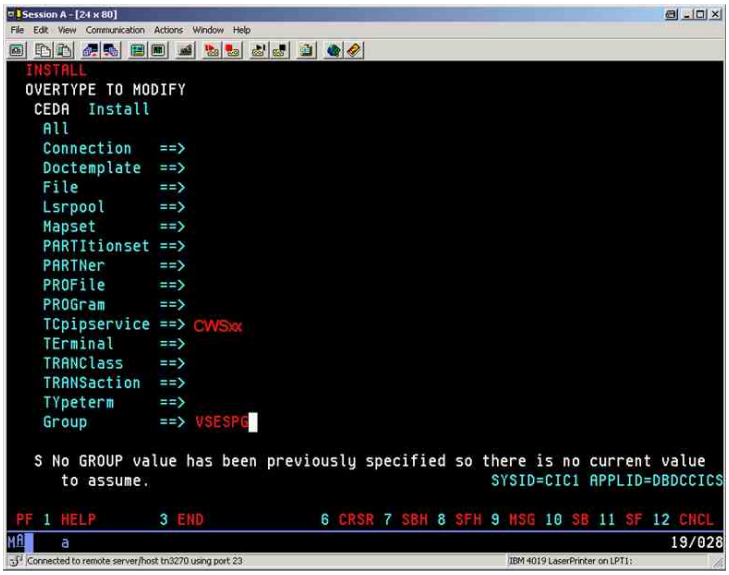
Fill out the remaining parameters as shown below and press enter:

```
CEDA  DEFine TCpipservice( CWSxx      )
  TCpipservice   : CWSxx
  Group          : VSESPG
  Description  ==> SERVICE FOR CWS
  Urm          ==> DFHWBADX
  Portnumber   ==> 80xx                1-65535
  Certificate  ==>
  STatus       ==> Open                Open | Closed
  SSl          ==> NO                  Yes | No | Clientauth
  Attachsec    ==> Local               Local | Verify
  TRansaction  ==> CWXN
  Backlog      ==> 00009               0-32767
  TSqprefix    ==>
  Ipaddress    ==>
  SOcketclose  ==> No                  No | 0-240000
```

Step 2.4: Install the TCP/IP service in a CICS group

To activate the TCP/IP service you need to install it.
From a CICS Command (see Step 2.1:) enter

CEDA INSTALL TCPIPS(CWSxx) in the group you specified in the definition:



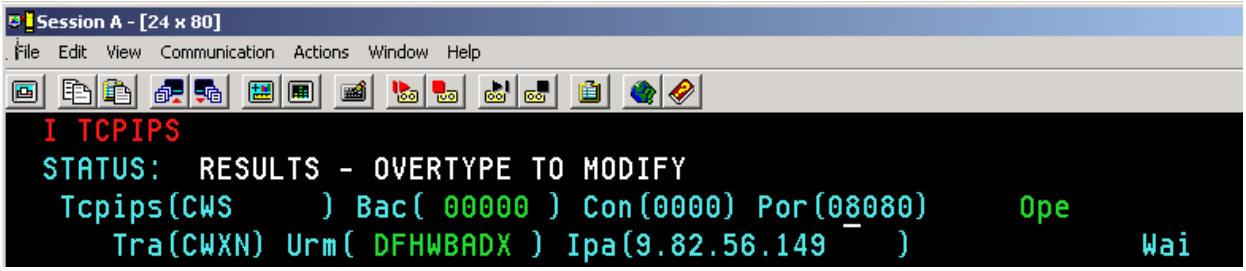
Step 2.5: Invoke the FFST application from Browser

Verify if the TCP/IP service in CICS is open:

In the Interactive Interface main panel as described in Step 2.1:):

Hit **PF6** and enter
CEMT I TCPIPS

Look for your CWSxx name. You should see something like:



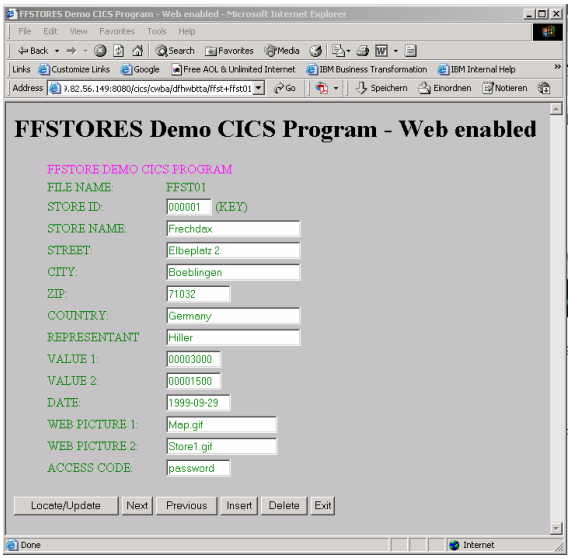
If the status is **CLO**sed, open it by overtyping it with **OPE**n and pressing enter.

You can now access the sample application FFST from a web browser. We need to tell the application which file to use. The application uses accepts the VSAM file name as parameter when invoked. Please use the VSAM file called FFSTxx (where xx is your team number).

Open a web browser (e.g. Internet Explorer) and type the following URL (where xx is your team number). Please note that the IP address may be different due to the different network setups:

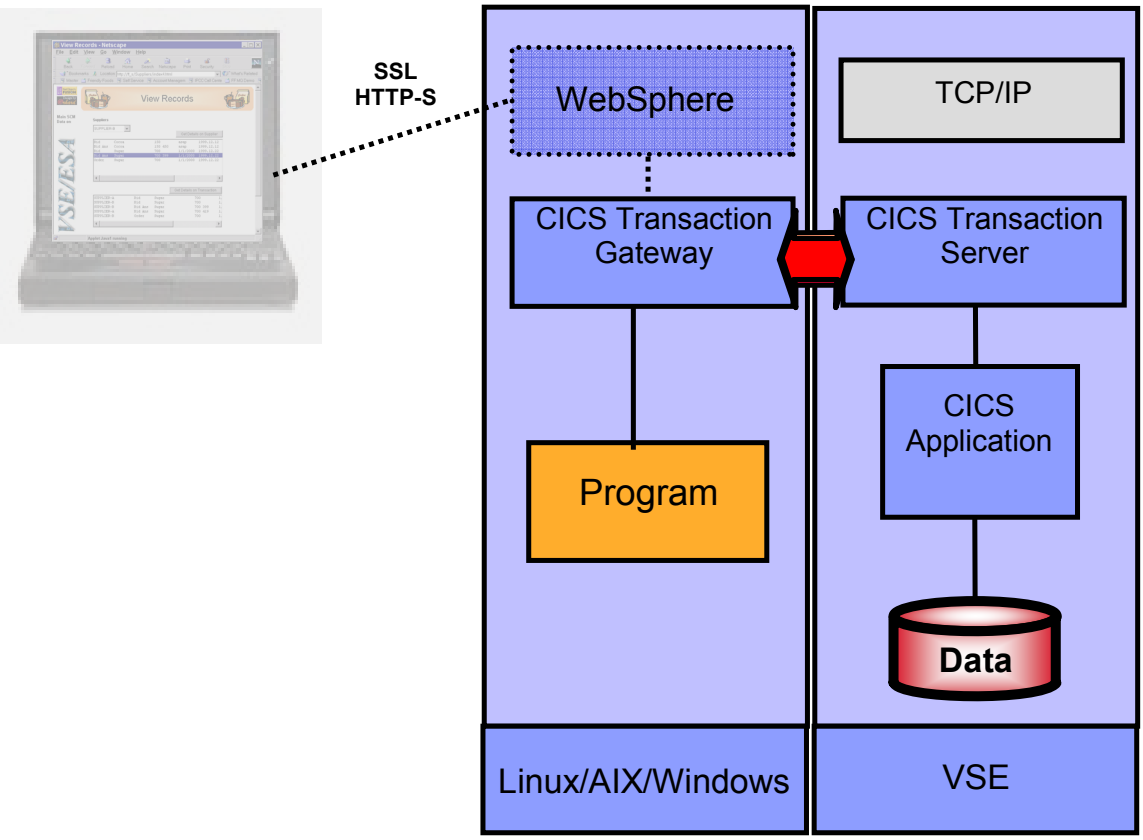
http://192.168.23.11:80xx/cics/cwba/dfhwbtta/FFST+FFSTxx

The application should now show up in your web browser. You can work with the application as done in step Step 2.1: Use the buttons instead of the PF keys.



Chapter 3.) Setup CICS Transaction Gateway

This chapter shows how to integrate CICS applications in distributed environments using IBM CICS Transaction Gateway:



The integration of CICS business logic in distributed transaction processes allows you to

- Invoke a CICS program remotely
- Maintain the transaction security on the remote side

The following steps will guide you through the setup and configuration of CICS Transaction Gateway. CICS Transaction Gateway is a priced IBM product.

Step 3.1: Installation and setup of CICS Transaction Gateway

CICS Transaction Gateway (CTG) is the remote component necessary to communicate with CICS TS on VSE. In order to use CTG you must install it on a server or on your workstation.

The installation of CTG was already done on your workstation prior to this workshop. t was installed into the default directory:

```
C:\Program Files\IBM\CICS Transaction Gateway
```

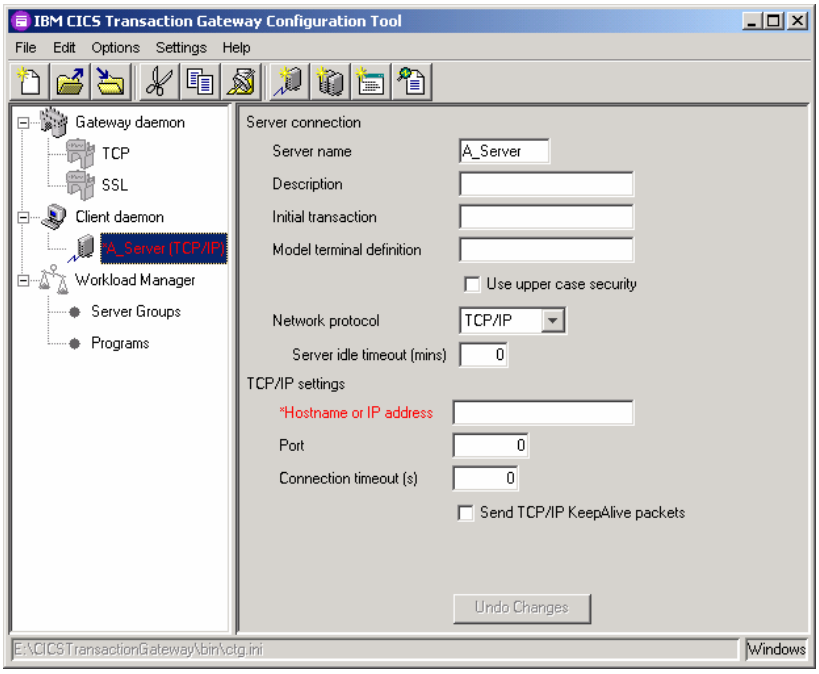
Step 3.2: Configure CICS Transaction Gateway on Windows

Before you can use CICS Transaction Gateway, you need to configure it. You do so using the "Configuration Tool". You find it in the Windows start menu:

START → Programs → IBM CICS Transaction Gateway → Configuration Tool

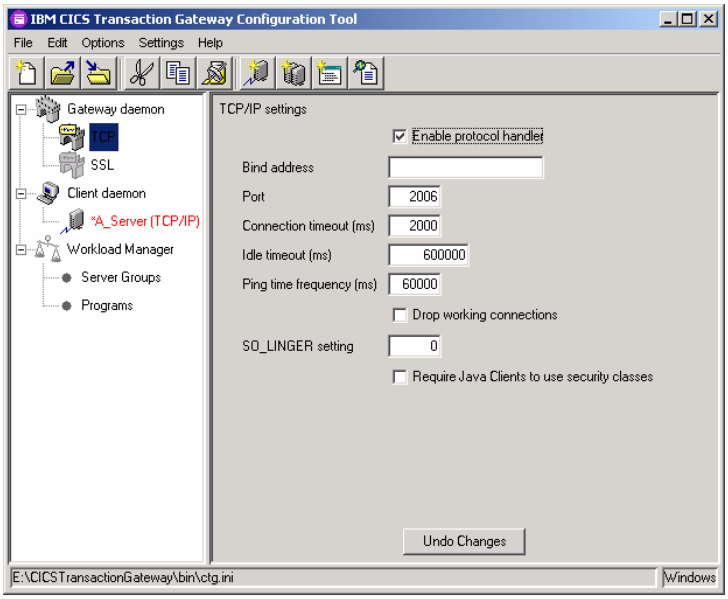
In case it asks if you wish to use the Task Guide, press "No".

The CTG Configuration Tool looks as follows:



Step 3.3: Enable the TCP protocol

Select “TCP” on the Gateway daemon node. Click on “Enable protocol handler” and verify the remaining parameters.

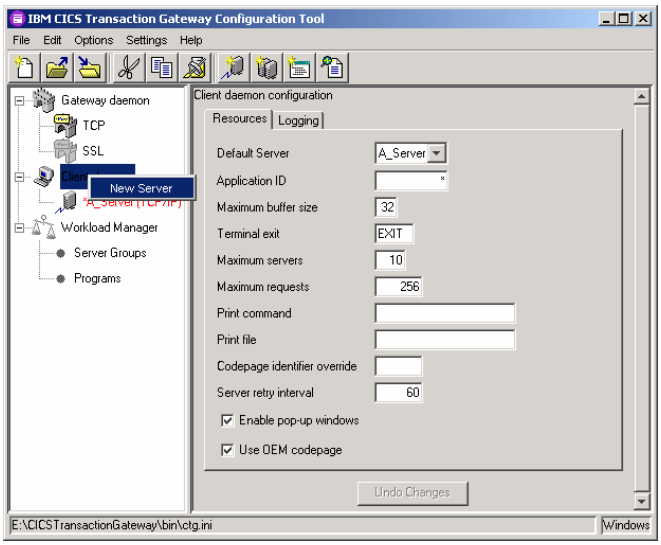


The “Port“ should be 2006.

Step 3.4: Define a new Server for the VSE system

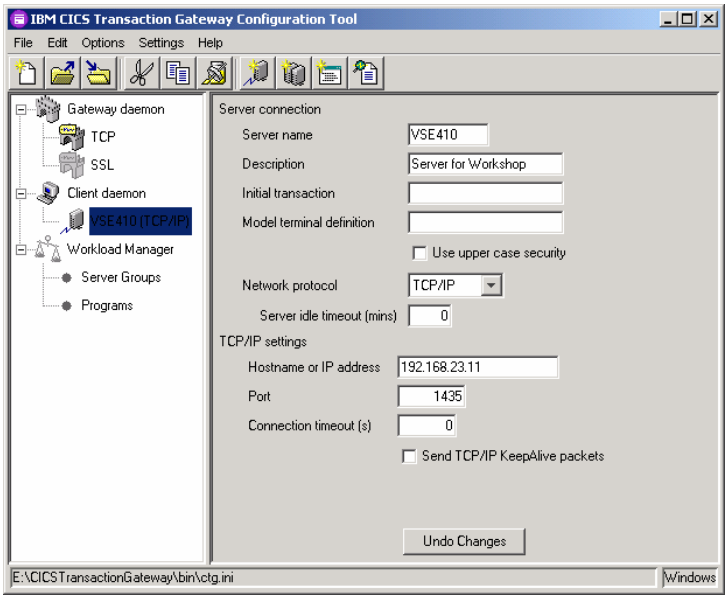
CICS Transaction Gateway can communicate with multiple backend systems (z/VSE or z/OS) simultaneously. Every backend system is represented by a Server definition in CTG.

To define a new server for the VSE system, right click on the “Client daemon” node and select “New Server” or use the existing template “A_Server”:



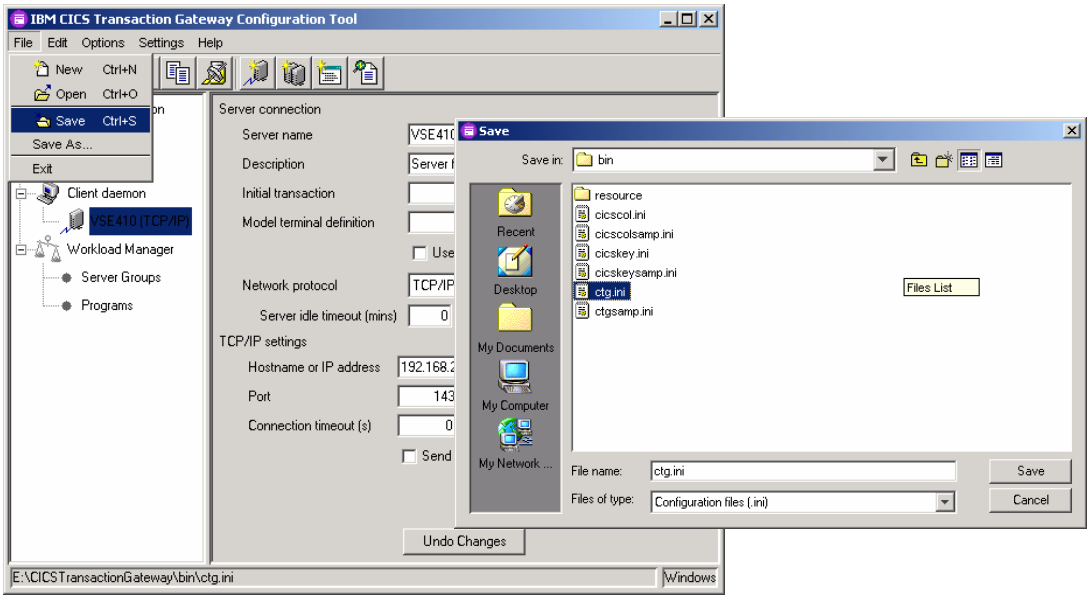
Enter the following parameters as shown:

Server Name: **VSE410**
Network protocol: **TCP/IP**
Hostname or IP address: **192.168.23.11** (Note: the IP address may be different)
Port: **1435**



Step 3.5: Save the configuration

To save the configuration, click on "**File -> Save**". Leave the default file name "**ctg.ini**" and default location and press "Save".



The CTG is now fully configured. You can close the Configuration Tool (File -> Exit).

Step 3.6: Start CTG and CICS Client

CICS Transaction Gateway is build of the CICS Client and the CICS Transaction Gateway (CTG) itself. CTG uses the CICS client to communicate with VSE. Each of these components runs in a separate process on Windows.

Starting CTG will automatically start CICS client:

START -> Programs -> IBM CICS Transaction Gateway -> IBM CICS Transaction Gateway (console mode)

You will see some messages like these:

```
CICS Transaction Gateway, Version 7.0 00. Build Level c700-20061113.
(C) Copyright IBM Corporation 1996, 2006.  All rights reserved.
08/20/07 10:34:11:802 [0] CTG6400I CICS Transaction Gateway is starting
08/20/07 10:34:11:862 [0] CTG8400I Using configuration file
E:\CICSTransactionGateway\bin\ctg.ini.
08/20/07 10:34:11:862 [0] CTG6577I Java version is 1.5.0
08/20/07 10:34:11:872 [0] CTG6502I Initial ConnectionManagers = 1, Maximum ConnectionManagers
= 100
08/20/07 10:34:11:872 [0] CTG6526I Initial Workers = 1, Maximum Workers = 100
08/20/07 10:34:11:882 [0] CTG6547W Gateway daemon will display symbolic TCP/IP hostnames in
messages
08/20/07 10:34:12:553 [0] CTG6981I Successfully initialized JNI library
08/20/07 10:34:12:603 [0] CTG6505I Successfully created the initial ConnectionManager and
Worker threads.
08/20/07 10:34:12:944 [0] CTG6524I Successfully started handler for the tcp: protocol on port
2006
08/20/07 10:34:13:024 [1] CTG6524I Successfully started handler for the localadmin: protocol
on port 2810
08/20/07 10:34:13:024 [0] CTG6597I The statsapi handler has not been started.
08/20/07 10:34:13:444 [0] CTG6512I CICS Transaction Gateway initialization complete
CTG6508I To shut down the Gateway daemon type
CTG6493I    Q or - for normal shutdown
CTG6494I    I for immediate shutdown
```

Note: CTG can also run as a Service under Windows.

Whenever you change the configuration (ctg.ini) you have to recycle (stopped / started) CTG.

Note: Stopping CTG will NOT stop CICS client.

- To stop CTG enter Q in the command prompt where CTG is running.
- To stop the client use command:
  `"C:\Program Files\IBM\CICS Transaction Gateway\bin\cicscli" -X`

Please leave the CTG running, since we need them later on. If you have stopped it, you need to restart it now.

Step 3.7: Setup VSE for access via CICS Transaction Gateway

External CICS calls from CICS Transaction Gateway (ECI calls) use the CICS Web Support interface of CICS TS. Therefore CICS Web Support has to be setup (please see Step 2.2:) and a TCP/IP service has to be defined for use with CTG.

For the workshop, following TCP/IP service was already defined prior to the workshop. Every team will use the same predefined TCP/IP service this time.

TCP/IP Service: **ECI**
Port: **1435** (This port has to be specified in the CTG Server definition)

Details for this definition can be found in Appendix D on page 31.

Step 3.8: Execute CTG sample program (ECI access)

For the workshop, we use a sample program. It was prepared already prior to the workshop and was copied to your workstation in directory "**`C:\ctg`**"

In a Windows command prompt change to C:\ctg and edit the batch script **runeci.bat** and adapt it for your team:

Enter: **C:**
cd C:\ctg
Notepad runeci.bat (make the changes described below and save them)

Adapt these values (where xx is your team number):

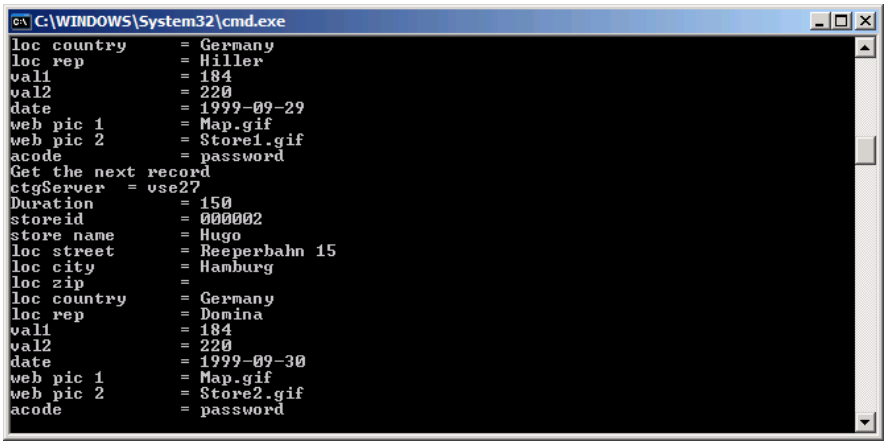
VSE Server name in CTG: **VSE410**
Host for Gateway: local:
Port: **2006**
VSAM file for your team: **FFSTxx**
User-id and password: **TExx / teamxx**

```
REM ------------------------------------------------
REM Sample for CICS access via ECI interface
REM ------------------------------------------------
set CTGDIR=C:\Program Files\IBM\IBM CICS Transaction Gateway
set
CLASSPATH=.;%CTGDIR%\classes\ctgclient.jar;%CTGDIR%\classes\ctgserver.jar;%
CLASSPATH%
java FFStoresECI local: 2006 VSE410 FFSTxx TExx password
```

Save your changes and exit Notepad. Next we run the sample program:

Enter: **runeci.bat**

You should see something like this:
(see next page)



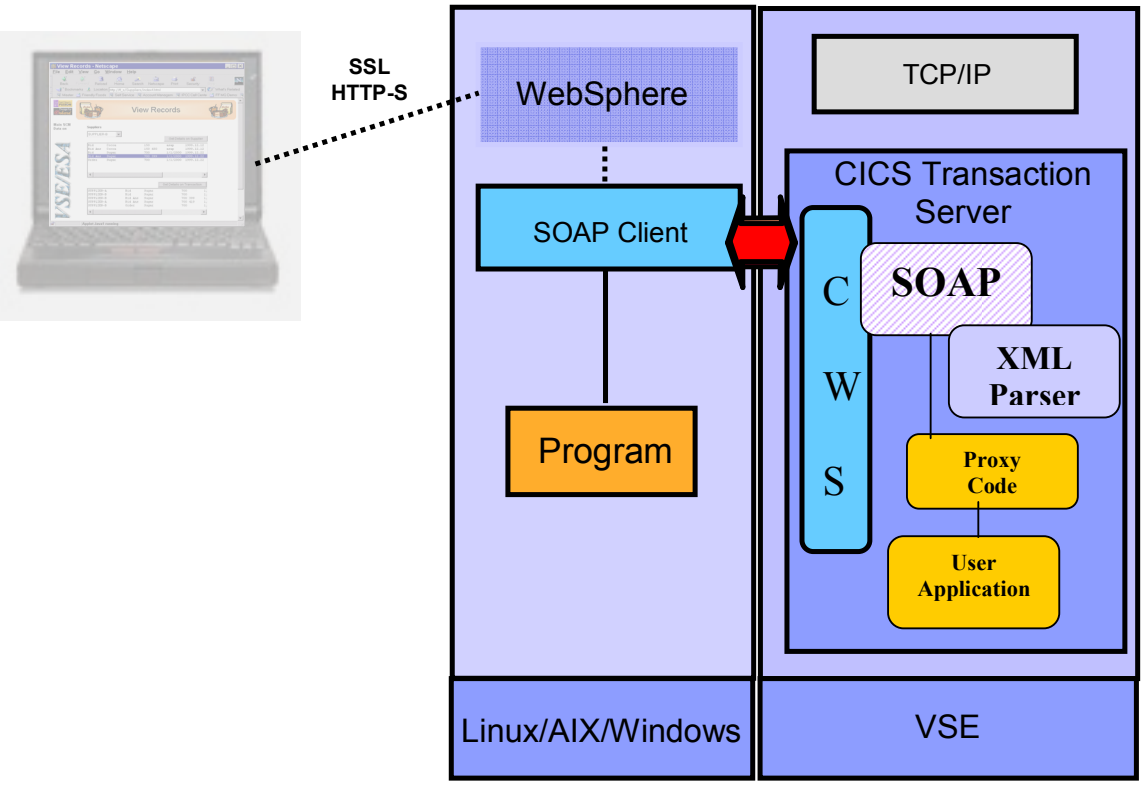
You may scroll up to see the output from the beginning.

The sample program calls the I/O module FFSTIO (refer to structure of the application on page 4) several times to get some records. Please note that this is a very simple sample application, without any user interface. Usually, applications using CTG would run in a web application inside a web application server like IBM WebSphere Application Server.

The call to FFSTIO is done using a COMMAREA. Please note that FFSTIO does not do any screen I/O, it just implements a kind of business logic, in this case the logic to retrieve VSAM records.

Chapter 4.) Setup CICS Web Services

This chapter shows how to integrate CICS applications in distributed environments using Web Services and SOAP (Simple Object Access Protocol).



The benefits of using Web Services are:

- Program to program communications based on open standards.
- Platform independent data interchange using XML with SOAP (Simple Object Access Protocol).
- Using HTTP as the transport protocol allows going through firewalls.

This chapter guides you through the following steps:

- Setup VSE Web Services support (included in VSE/ESA 2.7 and newer)
- Customize and run the SOAP sample program.
- Usage of the CICS2WS Tool to generate the proxy code.

Step 4.1: Setup Web Services Support in VSE

The VSE Web Services Support is based on CICS Web Support (CWS) which is a function of CICS TS in VSE. Therefore CICS Web Support has to be setup (please see Step 2.2:) and a TCP/IP service has to be defined for use with CTG.

The SOAP Engine on VSE does not need any additional setup.

In order to call the CICS program FFSTIO which is accessible via COMMAREA, a SOAP proxy program (FFSTSOAP) was created prior to this workshop, to make the translation from the incoming XML data to a COMMAREA (refer to structure of the application on page 4).

The SOAP engine on VSE gets the XML data stream, parses it using the VSE internal XML parser and then calls the SOAP proxy program (FFSTSOAP) which then will call the FFSTIO program. FFSTSOAP builds the COMMAREA to communicate with FFSTIO program in CICS. The COMMAREA structure used by FFSTIO is described in Appendix C on page 30.

To do Web Services using SOAP a TCP/IP service has to be defined in CICS. For this workshop, we will use the same TCPIP Service and Port from CWS, which we have already defined earlier in this workshop (see Chapter 2.).

TCPIPService: **CWSxx**
Port: **80xx** (were xx is the team number)

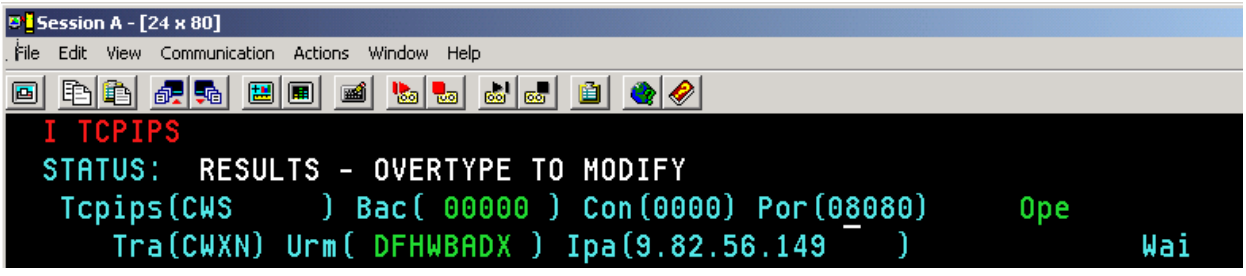
Note: If you wish, you can also define a separate TCP/IP Service for use with Web Services (i.e. TCPIPService SOAP, Port: 1080). The definition would look exactly as the one for CWS.

Step 4.2: Verify if TCPIP Service in VSE is opened

In the Interactive Interface main panel as described in Step 2.1:):

Hit **PF6** and enter
CEMT I TCPIPS

Look for your CWSxx name. You should see something like:



If the status is **CLO**sed, open it by overtyping it with **OPE**n and pressing enter.

Step 4.3: Run the SOAP sample program

For the workshop, we use a sample program. It was prepared already prior toe workshop and was copied to your workstation in directory "**`C:\soap`**". The sample program needs some Java libraries that can be downloaded from internet as described in Appendix B on page 29.

In a Windows command prompt change to C:\soap and edit the batch script **runsoap.bat** and adapt it for your team:

Enter: **C:**
cd C:\soap
Notepad runsoap.bat (make the changes described below and save them)

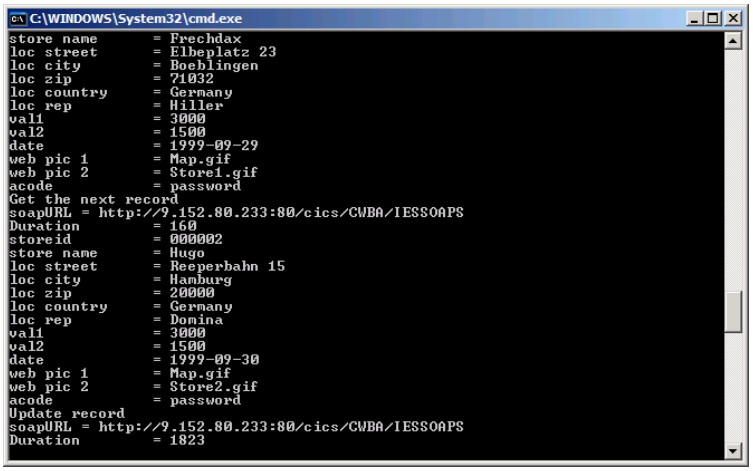
VSE IP address: 192.168.23.11 (Note: the P address may be different)
Port: 80xx
VSAM file for your team: FFSTxx.

```
REM -------------------------------------------------------
REM Sample for CICS access via WEB Services with SOAP and XML
REM -------------------------------------------------------
set CLASSPATH=.;j2ee.jar;soap.jar;xerces.jar;mail.jar;activation.jar
java FFStoresSOAP http://192.168.23.11:80xx/cics/CWBA/IESSOAPS FFSTxx
```

Save your changes and exit Notepad. Next we run the sample program:

Enter: **runsoap.bat**

You should see something like this:



You may scroll up to see the output from the beginning.

Similar to the CTG sample program, this sample program calls the I/O module FFSTIO through the SOAP proxy program FFSTSOAP (refer to structure of the application on page 4) several times to get some records. Please note that this is a very simple sample application, without any user interface.

Step 4.4: Using the CICS2WS Tool to generate the proxy code

In order to service enable a CICS program, you need to create a so called proxy program that translates input and output parameters from the SOAP specific format into a standard COMMAREA format.

In the previous steps, we have used a hand coded proxy program called FFSTSOAP (refer to structure of the application on page 4). To make it easier for you, IBM provides a tool called CICS2WS Tool. This tool can automatically generate the proxy program for you.

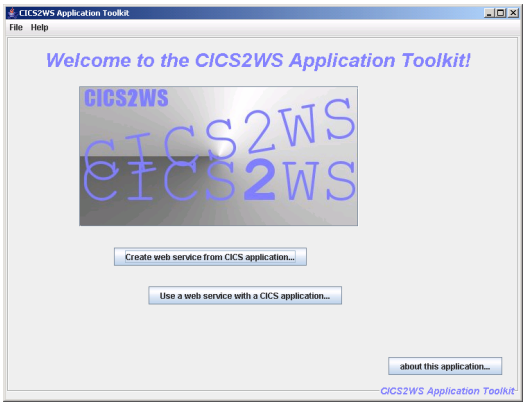
The CICS2WS Tool can be downloaded from the IBM web page:
http://www.ibm.com/servers/eserver/zseries/zvse/downloads/#cics2ws

We have already downloaded and installed the CICS2WS Tool on your workstation prior to the workshop. The CICS2WS Tool was installed in the directory **"C:\CICS2WS"**.

The Tool requires you to download some additional Java libraries from the internet. We have already done that for you. Please see the online help for more details:
C:\CICS2WS\help\howTo.html

To start the CICS2WS Tool, open a command prompt, change to the C:\CICS2WS directory, and run **run_cics2ws.bat:**

Enter: **C:**
cd C:\CICS2WS
run_cics2ws.bat



The tool allows generating proxy programs for:

- Service enabling an existing CICS application (VSE as Web Service provider)
- Call an external Web Service from within a CICS application (VSE as Web Service requestor).

In the workshop, we will service enable the existing CICS application FFSTIO. Therefore, please press the button "Create web service from CICS application".

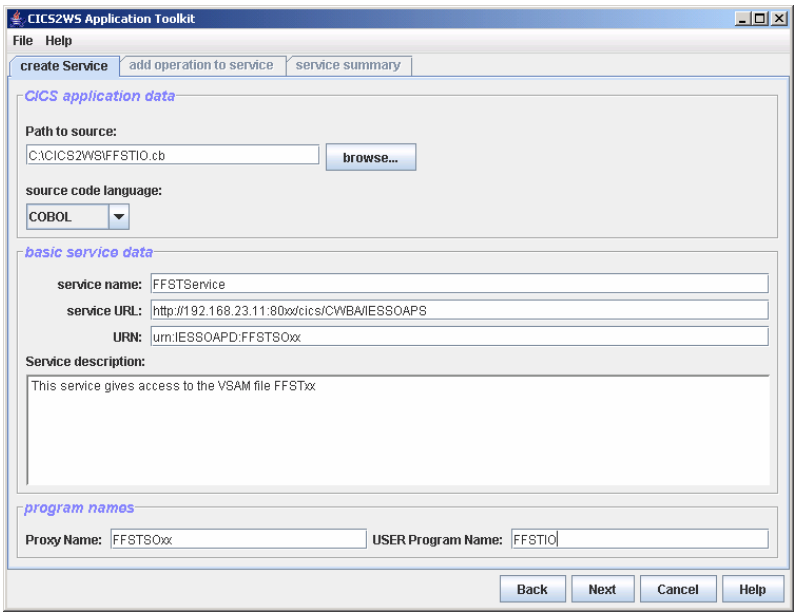
Step 4.5: Create a service for FFSTIO

Now we need to create the service we want to provide for FFSTIO.

First, we provide the COBOL copybook that describes the COMMAREA structure for FFSTIO. Click the "Browse" button and navigate to "**C:\CICS2WS**" and select the file "**FFSTIO.cp**". The tool automatically detects the source code language: **COBOL**.

Second, we need to enter the information about the service and where it will be located later on:

Service name:	**FFSTService**	
Service URL:	**http://192.168.23.11:80xx/cics/CWBA/IESSOAPS**	(xx is your team no.)
URN:	**urn:IESSOAPD:FFSTSOxx**	(xx is your team no.)
Proxy name:	**FFSTSOxx**	(xx is your team no.)
User Program:	**FFSTIO**	



Enter the information as shown above, and press "Next".

Step 4.6: Create a Operation

Conceptually, a Web Service is a kind of a container for operations (or methods). Similar to a Java Class can have multiple methods; a Web Service can have multiple operations. The Web Service definition holds incorporation about the service and its location (as entered in the first page in the tool). An operation holds information about a particular call or service, including information about input and output parameters.

Note: The CICS2WS Tool only supports only one operation per web service.

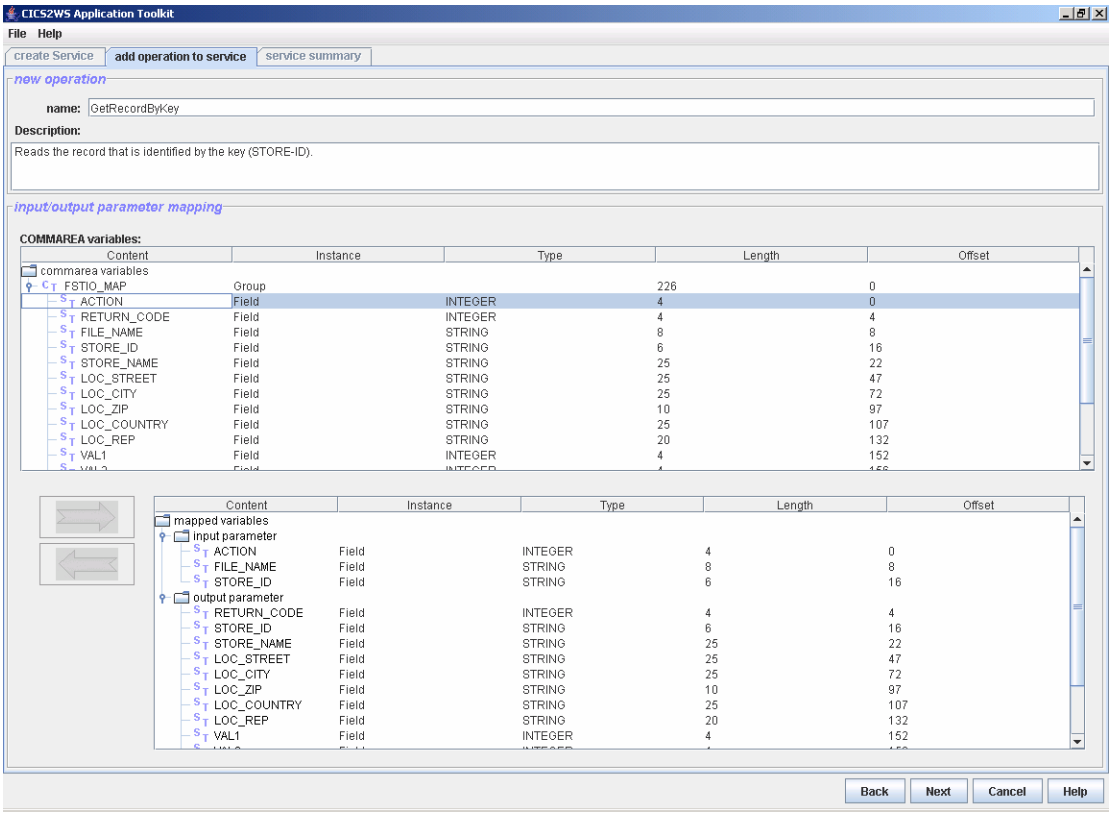
The I/O module we want to service enable, provides several operations like “get a record”, “insert a record”, “update a record” and “delete a record”. In this workshop, we will service enable only one operation:

Operation name: **GetRecordByKey**

In the field COMMAREA variables, you see the field from the COBOL copybook. Next, you need to specify which of the fields are input or output or both for the service:

Input: **ACTION**
FILE_NAME
STORE_ID

Output: **RETURN_CODE**
STORE_ID
...
A_CODE



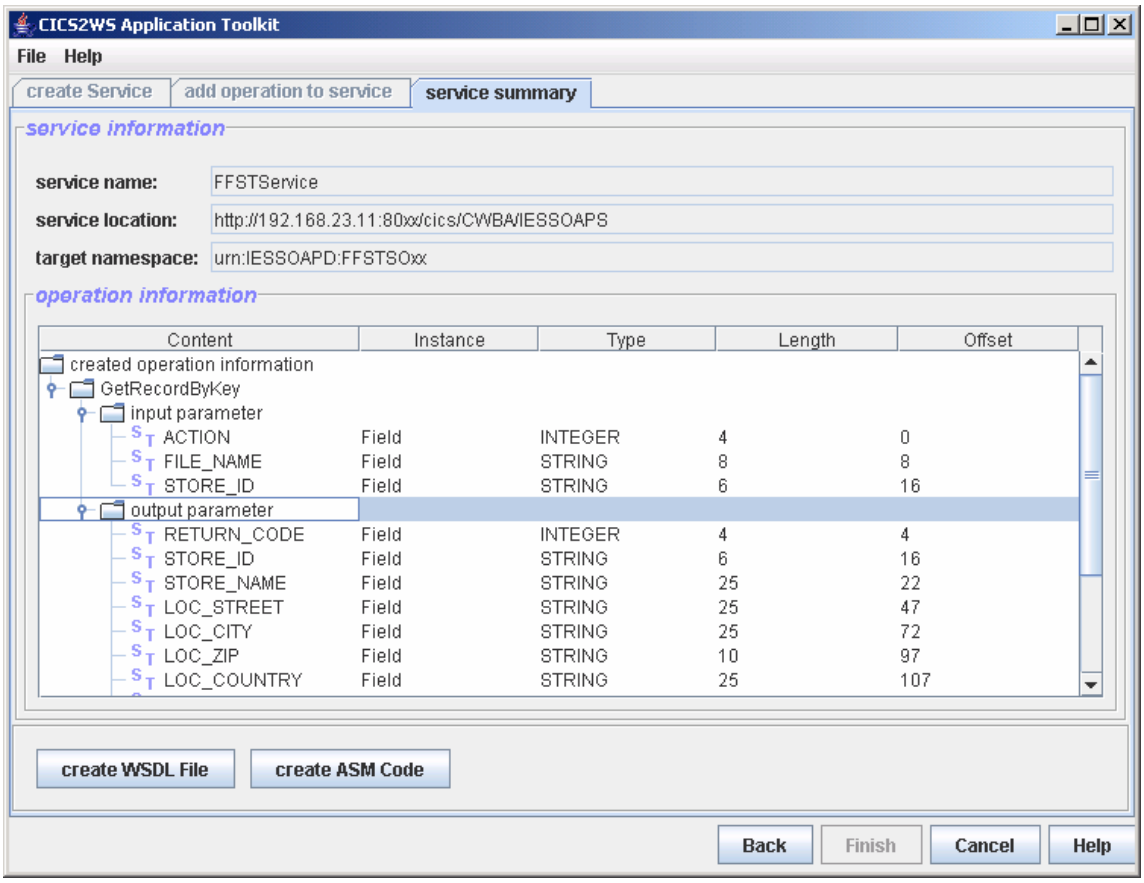
Select a field in the COMMAREA. Also select either “input parameter” or “output parameter” in the list below. Then press the button with the right arrow. This adds the COMMAREA field to the input or output parameters.

You can also change the names of the fields as they appear in the web service definition. In addition, you may build groups and put fields into the group.

When you are finished, press the “Next” button.

Step 4.7: Create the proxy code and WSDL

The following screen displays a summary of all the definitions you have done so far.



Press the “create WSDL File” button, to let the tool generate the WSDL (Web Service Description Language) file that describes the service we have created. Navigate to the “C:\CICS2WS” folder and save it as “FFSTService.wsdl”.

Press the “create ASM code” button to let the tool generate proxy program. The proxy program is generated in Assembler Language. Navigate to the “C:\CICS2WS” folder and save it as “FFSTSOxx.a” (where xx is your team number).

Besides the WSDL and the proxy program source code, a compile job has also been generated (called FFSTSOxx.job).

You may now have a look at the generated files. Open Windows Explorer and navigate to “C:\CICS2WS”. The files that have been generated are:

`FFSTService.wsdl`	- Web Service Description (XML)
`FFSTSOxx.a`	- Proxy program source code in Assembler
`FFSTSOxx.job`	- Compile job for proxy program

Step 4.8: Upload and compile the proxy program

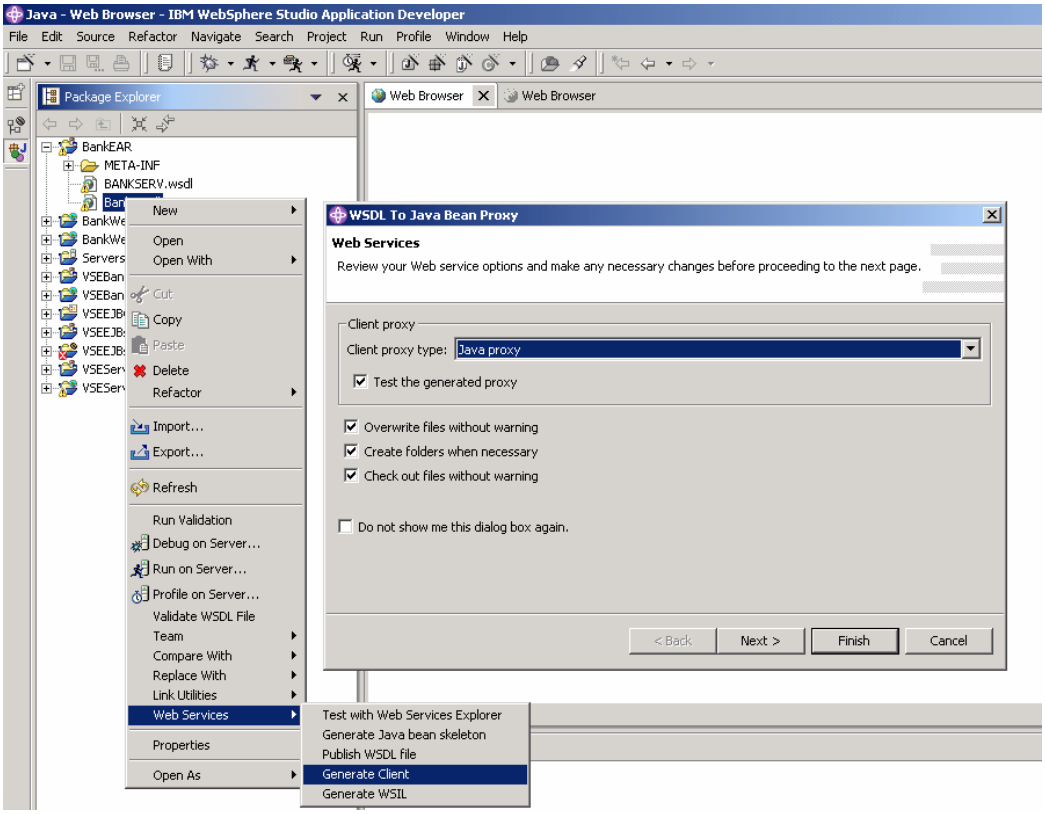
As last step, you would need to upload and compile the generated proxy program on VSE.

Note: We do not perform this step in the workshop.

You can upload the proxy program source code using FTP or any other file transfer method of your choice. Next, adapt the compile job to your needs and run it. You can do so by FTPing it into the VSE reader. Alternatively you may copy both files to your ICCF library and process them from there.

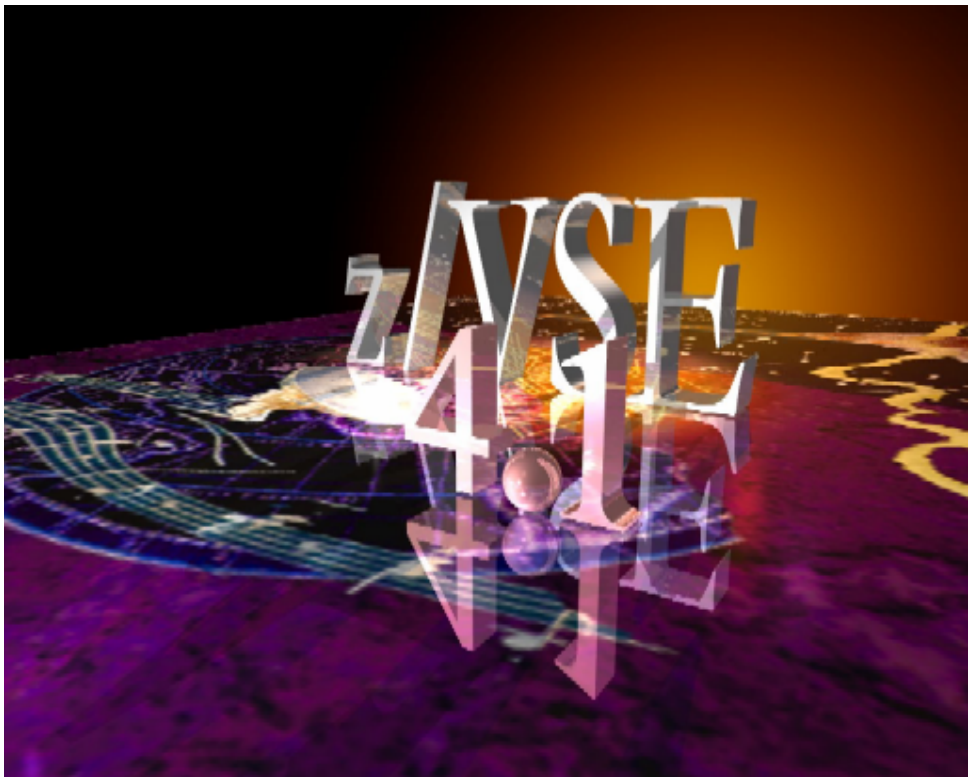
Finally, you would have to define the proxy program to CICS using CEDA DEFINE PROGRAM. The proxy program is a regular CICS program written in Assembler language.

To invoke the web service you have created, you would use the generated WSDL file. Import the WSDL file into your distributed development tool of your choice. You could for example import it into IBM Rational Application Developer (RAD). RAD allows you to generate the client side proxy code from the WSDL file and invoke the web service:



Congratulation

You have completed the workshop.



Appendix A: Setup CICS Web Support in VSE

The following steps have to be done in order to enable CICS Web Support. We have already done these steps for the VSE system we are using in this workshop:

- The following parameters have been changed in DFHSITSP for the CICS TS region DBDCCICS to enable CWS:
  - **ISC=YES** Intersystem communication enabled
  - **TCPIP=YES** TCP/IP protocol enabled
- Configure and enable codepage conversion in CICS. The IBM provided skeleton **DFHCNV** (ICCF library 59) has been adapted and submitted.
- Configure and enable the CICS Web Error program. The IBM provided skeleton **DFHWBEP** (ICCF library 59) has been adapted and submitted.
- The BMS map for application FFST was compiled with option **`SYSPARM='TEMPLATE'`**. This produces a HTML template that was adapted and stored in PRD2.DFHDOC. Please see the following Job how to generate the template:

```
* $$ JOB JNM=FFSTMAP,DISP=D,CLASS=A,NTFY=YES
* $$ LST DISP=D,CLASS=Q,PRI=3
// JOB FFSTMAP COMPILE PROGRAM FFSTMAP
 .....
#/ JOB FFSTMAP CATALOG MAP FFSTMAP
 .....
#/ JOB FFSTMAP CATALOG HTML  FFSTMAP
// EXEC LIBR
   ACCESS SUBLIB=PRD2.DFHDOC
* $$ END
// ON $CANCEL OR $ABEND GOTO ENDJ3
// OPTION NOLIST,ALIGN,DECK,SYSPARM='TEMPLATE'
// EXEC ASMA90,SIZE=(ASMA90,64K),PARM='EXIT(LIBEXIT(EDECKXIT)),
               SIZE(MAXC-200K,ABOVE)'
   PRINT NOGEN
* $$ SLI MEM=FFSTMAP.A,S=PRIMARY.WKS
/*
/. ENDJ3
// EXEC IESINSRT
/*
#&
$ $$ EOJ
* $$ END
/. ENDM
/&
* $$ EOJ
```

- The CICS startup job has been adapted to include the library PRD2.DFHDOC in the **LIBDEF**. This is required to allow CICS to find the HTM templates.

Appendix B: Download the required packages for SOAP

To run the SOAP sample program that we use in the workshop, you need to download additional Java packages from the internet.

Download the packages for SOAP

You have to download following packages (into a temp directory):

- Apache SOAP package: http://xml.apache.org/soap/
  Change into the directory with the latest version (e.g. version-2.3.1) and download the soap-bin package (e.g. soap-bin-2.3.1.zip)

- Apache xerces XML Parser: http://xml.apache.org/xerces-j/index.html
  Download the latest Xerces-J-bin package, e.g.
  Xerces-J-bin.1.4.4.zip

- Sun Java Mail API: http://java.sun.com/products/javamail/

- Sun JavaBeans Activation FrameWork (JAF):
  http://java.sun.com/products/javabeans/glasgow/jaf.html

Extract needed SOAP archives

To simplify the CLASSPATH definition save all .JAR files needed to run the SOAP sample into the same directory. Extract the .JAR files specified from the downloaded .ZIP files.

- Apache SOAP package: extract the file soap.jar from the soap-bin-2.3.1.zip file.
- Apache xerces XML Parser: extract the file xerces.jar from the Xerces-J-bin.1.4.4.zip file.
- Sun Java Mail API: extract the file mail.jar from the javamail-1_2.zip file.
- Sun JavaBeans Activation FrameWork (JAF): extract the file activation.jar from the jaf1_0_1.zip file.

You should now have the following files in the directory:

```
activation.jar
mail.jar
soap.jar
xerces.jar
```


Appendix C: COMMAREA for program FFSTIO

The following COMMAREA structure is used by the I/O module FFSTIO:

```
03 FSTIO-MAP.
   05 ACTION             PIC 9(8) COMP.
   05 RETURN-CODE        PIC 9(8) COMP.
   05 FILE-NAME          PIC X(8).
   05 STORE-ID           PIC X(6).
   05 STORE-NAME         PIC X(25).
   05 LOC-STREET         PIC X(25).
   05 LOC-CITY           PIC X(25).
   05 LOC-ZIP            PIC X(10).
   05 LOC-COUNTRY        PIC X(25).
   05 LOC-REP            PIC X(20).
   05 VAL1               PIC 9(8) COMP.
   05 VAL2               PIC 9(8) COMP.
   05 DATE               PIC X(10).
   05 WEB-PIC1           PIC X(20).
   05 WEB-PIC2           PIC X(20).
   05 A-CODE             PIC X(10).
   05 FILLER             PIC X(6).
```

Note: The definition here is in the COBOL programming language.

Appendix D: TCPIP Service definition for CICS Transaction Gateway (ECI)

To allow incoming CICS requests from remote sites using CICS Transaction Gateway through External Call Interface (ECI), the CICS Web Support interface must be setup. An additional TPC/IP service must be defined with the Port for ECI requests (1435) and the associated initial transaction name (CIEP).

The TCP/IP service definition parameters are as follows:

```
CEDA  DEFine TCpipservice( ECI      )
  TCpipservice   : ECI
  Group          : VSESPG
  Description  ==> SERVICE FOR ECI
  Urm          ==>
  Portnumber   ==> 01435              1-65535
  Certificate  ==>
  STatus       ==> Open               Open | Closed
  SSl          ==> No                 Yes | No | Clientauth
  Attachsec    ==> local              Local | Verify
  TRansaction  ==> CIEP
  Backlog      ==> 00001              0-32767
  TSqprefix    ==>
  Ipaddress    ==>
  SOcketclose  ==> No                 No | 0-240000
```


Additional information

- z/VSE Home Page
  http://www.ibm.com/servers/eserver/zseries/zvse/
- e-business Connectors User's Guide SC33-6719
  http://www.ibm.com/servers/eserver/zseries/zvse/documentation/#conn
- VSE Connectors: Components and Tools
  http://www.ibm.com/servers/eserver/zseries/zvse/downloads
- VSE solutions
  http://www.ibm.com/servers/eserver/zseries/zvse/solutions

IBM Redbooks and publications:

• z/VSE Basics	SG24-7436
• e-business Connectivity for VSE/ESE	SG24-5950
• e-business Solutions for VSE/ESA	SG24-5662
• Servlet and JSP Programming	SG24-5755
• Linux Web Hosting with WebSphere, DB2, and Domino	SG24-6007
• CICS Transaction Server for VSE – CICS Web support	SG24-5997
• WebSphere V5 for Linux on zSeries Connectivity Handbook	SG24-7042

- zJournal Articles about z/VSE and SOAP:
  http://www.zjournal.com/index.cfm?section=searchresults

Contact:

You can reach us at the following e-Mail address:

zVSE@de.ibm.com

