

Installation and Configuration Guide



Attachmate®

**Verastream®**

Bridge Integrator



# Table of Contents

<i>Verastream Bridge Integrator 4.0</i> .....	3
Enhanced SSH support using Reflection for Secure IT, v 7.2 .....	3
Prerequisites: .....	3
<i>Designer Supported Platforms</i> .....	3
<i>Verastream Bridge Integrator Requestors</i> .....	4
<i>Requestor Supported Platforms</i> .....	4
Notes on Supported Platforms .....	5
<b>Using a.NET requestor to create an application in Developer Studio .NET</b> .....	<b>5</b>
<i>Visual Basic .NET Application Code</i> .....	5
<i>Using the Java Application Requestor</i> .....	6
Installing a Requestor .....	8
<b>Installing the Data Engine</b> .....	<b>8</b>
<i>Overview</i> .....	8
<i>Data Engine Supported Platforms</i> .....	8
<i>Additional Information on Supported Platforms</i> .....	8
<b>Installing and Customizing the Data Engine</b> .....	<b>9</b>
<i>Installation steps</i> .....	9
Collect required site and CICS region-specific information .....	9
Review prerequisites and security considerations .....	10
<i>Installation Notes</i> .....	11
<i>Locate the Data Engine package in the Windows installation</i> .....	11
<i>Transfer the Data Engine package to a host program data set</i> .....	11
<i>Run the supplied library extractor to restore additional INSTLIB members and additional product libraries</i> .	14
<b>Customize the Data Engine</b> .....	<b>17</b>
<i>Analyzer Customization for binary HTTP packet traffic (optional)</i> .....	19
<i>XTR1S001</i> .....	19
<i>XTR1S001 TYPE=ANALYZER</i> .....	20
<i>XTR1S001 TYPE=IPPOOL</i> .....	21
<i>XTR1S001 TYPE=XMLDEF</i> .....	21
<i>XTR1S001 TYPE=FINAL</i> .....	23
<b>Complete Customization</b> .....	<b>26</b>
<i>Runtime Configuration Customization (optional)</i> .....	26
<i>Security Mechanism Customization</i> .....	28

Enhanced SSH support using Reflection for Secure IT, v 7.2.....	30
<i>Server Customization (optional)</i> .....	31
<i>Good Morning Message Customization</i> .....	31
<i>Complete Server Customization</i> .....	32
<b>Define Terminal Pool ID Manager Configuration VSAM File</b> .....	<b>33</b>
<i>Set Up ID Manager Bridge Facility Pools</i> .....	34
<b>Make Product Executables Available to CICS</b> .....	<b>34</b>
<i>Install the CICS Resource Definitions</i> .....	34
<b>Internationalization: Model terminal implications</b> .....	<b>36</b>
Customization for code page support.....	36
<b>TCTUA Handling (Optional)</b> .....	<b>37</b>
<b>About CICS Multi-Region Operations (MRO)</b> .....	<b>38</b>
<b>Getting Started Customizing the Data Engine</b> .....	<b>40</b>
<i>Installation Checklist</i> .....	40

# Verastream Bridge Integrator 4.0

## Enhanced SSH support using Reflection for Secure IT, v 7.2

### Prerequisites:

\*Mainframe must have IBM Ported Tools for z/OS setup to decrypt anything coming in through a port (identified below in example) and forward it to another port.

### To take advantage of the enhanced SSH support:

1. Install Reflection for Secure IT, v.7.2 to access it's SSH Windows client.
2. Open a command prompt and issue the following command to secure the SSH tunnel to your mainframe:

```
C:> C:\Program Files\Attachmate\RSecure\ssh.exe -L port_number:hostname:xxxx
```

Where:

The port\_number is that of the port configured in the VBI client. And xxxx is dependent on how the user configured VBI within CICS

For example, `-L22318:abc.example.com:2318 -N user1@abc.example.com` tells the RSIT client to encrypt and reroute anything on the local box sent out over TCP port 22318 (this port number must match that set up in the VBI client) to the host at IP address, abc.example.com, and when the message arrives at the mainframe, decrypt it and send it to port 2318 (dependent on how VBI was configured within CICS) at the same IP address (abc.example.com) using the RACF user ID, user1.

## Designer Supported Platforms

The Verastream Bridge Integrator has two design environments which run on Windows platforms:

- Bridge Integrator Designer

Using this designer you can generate source code modules that provide host access capability to your CICS 3270 Bridge applications. A designer project consists of a set of API calls that access the bridge on the host. These calls can be created, modified, and optimized within the designer, resulting in code that is ready to use or customize in your application. The designer also functions as a hands-on educational tool, for learning about the Bridge API.

Any code generated by this designer can be written manually using the Bridge API. However, the features of the designer can highlight aspects of using the Bridge API that may be helpful or instructive.

- Transaction Studio

With Transaction Studio, you can import COBOL copybooks that describe the transactions that you want to expose through your custom application. Client operations expose only the data fields that you choose, allowing for security and simplicity within your application.

Requirements for Verastream Bridge Designer and Transaction Studio	Supported
Operating System	<ul style="list-style-type: none"><li>Windows 7</li><li>Windows 2008 R2 Server (64-bit)</li></ul>

## Verastream Bridge Integrator Requestors

Bridge Integrator provides an easy to use API enabling session connection, HLLAPI-like processing, and symbolic BMS map field name processing of CICS 3270 screen applications and session cleanup. Some of the high level functionality for this symbolic exchange of 3270 information is based on the BMS map sets defined in the CICS 3270 application itself.

For example, to update the account number on a screen where the account number field is named "ACCOUNT" in the BMS map, the caller can exchange the information symbolically. In this case, placing the value "2002" in the "ACCOUNT" field could be done by issuing a `SetFieldValue("ACCOUNT", "2002")`. This capability frees the requestor from row/column dependency in the 3270 application, allowing a reference to information by name (or index).

There is also an API function that works with a collection of fields by name and index instead of row/column positions. The `GetAllFieldInformation` function retrieves all of the field information from the current host screen. The field level information can then be processed at the requesting application without the need for multiple trips to the host.

The Bridge Integrator Requestors support both Java and .NET environments. Each platform server is a thin runtime layer that takes the Bridge API Request function and its parameters, and packages them into a request format expected on the host. On the return trip, the platform server parses the host response and returns the function's response parameters to the client application.

## Requestor Supported Platforms

Requirements for .NET Requestor	Supported
Operating System	<ul style="list-style-type: none"><li>Windows 7</li><li>Windows 2008 R2 Server (64-bit)</li></ul>
Application Server	Microsoft .NET 2.0 or higher

Requirements for Java Requestor	Supported
Operating System	<ul style="list-style-type: none"><li>Solaris (SPARC) 10 and 11</li><li>Solaris (x86_64) 11</li></ul>

	<ul style="list-style-type: none"> <li>• Red Hat RHEL 6.x (86_64)</li> <li>• SUSE E12</li> <li>• Windows 2008 R2 Server (64-bit)</li> </ul>
Application Server	Java 1.5 or greater (tested with 1.5 and 1.6)

## Notes on Supported Platforms

- Applications for the .NET Managed Web Service import the reference from the installed component, and can be written and compiled in Visual Basic, Visual C#, or Visual C++ programs or included in an ASP.
- Installable Java requestors provide Java application support.
- Client Interfaces support Java applications (JSP or servlets) for the UNIX platform, including UNIX System Services (USS) and Linux for OS/390.

## Using a .NET requestor to create an application in Developer Studio .NET

1. Open Microsoft Visual Studio .NET
2. From the File menu, select New, then choose Project.
3. Select Visual Basic Project on the left side then select ASP .NET Web Application.
4. Specify a name for your service. Click OK.
5. Create the User Interface controls that you want to see in your application.
6. From the Project menu, select Add Reference and click Browse.
7. Enter the Windows directory path and file name to the Verastream Bridge Integrator .NET reference VBI.DLL (for example, c:\VBIV4R00\VBI.DLL). Click OK.

## Visual Basic .NET Application Code

'Declare your .NET object using the the newly imported reference eg:

```
<font color="#0000ff">Dim</font> Bridge <font color="#0000ff">As  
New</font> VBI.BridgeAPI("IBM037", "Windows-1252")
```

'Then issue the desired commands to harvest the screen information:

```
'Connect to a host session using a valid set of credentials
    rc = Bridge.ConnectBridge("n.n.n.n", 2327, 2, "USERID", "PASSWD")
'Move to the screen that contains the desired 3270 information
    rc = Bridge.SendKeys("ATM1@E")
    rc = Bridge.GetString(-1, 9, 41, 4, outText)
    TerminalID.Text = outText

'Exit out of your CICS application gracefully before disconnecting
    rc = Bridge.SendKeys("@C")
    rc = Bridge.Disconnect()
```

## Using the Java Application Requestor

In your favorite editor create the Java program as follows:

```
/* import the Bridge Integrator Bridge API from BridgeAPI.jar:*/
import com.attachmate.cicsbridge.ws.*;

/* declare your Java Class eg: */
public class BridgeSimple1 { public static void main(String[] args) {
    /* declare your local variables */

    String hostIP = "1.2.3.4";          /* set this to the IP address of the
CICS region */

    int hostPort = 2317;              /* set this to the TCPIP SERVICE Port defined
in the target CICS */

    int terminalModel = BridgeAPI.MODEL_2;

    String userid = "UNAME";          /* Set up your RACF USERID here */

    String password = "12345"; /* and corresponding RACF PASSWORD here */

    MutableInt scrRows = new MutableInt();

    MutableInt scrCols = new MutableInt();

    MutableObject scrText = new MutableObject();

    int rc = 0;                      // result code of each method call

    try {
```

```
BridgeAPI bridge = new BridgeAPI(); /* create an instance of the
bridgeAPI object */

/* Issue the desired commands to harvest the 3270 screen information eg:
*/

/* Connect to the target CICS host region */

        bridge.connectBridge(hostIP, hostPort, terminalModel, userid,
password);

        /* Move to the screen that contains the desired 3270 information */

        bridge.sendKeys("ATM1@E");

        /* Collect the desired 3270 information */

        rc = bridgeAPI.getString( -1,          /* the value of (-1)
indicated current screen */

                                9,           /* int inStartRow */

                                41,          /* int inStartColumn */

                                4,           /* int number positions */

                                scrText ); /* where the screen text should go
*/

String myterminalID = (String) scrText.getObject();

        System.out.println("You are on Terminal: ");

System.out.println(myterminalID);

/* Remember to exit out of your CICS application gracefully before
disconnecting */

rc = bridge.SendKeys("@C");

rc = bridge.Disconnect()

        }

        catch (Exception e) {

                // TODO Auto-generated catch block

                e.printStackTrace();

                System.exit(16);

                }

        }

}
```



```
/* Save your program and compile with: javac -classpath .;BridgeAPI.jar  
BridgeSimple1.java */  
  
/* Run with : javac -classpath .;BridgeAPI.jar BridgeSimple1.class */
```

### Installing a Requestor

To install the .NET requestor on a computer running Windows Server copy the .Net VBI reference 'VBI.DLL' to the desired Windows directory.

To install the Java requestor on a computer Java archive 'BridgeAPI.jar' to the desired target directory.

To install the VBI Designer, copy the BridgeDesigner.zip archive to the desired location and extract the contents into the target developer directory.

To install the Transaction Studio, copy the TransactionStudio.zip archive to the desired location and extract the contents into the target developer directory.

## Installing the Data Engine

### Overview

The Data Engine is the host component of the Verastream Bridge Integrator. The Data Engine is installed in each CICS region that your installation sets up to receive network requests from the Bridge Integrator CICS server or client.

### Data Engine Supported Platforms

The Data Engine installs in the following environment:

Requirements for	Supported
Operating System	OS/390 V2R10 and z/OS
Transaction Server	CICS Transaction Server for z/OS 3.1 and above
Other	TCP/IP must be configured and active

### Additional Information on Supported Platforms

In an MRO environment, the load library, containing the BMS mapsets, must be available in the CICS region where the Data Engine is installed (the Routing or Terminal Owning Region). Ensure the load library containing the BMS mapsets is included in the CICS PROC/JCL DFHRPL concatenation for that CICS region.

Not all CICS 3270 applications are compatible with VBI. The Data Engine adheres to the same restrictions as IBM documents as programming considerations for Link3270 Bridge applications.

If possible re-assemble BMS mapsets that were assembled on CICS TS releases prior to CICS TS 1.2. This ensures that the BMS mapset load modules contain the symbolic field information in the application data structure (ADS) that will provide full functionality and the capability to access fields by name in the client and designer.

Refer to the *CICS External Interfaces Guide* for your installation's release of CICS TS for more information regarding these issues.

For proper operation, the IBM supplied module DFH0CBRF must be available in the CICS DFHRPL. If it is not available, the source can be located in the CICS library member SDFHSAMP(DFH0CBRF), compiled and link-edited into an accessible load library in the DFHRPL concatenation. There is a sample CEDA definition for the program in the IBM supplied group DFH\$BR. The program may be auto-installed if program auto-install is active (SIT parameter PGAIPPGM=ACTIVE).

By default, the supplied CICS Resource definitions (VBICICSD) refer to IBM supplied TYPETERM definitions that are in the RDO group DFHTYPE. If you use these definitions unmodified, ensure the DFHTYPE group is installed in the CICS region or these TYPETERMs are copied to one of your installation defined RDO groups.

Install the IBM-supplied CSD group DFHWEB to supply the CWXN and CWBA transactions.

The CBRF terminal definition is defined in the IBM-supplied RDO group DFHTERM. If required, make sure this definition is available.

## Installing and Customizing the Data Engine

There are two major steps in installing the Data Engine; the installation and the customization.

### Installation steps

1. Collect required site and CICS region-specific information
2. Review prerequisites and security considerations
3. Locate the Data Engine package in the Windows installation
4. Transfer the Data Engine package to a data set on your host program
5. Expand the package to create the INSTLIB library
6. Run the supplied library extractor to restore additional INSTLIB members and additional product libraries

### Collect required site and CICS region-specific information

- High Level Qualifier (HLO) you intend to use for the product libraries. This guide uses the HLO ATTMF.

- Allocation information for the product libraries, such as UNIT, VOLSER, STORCLAS, as required by your site.
- CICS region SDFHLOAD library dataset name.
- CICS DFHCSD dataset name.
- Host Code Page requirements.
- SSL and authentication requirements.
- Desired CICS RDO Group name and List name.
- The TCP/IP port for client access. Attachmate recommends and reserves port 2317. If you intend to install the Data Engine into more than one CICS region, you may want to specify different ports for each region. Make this change when you update the CICS Resource Definitions (VBICICSD).
- The Clean up Transaction ID.  
Whenever possible use ATMC as the Clean up Transaction ID. If this name conflicts with another existing transaction on your system, you can change it to a name of your choice. Make this change with the Converter Customization and by updating the CICS resource definitions (VBICICSD).
- The Sample/IVP Transaction ID.  
Whenever possible use ATM1 as the Sample/IVP Transaction ID. If this name conflicts with another existing transaction on your system, you can change it to a name of your choice. Make this change when you update the CICS Resource Definitions (VBICICSD).
  - The ID Manager Pool Status Display Transaction ID.  
Whenever possible use IDMG as the ID Manager Pool Status Display Transaction ID. If this name conflicts with another existing transaction on your system, you can change it to a name of your choice. Make this change when you update the CICS Resource Definitions (VBICICSD).
  - To use SSL, follow the steps for your CICS release for assigning a keyring and server certificate to your CICS regions. Refer to the *CICS RACF Security Guide*. SSL client authentication may also be used. Use a different CICS Transaction ID for the Analyzer to provide security through a UserId on the host engine.

### Review prerequisites and security considerations

- To install the product, you must have sufficient authority to update the product libraries; read access to the SDFHLOAD library and update access to the CICS region DFHCSD is required.
- Verify that the CICS region has read access to the product LOADLIB library or other CICS DFHRPL library into which the product load modules are stored.
- Set up the appropriate permissions for:
  - Execution of the Cleanup transaction (ATMC by default)
  - (Optional) The Sample/IVP transactions (ATM1) and ID Manager Pool Status Display transaction IID (IDMG).

- Other CICS security permissions may be required, depending on which CICS security facilities you are using. For example, if program security is in use (XPROG=YES or XPROG=classname in the CICS SIT), then permissions may be required to execute Data Engine programs.
- (Optional) Reserve the chosen TCP/IP port. Refer to the *z/OS Communications Server: IP Configuration Reference* for your current z/OS release on how to use the PORT statement in the TCP/IP Profile.
- The Web Analyzer commonly runs under the CICS default UserID until the Client UserID and password are validated. If necessary, a different UserID can be specified in the Analyzer Customization.
- Application security based upon CICS virtual terminal ID and Netname can be set up. See the Server Customization section of this document.
- For SSL connections to CICS, CICS must have a keyring and default server certificate assigned.

## Installation Notes

- Reassigning file ownership. If you perform an installation while logged in as the root user, you can re-assign file ownership for all Attachmate files to another user. You can choose a user name other than the default. If this user does not exist on the system, the installation script creates it for you, and prompts you for group name, home directory, and password. If you specify a user other than the default, you must specify an existing user account for which a home directory already exists. Assigning files to a system user account such as "nobody" or "other" without an existing home directory will prevent the HTTP engine from restarting properly.
- Installing with limited rights. If you are not logged on as the root user or if you do not have root-level rights, your installation location may be limited to your own home directory.

## Locate the Data Engine package in the Windows installation

From the Server directory of the Verastream Bridge Integrator product copy the Data Engine package. The file name is VBI.V4R00.INSTLIB.XMIT

## Transfer the Data Engine package to a host program data set

The Data Engine package, VBI.V4R00.INSTLIB.XMIT, is in TSO Transmit/Receive (IDTF) format.

Transfer the file to a host system dataset in binary format and allocate the dataset with RECFM=FB and LRECL=80. These allocation parameters are required for the TSO RECEIVE command to process the dataset successfully. The package dataset will be about three 3390 cylinders.

Transfer the file to a dataset with the selected HLO to this installation using the chosen allocation parameters (UNIT, VOLSER, STORCLAS, etc.) as appropriate.

You can use the file transfer capability of your 3270 emulator or a graphical or command line FTP client to transfer the package to a host dataset.

The following example illustrates how to transfer the package to a dataset with the HLQ of ATTMF using the Windows command line FTP client using default allocation parameters.

```
C:\>ftp s390.attachmate.com

Connected to s390.attachmate.com.

220-FTPSERVD IBM FTP CS V1R5 at S390.ATTACHMATE.COM, 16:28:09 on 2012-10-05.

220-connecting from master.attachmate.com

220 Connection will close if idle for more than 5 minutes.

User (s390.attachmate.com:(none)): ibmuser

331 Send password please.

Password:

230 IBMUSER is logged on. Working directory is "IBMUSER.".

ftp> cd 'attmf'

250 "ATTMF." is the working directory name prefix.

ftp> quote site recfm=fb lrecl=80 blksize=0

200 SITE command was accepted

ftp> quote site primary=3 secondary=1 cylinders

200 SITE command was accepted

ftp> bin

200 Representation type is Image

ftp> put vbi.v4r00.instlib.xmit

200 Port request OK.

125 Storing data set ATTMF.VBI.V4R00.INSTLIB.XMIT

250 Transfer completed successfully.

ftp: 2840720 bytes sent in 59.02Seconds 48.14Kbytes/sec.

ftp> quit

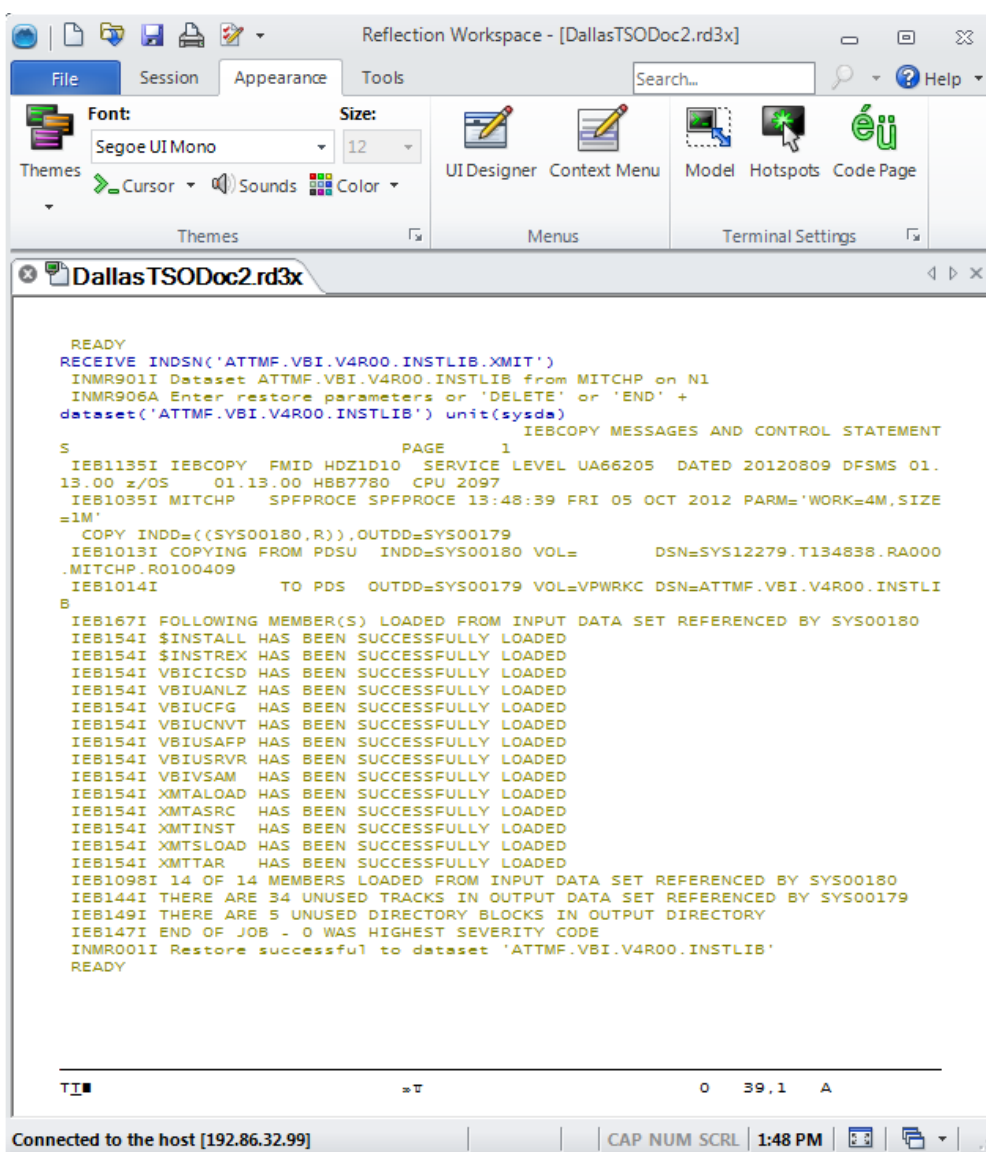
221 Quit command received. Goodbye.
```

C:\Temp>

Expanding the package to create the INSTLIB library

- Use the TSO RECEIVE command to create the INSTLIB library from the Data Engine package that you have transferred to a host dataset. The command can be issued from the TSO READY prompt or from the ISPF Command Shell (Option 6). Assuming the package was transferred to dataset 'ATTMF.VBI.V4R00.INSTLIB.XMIT', the command would be,
- RECEIVE INDSN('ATTMF.VBI.V4R00.INSTLIB.XMIT')
- The command pauses and if necessary, you can specify allocation parameters. You can also override the dataset name of the library. If you do not receive the prompt, the TSO account may have NOPROMPT set in its profile. You can use the TSO FILE PROMPT command to turn this on.

The following example, illustrates a TSO RECEIVE command to create the INSTLIB, overriding the UNIT and dataset name:



```

Reflection Workspace - [DallasTSODoc2.rd3x]
File Session Appearance Tools Search... Help
Font: Segoe UI Mono Size: 12
Themes Cursor Sounds Color
UI Designer Context Menu Model Hotspots Code Page
Themes Menus Terminal Settings

Dallas TSODoc2.rd3x

READY
RECEIVE INDSN('ATTMF.VBI.V4R00.INSTLIB.XMIT')
INMR901I Dataset ATTMF.VBI.V4R00.INSTLIB from MITCHP on N1
INMR906A Enter restore parameters or 'DELETE' or 'END' +
dataset('ATTMF.VBI.V4R00.INSTLIB') unit(sysda)
IEBCOPY MESSAGES AND CONTROL STATEMENT
S
PAGE 1
IEB1135I IEBCOPY FMID HDZ1D10 SERVICE LEVEL UA66205 DATED 20120809 DFSMS 01.
13.00 z/OS 01.13.00 HBB7780 CPU 2097
IEB1035I MITCHP SPFPROCE SPFPROCE 13:48:39 FRI 05 OCT 2012 PARM='WORK=4M,SIZE
=1M'
COPY INDD=((SYS00180,R)),OUTDD=SYS00179
IEB1013I COPYING FROM PDSU INDD=SYS00180 VOL= DSN=SYS12279.T134838.RA000
.MITCHP.R0100409
IEB1014I TO PDS OUTDD=SYS00179 VOL=VPWRKC DSN=ATTMF.VBI.V4R00.INSTLI
B
IEB167I FOLLOWING MEMBER(S) LOADED FROM INPUT DATA SET REFERENCED BY SYS00180
IEB154I $INSTALL HAS BEEN SUCCESSFULLY LOADED
IEB154I $INSTREX HAS BEEN SUCCESSFULLY LOADED
IEB154I VBIICISD HAS BEEN SUCCESSFULLY LOADED
IEB154I VBIUANLZ HAS BEEN SUCCESSFULLY LOADED
IEB154I VBIUCFG HAS BEEN SUCCESSFULLY LOADED
IEB154I VBIUCNVT HAS BEEN SUCCESSFULLY LOADED
IEB154I VBIUSAFP HAS BEEN SUCCESSFULLY LOADED
IEB154I VBIUSRVR HAS BEEN SUCCESSFULLY LOADED
IEB154I VBIVSAM HAS BEEN SUCCESSFULLY LOADED
IEB154I XMTALOAD HAS BEEN SUCCESSFULLY LOADED
IEB154I XMTASRC HAS BEEN SUCCESSFULLY LOADED
IEB154I XMTINST HAS BEEN SUCCESSFULLY LOADED
IEB154I XMTSLOAD HAS BEEN SUCCESSFULLY LOADED
IEB154I XMTTAR HAS BEEN SUCCESSFULLY LOADED
IEB1098I 14 OF 14 MEMBERS LOADED FROM INPUT DATA SET REFERENCED BY SYS00180
IEB144I THERE ARE 34 UNUSED TRACKS IN OUTPUT DATA SET REFERENCED BY SYS00179
IEB149I THERE ARE 5 UNUSED DIRECTORY BLOCKS IN OUTPUT DIRECTORY
IEB147I END OF JOB - 0 WAS HIGHEST SEVERITY CODE
INMR001I Restore successful to dataset 'ATTMF.VBI.V4R00.INSTLIB'
READY

T I 0 39,1 A
Connected to the host [192.86.32.99] CAP NUM SCRL 1:48 PM

```

Example TSO Receive of package, creating INSTLIB

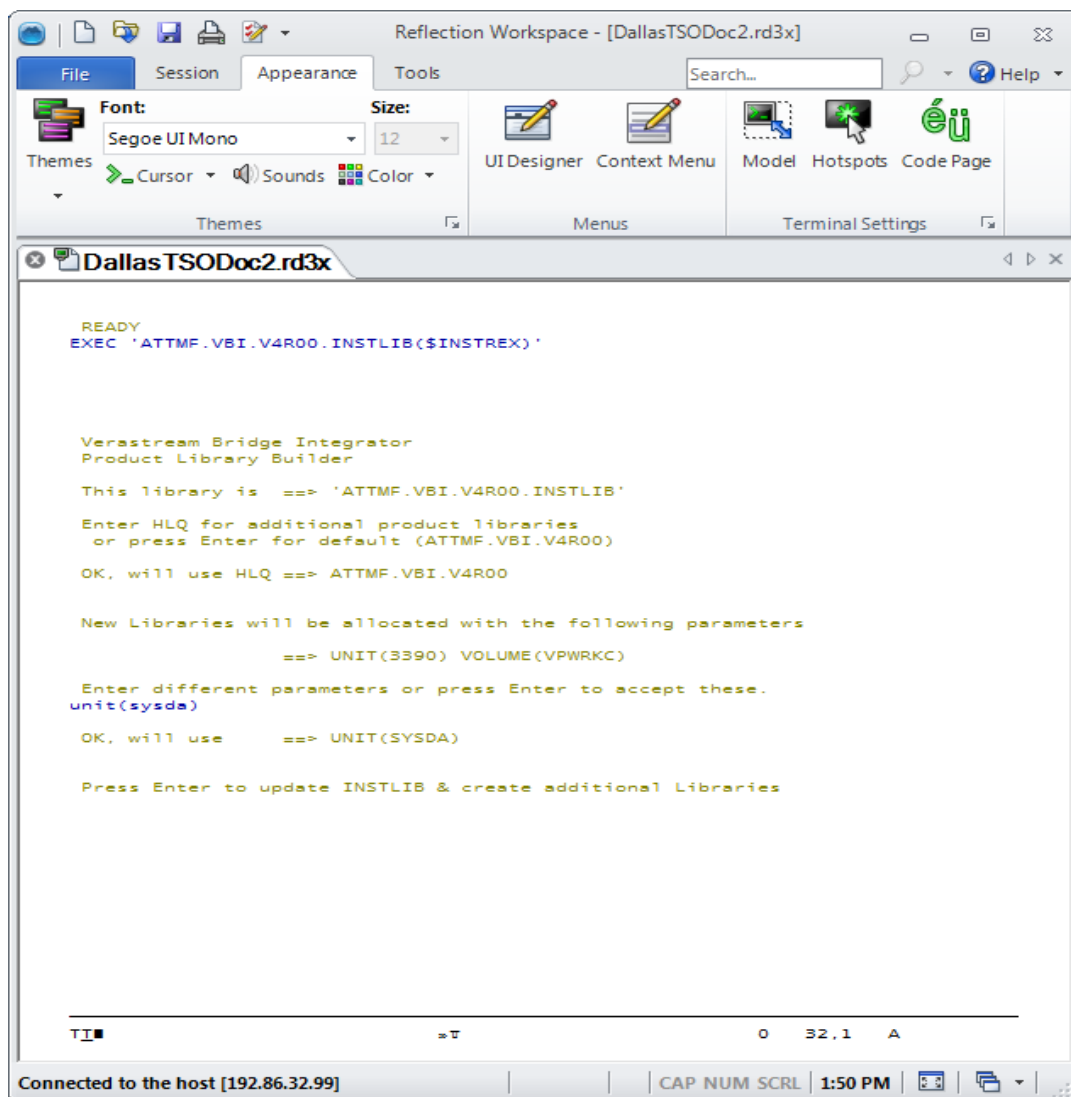
## Run the supplied library extractor to restore additional INSTLIB members and additional product libraries

The remaining Data Engine components are stored in members of the INSTLIB library in TSO TRANSMIT/RECEIVE (ITDF) format. REXX exec, \$INSTREX is an aid provided to allocate additional libraries and to extract these components. Assuming the dataset name for INSTLIB is 'ATTMF.VBI.V4R00.INSTLIB', the command to execute the Library Extractor aid is:

```
EXEC `ATTMF.VBI.V4R00.INSTLIB($INSTREX) `
```

This updates INSTLIB, and creates LOADLIB and ASM libraries. By default, the new libraries have the same dataset names as INSTLIB except for the last dataset name qualifier, as well as, by default, the same allocation parameters (UNIT, VOLSER, STORCLAS) as INSTLIB.

The following example illustrates an execution of \$INSTREX, taking the default dataset names but overriding the allocation parameters to UNIT(SYSDA):



Sample execution of Library Extractor aid, screen 1 of 3

```

READY
EXEC 'ATTMF.VBI.V4R00.INSTLIB($INSTREX)'

Verastream Bridge Integrator
Product Library Builder

This library is ==> 'ATTMF.VBI.V4R00.INSTLIB'

Enter HLQ for additional product libraries
or press Enter for default (ATTMF.VBI.V4R00)

OK, will use HLQ ==> ATTMF.VBI.V4R00

New Libraries will be allocated with the following parameters
==> UNIT(3390) VOLUME(VPWRKC)

Enter different parameters or press Enter to accept these.
unit(sysda)

OK, will use ==> UNIT(SYSDA)

Press Enter to update INSTLIB & create additional Libraries

INMR901I Dataset ATTMF.VBI.V4R00.INSTLIB from MITCHP on N1
INMR906A Enter restore parameters or 'DELETE' or 'END' +
IEBCOPY MESSAGES AND CONTROL STATEMENT
S PAGE 1
IEB1135I IEBCOPY FMID HDZ1D10 SERVICE LEVEL UA66205 DATED 20120809 DFSMS 01.
13.00 z/OS 01.13.00 HBB7780 CPU 2097
IEB1035I MITCHP SPFPROCE SPFPROCE 13:54:35 FRI 05 OCT 2012 PARM='WORK=4M,SIZE
=1M'
COPY INDD=((SYS00189,R)),OUTDD=SYS00187
IEB1013I COPYING FROM PDSU INDD=SYS00189 VOL= DSN=SYS12279.T135435.RA000
***

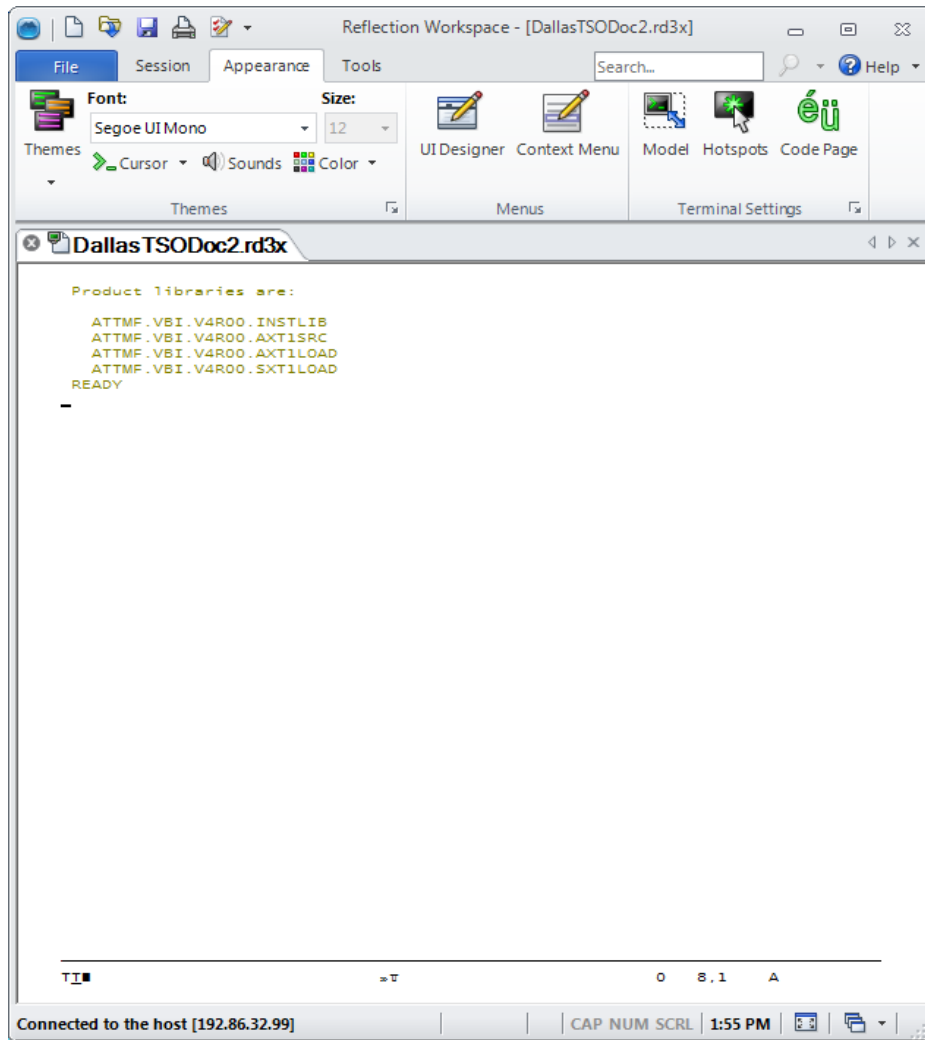
TI# =>T 0 43,6 A

```

Connected to the host [192.86.32.99] | CAP NUM SCRL 1:54 PM

Sample execution of Library Extractor aid, screen 2 of 3





Sample execution of Library Extractor aid, screen 3 of 3

## Customize the Data Engine

*For an overview and checklist of the installation process, see [Customizing the Data Engine – Getting Started – Installation Checklist at the back of this guide](#).*

VBI release 4.0 comes pre-configured with default standard settings. If the default settings are acceptable, then you can simply copy the load modules from the Customized LOADLIB <&HLQ>.VBI.V4R00.SXT1LOAD to a LOADLIB accessible to the target CICS region.

The DFHRPL DD in your CICS startup JCL must refer to the LOADLIB containing these customized load modules:

XTR1A000	Analyzer for binary HTTP packet traffic default Default settings allow for any mid-tier client requester IP addresses to send requests and allow your firewall infrastructure to define acceptable client access. The default transaction ID for the converter and server is preconfigured to CWBA. There is no SPECUID configured.
XTR1A001	Converter for binary HTTP packet traffic Default settings specify 600 second (10 minute) timeout for 3270 bridge terminals, a default cleanup transaction ID of ATMC, and default model terminal IDs of CBRF. A  RC=4 is acceptable, and is generated by linking the configuration (XTR1S002) with the NCAL option, thus leaving final resolution to the links step of the converter and server programs.
XTR1A002	Server for the Verastream Bridge Integrator Default settings specify 600 second (10 minute) timeout for 3270 bridge terminals, a default cleanup transaction ID of ATMC, a default model terminal IDs of CBRF, and specify that the User ID and password access control already in place are sufficient to control access to host transactions

The above parameter specifications are all customizable via the configuration macros defined later in this publication.

XTR1A003	Test 3270 application for validation of product installation Suggested transaction ID is 'ATM1'.
XTR1A005	Optional Web service SOAP processor for SOAP based client requests.
XTR1A006	Optional Web service SOAP Authorization processor for SOAP based client requests.
XTR1A007	Terminal pool configuration 3270 application. Suggested transaction ID is 'IDMG'.

For customization, the JCL you will modify and submit to configure your VBI system resides in the <&HLO>.VBI.V4R00.INSTLIB from the prior step.

---

*There is an overview of the installation and customization steps located in the PDS member: <&HLO>.VBI.V4R00.INSTLIB(\$INSTALL) at the back of this guide.*

---

The JCL members are:

VBIUANLZ - Optional Analyzer Customization JCL

VBIUCFG – Optional Configuration of run-time system

VBIUCNVT - Optional Converter Customization JCL

VBIUSAFP - Optional External Resource Access Control

VBIUSRVR - Optional Server Customization JCL

VBICICSD - Optional Sample CICS CEDA definitions

VBIVSAM - Required VSAM terminal pool configuration file definition

### Additional information:

- Run all the above jobs in the order they are listed above. If the default configuration parameters are acceptable then the only required JCL step is VBIVSAM which defines the Terminal Pool ID Manager KSDS VSAM configuration file.
- Follow the instructions at the top of the individual JCL members
- Check the output after you submit the jobs
- If you encounter a return code greater than 4, check the job output listings for more information.
- Because customization involves the coding of Assembler language macros, you must follow normal Assembly language requirements.

These include:

- Names start in column one and are of eight or fewer characters
- XTR1S001 starts in column 10
- To continue a line, it must end in a comma and contain a continuation character in column 72. The continued portion must start on the next line in column 16.
- The last line of the macro must not end in a comma.

---

*The supplied sample customization jobs include comments to augment the discussion in this document. You should update JOB cards to conform to your own installation standards and guidelines. Follow the instructions at the top of each JCL job to supply user defined values appropriate for your installation.*

---

## Analyzer Customization for binary HTTP packet traffic (optional)

Customize the Data Engine Analyzer if your installation uses:

- A different CICS Web Alias Transaction ID than the IBM supplied CWBA.
- IP address filtering to restrict which network IP addresses can connect to the Data Engine.
- A different UserID for the Analyzer running under the chosen Web Alias Transaction ID (CWBA) than the CICS default UserID.

Refer to the sample job stream in the <&HLQ>.VBI.V4R00.INSTLIB member VBIUANLZ for JCL customizations and refer to the syntax information provided in this guide.

### XTR1S001

The Analyzer Controller template is customized using the XTR1S001 macro.

#### Syntax

```
XTR15001 TYPE= < ANALYZER | IPPOOL | XMLDEF | FINAL >
```

#### Parameters

Name	Description
ANALYZER	Begin definition for the Attachmate Analyzer. An Analyzer is defined by a single TYPE=ANALYZER macro, followed by one or more TYPE=IPPOOL and TYPE=XMLDEF macros. The specification is completed with a single TYPE=FINAL macro.
IPPOOL	Optional. A set of IP addresses or ranges that are defined once and referred to in the definitions of one or more XML messages using the TYPE=XMLDEF macro. Provides flexibility; authorized IP addresses can be entered on each specific XMLDEF macro or defined in a pool and referred to by the IPPOOL= parameter of the XMLDEF macro.
XMLDEF	Define an XML message that, when received over a CWI HTTP connection, is directed to the specified CWI Converter and subsequently to the CWI-compliant server that handles the request.

FINAL	Construct the Analyzer Controller Template (ACT). This template is linked into the Attachmate CWI Analyzer. IP addresses and pools are resolved and efficient IP filtering is linked into the CWI analyzer along with the specifications for each defined XML message and an indication of a default HTTP handler for undefined messages. The Analyzer is assigned designated TCP/IP port using the standard CEDA TCPIP SERVICE customization. (See the IBM CICS Resource Definition Guide for further information.)
-------	--

## XTR1S001 TYPE=ANALYZER

Define a CWI Analyzer and a default analyzer for undefined messages.

### Syntax

XTR1S001 TYPE=ANALYZER, X

DEFAULT= < Name | NONE >

### Parameters

Name	Description
ANALYZER	Begin definition for the Attachmate Analyzer. An Analyzer is defined by a single TYPE=ANALYZER macro.
DEFAULT Name	Name of the CWI Analyzer program load module to be given control in the event that the HTTP message received does not match any of the XML messages defined for the Analyzer. An example might be HTML received which should be routed to the IBM default HTML handler (DFHWBADX) to be processed by the default Web service provided by IBM. This designation allows a single TCP/IP port to be shared between the HTML and XML requests not routed by the Attachmate Analyzer and those messages routed by the Attachmate Analyzer.  NONE indicates that any HTTP to this port not specifically defined in TYPE=XMLDEF macros as Invalid Requests.
DPLS	Specify that Distributed Program Linking (DPL) is supported and the converter program XTR1A008, which supports DPL calls to customer CICS based application programs will be invoked.  DPL=YES enables DPL linking via VBI. DPLS=NONE is the default.
ALIAS	The CICS transaction ID that the converter and server are to execute under. The default value is CWBA.
SPECUID	A valid CICS USERID under which invalid request return codes are returned. This allows a return code indication that a USERID or PASSWORD value passed in is an invalid combination.

## XTR1S001 TYPE=IPPOOL

Define a group of IP addresses for authorization to conduct XML message exchanges with the target CICS region.

### Syntax

```
Name TYPE=IPPOOL, X
    nnn.nnn.nnn.*, X
    nn.nnn.nnn.nnn, X
    nn.nn.nnn.*
```

### Parameters

Name	Description
POOLNAME Name	An eight-character name given to this group of IP addresses. Subsequent definitions of XML messages (XMLDEF) will refer back to this name as the collection of IP addresses to be granted access via this XML message.
IP ADDRESSES nnn.nnn.nnn.nnn	A list (up to 16,000 elements) of IP addresses. Each IP address must consist of four components, each of which is either an integer less than 255 or the wildcard character (*). DNS names cannot be used.

## XTR1S001 TYPE=XMLDEF

Define an XML message, or family of XML messages, and the corresponding CWI Converter, server, and parameters to use to execute XML-based server programs.

### Syntax

```
Name TYPE=XMLDEF, X
    HLQTAG=1stXMLtag, X
    XMLTAGA=2ndXMLtag, X
    TAGAID=value X
    CWICONV=converter, X
    CWISERV=server, X
```

XLAT=< name | NONE >, X

IPPOOL=poolname

[ or ]

nn.nnn.nnn.nnn, X

nn.nn.nnn.\*

## Parameters

Name	Description
MESSAGENAME Name	An optional eight-character name to be given this XML definition. By specifying a name, the group of IP addresses defined in this macro can be treated as an IPPOOL definition, and reference in subsequent XMLDEF macros may be made in the same way an IPPOOL macro is referenced.
HLQTAG 1stXMLtag	The first XML tag to be expected from the HTTP conversation for this XML message type. This tag should be exactly as the first tag in the data stream after the <xml version=1.0> appears, but without the angle brackets (< >). For example, if <ATTACHMATE> is the first expected XML tag for a supported message, this parameter should appear as:  HLQTAG=ATTACHMATE, X
XMLTAGA 2ndXMLtag	The second XML tag to be expected from the HTTP conversation for this XML message type. This tag should be exactly as the second tag in the data stream after the <xml version=1.0> appears, but without the angle brackets. This tag may be identified with a wildcard character (*), indicating that all HTTP XML traffic bearing the HLQTAG should be passed to the Converter without further inspection.
TAGAID Value	If XMLTAGA is not specified as a wildcard, use this parameter to match a specific value inside the XMLTAGA parameter in the actual XML data stream. For instance, the XML stream <ATTACHMATE><CONNECTORID>QACOM_BIN</CONNECTORID> would be defined as TAGAID=QACOM_BIN.
CWICONV converter	The name of the CWI Converter program load module to be given control when this defined XML message is received in the CICS region. For non-Attachmate product XML messages, the converter name must be supplied.
CWISERV server	The name of the Server program load module given control when this defined XML message is received in the CICS region. For non-Attachmate XML messages, the server name must be supplied.

XLAT name	The name of the DFHCNV module used for code page translation. This parameter is optional and defaults to NONE. A name provided will identify which DFHCNV translation module to use. An example module given by IBM is DFHWBHH. Attachmate products will default the correct XLAT parameter where possible.
IPPOOL name	The name of the previously defined pool of acceptable IP addresses from which XML exchanges are allowed. This pool may have been defined by use of the TYPE=IPPOOL, or by putting the specified IPPOOL=name as the Name field on another TYPE=XMLDEF macro. If the name comes from the Name field of a previous XMLDEF macro, the set of specific IP addresses configured for that XMLDEF are the same set allowed to access the transaction defined by this XMLDEF message.
IP ADDRESSES nnn.nnn.nnn.nnn	A list (up to 16,000 elements) of IP addresses. Each IP address must consist of four components, each of which is either an integer less than 255 or the wildcard character (*). DNS names cannot be used.

## XTR1S001 TYPE=FINAL

Specify the conclusion of customization to the Attachmate CWI Analyzer

### Syntax

XTR1S001 TYPE=FINAL

### Examples using various aspects of customization

#### *Filtering Requestor IP Addresses*

This example illustrates the use of IP filtering to limit the requestor addresses to be serviced. In this example, the specific requestor address 140.147.249.7 will be accepted, as will requestor IP addresses beginning with 63.71.228. You can use the wild card character "\*" on any or all of the four-part IP address segments.

Note the use of the SPECUID here. If a client request comes in that fails a CICS Verify check of UserID/PASSWORD then control is subsequently executed under the RACF user ID CICSBADU. This allows intelligent return codes to be processed and bypasses HTTP 400 return codes and the associated core dump from occurring.

```

* ****
* ****
* **              ANALYZER CUSTOMIZED FOR THE              **

```



```
* **                ATTACHMATE DATA ENGINE                **
* ****
* ****

XTR1S001 TYPE=ANALYZER,DEFAULT=NONE,ALIAS=CWBA,      X
          SPECUID=CICSBADU

XTR1S001 TYPE=IPPOOL,HLQTAG=ATTACHMATE,              X
          XMLTAGA=CONNECTORID,                        X
          TAGAID=QACOM_BIN,                            X
          140.147.249.7,                                X
          63.71.228.*

XTR1S001 TYPE=FINAL
```

### *XML Messages Supporting the Same Client IP Addresses*

This example illustrates a customization in which two different categories of XML messages are accepted from a single set of IP addresses. An IPPOOL is set up once at the beginning and then referenced by XMLDEF definitions for both XML message types.

```
XML Stream

QACOM_BIN

...

OR:

PART_INQUIRY

* ****
* ****
* **                ANALYZER CUSTOMIZED FOR THE                **
* **                ATTACHMATE DATA ENGINE                **
* ***                ATTACHMATE SERVER TOOLKIT                **
* ****
* ****

XTR1S001 TYPE=ANALYZER,DEFAULT=DFHWBADX
```

```

ADJUSTER XTR1S001 TYPE=IPPOOL, X
          140.147.249.*, X
          65.205.231.130, X
          63.71.228.*
XTR1S001 TYPE=XMLDEF, X
          HLQTAG=MYCO, X
          XMLTAGA=SUBSYSREQ, X
          TAGAID=PART_INQUIRY, X
          CWICONV=PARTINQC, X
          CWISERV=PARTINQS, X
          XLAT=DFHWBHH, X
          IPPOOL=ADJUSTER
XTR1S001 TYPE=XMLDEF, X
          HLQTAG=ATTACHMATE, X
          TAGAID=QACOM_BIN, X
          IPPOOL=ADJUSTER
XTR1S001 TYPE=FINAL

```

### *Any Client IP Address Permitted*

In this example an XML message has been defined as unsecured. A wildcard IP address has been set up to facilitate traffic from any client that is capable of formatting the proper XML message.

```

* *****
* **          ANALYZER CUSTOMIZED FOR THE          **
* **          ATTACHMATE DATA ENGINE             **
* *****
* *****
XTR1S001 TYPE=ANALYZER,DEFAULT=DFHWBADX
XTR1S001 TYPE=XMLDEF,HLQTAG=ATTACHMATE, X
          XMLTAGA=CONNECTORID, X

```

```
        TAGAID=QACOM_BIN,                X

        *.*.*.

XTR1S001 TYPE=FINAL
```

## Complete Customization

To customize the Analyzer, update the VBIUANLZ job stream in INSTLIB and submit the job.

## Runtime Configuration Customization (optional)

Customize the Data Engine Converter if your installation:

- Uses a different Bridge Facility Models than the IBM supplied CBRF. For example, if your installation uses TCT User Areas (TCTUAs) then you will want to use terminal definitions that refer to TYPETERMs with the correct User Area Length specified.
- Uses a different Cleanup Transaction ID than the Data Engine default, ATMC.
- Certain time out values can be adjusted with Converter customization.

Customize the source for configuration in the <&HLQ>.VBI.V4R00.INSTLIB member VBIUCFG and define any desired changes to the XTR1S003 customization macro therein. The default parameters are as follows.

```
XTR1S003 MOD1FAC=CBRF,    PRINCIPLE TERMINAL FACILITY MOD1    X
                          MOD2FAC=CBRF,    PRINCIPLE TERMINAL FACILITY MOD2    X
                          MOD3FAC=CBRF,    PRINCIPLE TERMINAL FACILITY MOD3    X
                          MOD4FAC=CBRF,    PRINCIPLE TERMINAL FACILITY MOD4    X
                          MOD5FAC=CBRF,    PRINCIPLE TERMINAL FACILITY MOD5    X
                          GMMTRAN=NO,      DONT RUN A GOOD MORNING TRANSACTION X
                          CTRAN=ATMC,      SESS INACTIVE TIMEOUT CLEANUP TRANS X
                          AIDPROC=NOOP,    SERIALIZE AID PROCESS TRANID OR NOOP X
                          FACKPTIM=00000258, FACILITY KEEPTIME OF 600 SECONDS    X
                          GETWAIT=000927C0, SERVER:BRIDGE TIMEOUT 600,000 MSECS X
                          TIMEOUT=00000258 SESSION INACTIV TIMEOUT OF 600 SECS
```

When customization of the XTR1S003 macro is complete, modify the JCL in VBIUCFG as indicated in the comments at the top of that member and the save abs submit the JCL.

Note that a return code greater than 4 indicates a problem with the customization.

Parameter	Description
MOD1FAC through MOD5FAC	3270 terminal profiles for terminal models 1 through 5. If you use a different value, coordinate this change with the CICS resource definitions, VBICICSD, in a later step.
GMMTRAN	<p>Determines the initial state for a session. Using the default value GMMTRAN=NO, the connected session begins at a blank screen, ready to start a CICS transaction. If a valid user-id and password are not supplied with the ConnectBridge API, GMMTRAN can be used to identify the sign on transaction or good morning banner to be issued to the newly created session. Candidates for GMMTRAN must be CSGM, CESN, or a 3270 Bridge compatible CICS transaction program.</p> <p>If a valid set of user-id and password credentials are supplied with the ConnectBridge API, the External Security Manager (RACF) authenticated session begins at a blank screen and the GMMTRAN transaction is bypassed.</p>
CTRAN	The cleanup Transaction ID defined for the data engine server program (XTR1A002). The recommended value is ATMC. If you use a different value, coordinate this change with the CICS resource definitions, VBICICSD, in a later step.
FACKPT1M	The maximum time a terminal facility will be kept before the system frees it. This value needs to be hexadecimal and should be less than the BRMAXKEEPTIME= set up for the CICS region in the SIT.
GETWAIT	The maximum time in milliseconds that CICS will wait for a conversational 3270 response from the client. This value needs to be hexadecimal and should be less than the BRMAXKEEPTIME= set up for the CICS region in the SIT.
TIMEOUT	The maximum inactivity time in seconds for a given VBI terminal and its state memory. This value needs to be hexadecimal and should be less than the BRMAXKEEPTIME= set up for the CICS region in the SIT.

After the job stream is updated, submit the job to assemble the customization modules and relink the Converter by submitting the JCL in member VBIUCNVT.

Once the above runtime configuration process is complete you may continue with customization of the Server module.

If you would like to control access to your application accessed via VBI by the use of user supplied USERID/PASSWORD credentials, the default configuration shipped will suffice then you may proceed with the Server customization. Alternatively, if you would like to restrict access to resources to the system using control based upon resource permission such as CICS region, network name, terminal ID and Transaction ID either by using a RACF specified FACILITY or by invoking a User Written Module to apply customized resource protection based upon the aforementioned resources you may choose to follow the steps in "Security Mechanism Customization" as follows.

## Security Mechanism Customization

To increase the level of security, some installations control which transactions are allowed to run based upon the terminal ID. With the CICS Bridge there is no terminal ID, but the Data Engine can be configured to simulate this level of security.

To allow or restrict access by Applid/Netname/TerminalID/TransactionID, configure the server to invoke CICS application "exit" which your installation writes, or issue a EXEC CICS QUERY SECURITY to control transaction execution.

This customization is controlled with the XTR1S005 macro:

```
SAFPREF XTR1S005 RESSEC=, X
          LOGGING=, X
          RESCLASS=, X
          URMNAME=
```

- **RESSEC=NO** - VBI runs as it always has by using standard CICS systems customized security based upon user ID
- **RESSEC=CUSTOM, URMNAME=<custprog>** - Invoke a customer-provided CICS application program that decides whether or not to grant access to transactions. The User Replaceable Module handles any desired security or logging requirements and returns a zero return code to allow the transaction to continue.

The customer written URM takes responsibility for any challenge to run a given transaction and of any logging of such challenges. The URM is passed a parameter block of:

```
01 SAF-URM-AREA.
      03 URM-RETURN-CODE PIC S9(9) COMP.
      03 URM-SAF-INPUT-PARMS.
      05 URM-XTR1A002-REGION PIC X(8).
*      Note - region APPLID of the region in which XTR1A002 resides
      05 URM-NETNAME PIC X(8).
*      Note - 3270 Bridge 'VTAM Style' Netname for this session
      05 URM-TERMINAL-ID PIC X(8).
*      Note - 3270 Bridge CICS TERMINAL ID for this session
```

```
05 URM-TRANSACTION-ID PIC X(4).
* Note - The proposed 3270 Transaction to be run
```

```
05 URM-USER-ID PIC X(8).
* Note - the CICS USERID under which this transaction is to run.
```

The return value passed back in URM-RETURN-CODE has the following effect:

Zero to allow the call to the IBM linkable 3270 bridge 'DFHL3270'.

or

Non-zero to forbid the transaction from executing

RESSEC=YES, RESCLASS=classname, LOGGING=yesorno - specifies that resource access checking is performed by constructing a RESID for an EXEC CICS QUERY SECURITY call against the specified security profile class. The QUERY SECURITY call is made before passing the request for a given transaction to the IBM CICS 3270 Linkable Bridge. If the user has READ access to the RESID/RESCLASS combination then the request is passed through to the Linkable bridge. If not, the request is denied and a 1330 return code is returned to the client.

The RESID resource string is constructed by the Attachmate VBI server by appending the constant "ATTMBR" + <cics\_region\_applid> + <3270\_bridge\_netname> + <3270\_bridge\_terminal\_id> + <3270\_transaction\_id>

\$ATTMBR	High Level Qualifer CONSTANT
<cics_region_applid>	Detected APPLID of region where XTR1A002 resides
<32790_bridge_netnam>	Netname passed in with ConnectBridge or generated
<3270_bridge_terminal_id>	Four char bridge terminal ID passed in or generated
<3270_transaction_id>	Transaction Identifier that is being invoked

The default RESCLASS value is FACILITY. The use of the FACILITY class can be overridden by specifying a different class name in the **RESCLASS** parameter on the XTR1S005 macro. Whatever class you choose must have a maximum profile size large enough to contain the generated RESID.

The Customer Security Administrator can either:

- Allow all possible terminal IDs except for a specific pattern of names. For example if terminal IDs beginning with MP cannot access transactions beginning with TR or TX in the CICS region with APPLID CICSPROD

```
RDEFINE FACILITY $ATTMBR.** UACC(READ)
RDEFINE FACILITY $ATTMBR.CICSPROD.*.MP*.TR* UACC(NONE)
RDEFINE FACILITY $ATTMBR.CICSPROD.*.MP*.TX* UACC(NONE)

SETROPTS RACLIST(FACILITY) REFRESH
```

- Conversely, if the goal is to disable access to certain transactions from all terminal IDs except for a selected pattern of names:

```
RDEFINE FACILITY $ATTMBR.** UACC(NONE)
RDEFINE FACILITY $ATTMBR.CICSPROD.*.MP*.TR* UACC(READ)
RDEFINE FACILITY $ATTMBR.CICSPROD.*.MP*.TX* UACC(READ)
```

```
SETOPTS RACLIST(FACILITY) REFRESH
```

Often, using a UACC of other than NONE is discouraged on any RACF profile. In this case they may want to have UACC(NONE) on all the profiles and PERMIT groups of allowed users READ access to the appropriate profiles. This should be reviewed with the security administrator.

Specify YES for LOGGING to turn on the LOG option on the EXEC CICS QUERY SECURITY.

---

*This level of access restriction is applied in addition to any security already in place for the CICS transaction through normal CICS customized security.*

---

In order to customize the above specification make the desired modifications to the to the XTR1S005 macro of the VBIUSAFP member, modify the JCL as directed in the comments at the top of the JCL and submit the job.

## Enhanced SSH support using Reflection for Secure IT, v 7.2

### Prerequisites:

\*Mainframe must have IBM Ported Tools for z/OS setup to decrypt anything coming in through a port (identified below in example) and forward it to another port.

### To take advantage of the enhanced SSH support:

1. Install Reflection for Secure IT, v.7.2 to access it's SSH Windows client.
2. Open a command prompt and issue the following command to secure the SSH tunnel to your mainframe:

```
C:> C:\Program Files\Attachmate\RSecure\ssh.exe -L port_number:hostname:xxxx
```

Where:

The port\_number is that of the port configured in the VBI client. And xxxx is dependent on how the user configured VBI within CICS

For example, `-L22318:abc.example.com:2318 -N user1@abc.example.com` tells the RSIT client to encrypt and reroute anything on the local box sent out over TCP port 22318 (this port number must match that set up in the VBI client) to the host at IP address, abc.example.com, and when the message arrives at the mainframe, decrypt it and send it to port 2318 (dependent on how VBI was configured within CICS) at the same IP address (abc.example.com) using the RACF user ID, user1.

## Server Customization (optional)

Customize the server if:

- Your client application expects to receive the CICS Good Morning message screen and the screen does not match the default.
- Your installation requires additional security based upon the ID Manager selected virtual terminal ID or virtual netname.
- You submitted runtime configuration customizations in the VBIUCFG member or the resource access restriction customizations in the VBIUSAFP member.

## Good Morning Message Customization

To customize the Good Morning Message (CSGM)

Update the XTR1S004 logo embedded in the VBIUSRVR job stream in the INSTLIB library to match the CICS logo screen used in your installation.

By default, if new 3270 connections in your local environment are greeted with a Good Morning Message rather than a security logon, a standard representation of the actual IBM CICS CSGM logo screen is displayed on the 3270 screen of the connecting client.

As shipped, each line of the logo in the VBIUSRVR job stream is 40 characters wide, and each logo is 11 lines long. At run-time, the Data Engine places this logo in the center of the 3270 screen of the connecting client. The placement varies based on the number of columns and lines of the particular terminal model.

The default is:

```
XTR1S004 CSECT

XTR1S004 AMODE 31

XTR1S004 RMODE ANY

        DC A(GMLINLEN)

        DC A(GMSCRLNS)

        DC A(GMSCRLEN)

*-----
* 24X80 DC'D AS 48 LINES OF 40 BYTE CONSTANTS COMPRISING
* THE WELCOME TO CICS SCREEN
```



```

*-----
GMSCREEN DS 0H

GMSCRLN1 DC CL40' *****\ *****\ *****\ *****\ '
GMSCRLN2 DC CL40' *****\ *****\ *****\ *****\ '
          DC CL40' **\|\|\|\|\**\  **\|\|  **\|\|\|\|\**\  **\|\|\|\|\**\ '
          DC CL40' **\  \|\  **\  **\  \|\  **\  \|\ '
          DC CL40' **\  **\  **\  *****\ '
          DC CL40' **\  **\  **\  *****\ '
          DC CL40' **\  **\  **\  \|\|\|\|\**\ '
          DC CL40' **\  **\  **\  **\  **\  **\  **\ '
          DC CL40' *****\ *****\ *****\ *****\ '
          DC CL40' *****\ *****\ *****\ *****\ '
          DC CL40' \|\|\|\|\  \|\|\|\|\  \|\|\|\|\  \|\|\|\|\ TM'

GMSCREND DC X'FF'

GMLINLEN EQU GMSCRLN2-GMSCRLN1

GMSCRLN EQU GMSCREND-GMSCREEN

GMSCRLNS EQU ((GMSCRLN/GMLINLEN)+1)

          END XTR1S004

```

Only alter the lines GMSCRLN1 to the line preceding GMSCREND to match your customized CICS logo; all other lines must remain unaltered for the runtime to support logo customization.

The one-line superimposed "Welcome to CICS..." message is customized by the CICS systems programmer in the CICS Systems Initialization Table (SIT), and shouldn't be added to the modified logo.

The Attachmate run-time 3270 Bridge Server harvests this customized string and superimposes it at run time, in a manner identical to that used by IBM.

## Complete Server Customization

To complete the server customization, update the VBIUSVRV job stream in INSTLIB and submit the job.

## CICS Web Services Interface (Optional, CICS TS V4)

You can configure the Verastream Bridge Integrator server to act as a CICS Web Services provider.

You may want to use this support if

- You are using CICS Transaction Server V4 or above.
- The performance benefit of persistent TCP connections is important to you.
- You want to connect to CICS using SSL or HTTP authentication.

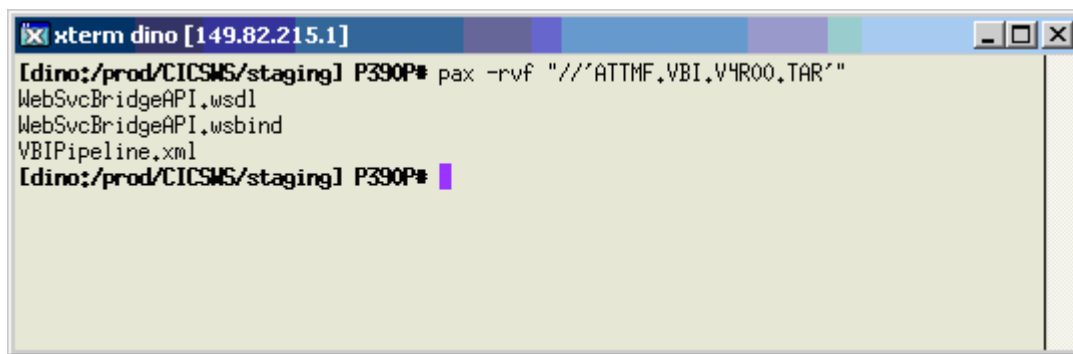
Verastream Bridge Integrator provides the WSDL that represents the VBI server's interface (via the interface program XTR1A005) that your middle-tier application can use as a Web Service Reference. There is a Java Bridge API jar (BridgeAPI.jar) and .NET Reference Object as assists. These provide a newer API that is more closely aligned with the SOAP message structure as well as an API that is compatible with the previous APIs.

To add support for the CICS Web Services Interface, perform the following actions:

Uncomment and complete the CICS Web Services Interface section of the VBICICSD before running the job to create the CICS definitions. Make changes reflecting your SSL and Authentication requirements. Refer to the section "Defining TCPIP SERVICE Resources for SSL" in the CICS RACF Security Guide for more information.

Extract the TAR dataset to a CICS staging directory in the Unix File System. For example, from a UNIX telnet session

```
cd /prod/CICSWS/staging
pax -rvf "'/'ATTMF.VBI.V4R00.TAR' "
```



When you **CEDA INSTALL** the group, CICS copies the **WSDL** and **WSBIND** to the **SHELF** and dynamically defines a **URIMAP** and **WEBSERVICE** in the CICS region. Refer to the *CICS Web Services Guide* for your CICS TS release for more information.

## Define Terminal Pool ID Manager Configuration VSAM File

To define the KSDS VSAM Terminal Pool ID Manager Configuration file, open the PDS member VBIVSAM edit the JCL by providing the parameters for the IDCAMS utility. You will need to provide the dataset name for the ID Manager for example 'ATM.IDMGR.TERMPPOOL' and the device on

which the ID Manager configuration file is to reside in your installation.

Once these parameters are defined, please provide the JCL parameters compliant with your installations standard for JOB card as described in the comments at the top of the JCL member save the modified JCL and submit the job.

## Set Up ID Manager Bridge Facility Pools

ID Manager Bridge Facility pools are set up using the 3270 ID Manager configuration transaction. The default transaction id for the ID Manager application is: 'IDMG'.

See the Terminal Pool ID Manager guide for information on using the terminal pool ID manager utility and instructions on migrating existing pools from VBI 2.0. The guide is available from the documentation folder at the root of the CD and on the product download site.

## Make Product Executables Available to CICS

- Copy all the members of the <&HLQ>.VBI.V4R00.SXT1LOAD LOADLIB to a library in the CICS DFHRPL concatenation  
or alternatively
- Add the LOADLIB to the CICS PROC/JCL DFHRPL concatenation and shutdown & restart the CICS region.

## Install the CICS Resource Definitions

Review and modify the job VBICICSD job stream in the INSTLIB library. Instructions are included as comments in the VBICICSD member.

If the Clean Up or Sample/IVP Transaction IDs were changed during installation, you must duplicate the change here as well as in the Converter customization process.

To use different Bridge Facility Model names than the IBM supplied CBRF, and if those different terminal definitions do not already exist, you can set up the definitions here. This must be coordinated with the Converter customization, above.

The following definitions are included.

TCPIPSERVICE	Port	URM	Transaction	Description
XTR1TCPS	2317	XTR1A000	CWXN	VBI Listening Port
XTR1TCWS		None	CWXN	CICS Web Services Listening Port

TRANCLASS	Max Active	Description
-----------	------------	-------------

XTR1CLAS	150	Cleanup transaction limit
----------	-----	---------------------------

Transaction	Program	Task Data Location	Task Data Key	Description
ATMC	XTR1A000	BELOW	USER	Cleanup
ATM1	XTR1A003		USER	Sample/IVP (Optional)
IDMG	XTR1A007		USER	ID Manager Pool Status Display (Optional)

---

*VBI also uses IBM supplied transactions CWXN and CWBA. Please ensure these transactions are installed with their IBM supplied default attributes. If you use a different Analyzer transaction than CWBA (see the Converter Customization step), copy the IBM supplied definition and give the copy the Transaction.*

---

Program	Language	Data Location	Exec Key	Description
XTR1A000	Assembler	Below	CICS	CICS Web Interface Analyzer
XTR1A001	Assembler	Below	CICS	Web XML Converter
XTR1A002	COBOL	Below	CICS	Server
XTR1A003	COBOL	Any	User	Optional Sample/IVP
XTR1A005	COBOL	Any	User	Optional Web Service Interface
XTR1A006	COBOL	Any	User	Opt. Web Service ConnectBridge Auth Handler
XTR1A007	COBOL	Any	User	Optional ID Manager Pool Status Display

Mapset	Description
XTR1M30	Optional Sample/IVP Maps
XTR1M40	Optional ID Manager Pool Status Display Maps

Terminal	Netname	Typeterm	Description
CBR2	CBR2	DFHLU2M2	Terminal MODEL for Model 2
CBR3	CBR3	DFHLU2M3	Terminal MODEL for Model 3
CBR4	CBR4	DFHLU2M4	Terminal MODEL for Model 4
CBR5	CBR5	DFHLU2M5	Terminal MODEL for Model 5

The TYPETERMS referred to by these terminal definitions are from the IBM supplied DFHTYPE group.

Submit the job and verify its successful execution, then log on to CICS and use the CEDA transaction to install the Group.

## Internationalization: Model terminal implications

### Customization for code page support

If you are using international code page support, make sure that the CICS definition for the model terminal (referenced by the 'FacilityLike' parameter of connection API and configuration) references a TYPETERM definition with the corresponding international character set references.

In the following example the terminal model definition CBRG references a German language codepage TYPETERM named DFHLU2GE. Please see the CICS resource definition guide for more information.

```
CEDA EXP G(*) TERMINAL(CBRG)<enter>

OBJECT CHARACTERISTICS                                CICS RELEASE = 0640
CEDA View TErminAl( CBRG )
TErminAl      : CBRG
GrOuP        : GGROUPE

Description : DEFAULT 3270 BRIDGE TEMPLATE FOR GERMAN LANGUAGE

... ..

TERMINAL IDENTIFIERS
TYpeterm    : DFHLU2GE

.. ..

CEDA EXP G(*) TYPETERM(DFHLU2GE)

OBJECT CHARACTERISTICS                                CICS RELEASE = 0640
CEDA View TYpeterm( DFHLU2GE )
TYpeterm    : DFHLU2GE
```

```

Group           : GGROUP
DEscription     : CODE PAGE 1141 (GERMAN)
RESOURCE TYPE
DEvice         : LUTYPE2
TERmodel       : 2

... ..

CGcsqid        : 00695 , 01141      0-65535

```

## TCTUA Handling (Optional)

If you are using the TCT User Area (TCTUA) in your installation, the module XTR1E000 must be installed. If TCT User Area is not being used, then there is no need to install this Global User Exit

If your installation uses the TCTUA a facility model with User Area defined is also required. Refer to the Converter customization, above.

The CICS Global User exit XTR1E000 must be installed in every region that contains application transactions that you are going to contact via the Bridge. This usually means AORs, but if you have target applications in your TOR/routing region, then XTR1E000 must be installed there as well.

The other Data Engine components only need to be in the routing region.

As a result, XTR1E000 and the Bridge product programs are independent in that they can (and probably should) be in different groups. Since XTR1E000 is only necessary if you are accessing applications that make use of TCTUA, group and program definition is not included in the sample VBICICSD.

In the SIT, have an entry for PLTPI=*xx* where *xx* is the suffix identifying the list of programs you want to run at Post Initialization. For example, at our site suffix=01 so PLTPI=01.

In a PDS of members containing our PLTs we have a member named DFHPLT01. In member DFHPLT01 there is a line:

```
DFHPLT  TYPE=INITIAL,SUFFIX=01
```

then among the program lines there, we add a line:

```
DFHPLT  TYPE=ENTRY,PROGRAM=XTR1E000
```

After adding the exit program into the member, assemble the list into the appropriate library in the DFHRPL for the region. When CICS comes up, it will execute this list of programs.

The GLUE needs to be RACF authorized to execute.

## About CICS Multi-Region Operations (MRO)

*For instructions on how to configure MRO see the CICS TS Resource Definition Guide. Your systems programmer may have more information.*

VBI can execute in a CICS MRO environment. Install the VBI and ID Manager components in a Terminal Owning Region (TOR) with access to the z/OS load libraries that contain all CICS application BMS mapsets used by the customer's applications within the customer's CICS Application Owning Regions (AOR).

When defining MRO connections via the CEDA resource definition transaction we typically use the "define CONNECTION" security settings that include:

```
Attachsec      :      Identifying
BINDSecurity   :      No
Usedfltuser    :      No
```

If you are unable to get Bridge communication over MRO links to work: Define "CONNECTION" using CEDA with "ATTACHSEC=IDENTIFY", and "Bind User" and "Default User" set to "NO".

CEDA Object Characteristics	Description
CONnection:	CDS1
Group:	INSGRP
Description:	MRO CONNECTION TO CDESTIN1

Connection Identifiers	Description
Netname:	CDESTIN1
INDsys:	

Remote Attributes	Description
REMOTESYSem:	
REMOTEName:	
REMOTESYSNet:	

Connection Properties	Description
ACcessmethod:	Xm Vtam   IRc   INDirect   Xm

Protocol:	Appc   Lu61   Exci
Conntype:	Generic   Specific
SIngleless:	No No   Yes
DAstream:	User User   3270   SCs   STRfield   Lms
RECORDformat:	U U   Vb
QueueLimit:	No No   0-9999
Maxqtime:	No No   0-9999

Operational Properties	Description
AUTOconnect:	Yes No   Yes   All
INService:	Yes Yes   No

Security	Description
SEcurityname:	
ATTachsec:	Identify Local   Identify   Verify   Persistent   Mixidpe
BINDPassword:	PASSWORD NOT SPECIFIED
BINDSecurity:	No No   Yes
Usedfltuser:	No No   Yes
Recovery	Description
PSrecovery:	Sysdefault   None
XInaction:	Keep Keep   Force



## Getting Started Customizing the Data Engine

### Installation Checklist

You can use the following checklist as an overview. An additional summary is available in the `<&HLQ>.VBI.V4R00.INSTLIB($INSTALL)` PDS member.

In each JCL member resolve all of the environmental issues listed at the top of each JCL member before submitting the JCL for execution.

1. Upload the `<&HLQ>.vbi.v4r00.instlib.xmit` file from the Server directory to the host in binary 80 column Fixed format. See page 13 for more information.
2. RECEIVE INDSN('<&HLQ>.VBI.V4R00.INSTLIB.XMIT') dataset('<&HLQ>.VBI.V4R00.INSTLIB')  
See page 14 for more information.
3. EXEC 'ATTMF.VBI.V4R00.INSTLIB(\$INSTREX)'. This unpacks the needed PDS files to continue the installation. See page 15 for more information.

If the default configuration is acceptable for your installation, skip to step 9.

4. Modify the Analyzer customization JCL and configuration macro in PDS member `<&HLQ>.VBI.V4R00.INSTLIB(VBIUANLZ)` and submit. See page 19 for more information. If the configuration for the converter and server are acceptable, skip to step 9
5. Modify and submit the configuration JCL and macro shared by the Converter and Server: `<&HLQ>.VBI.V4R00.INSTLIB(VBIUCFG)`. A return code of 4 is acceptable
6. If you performed this step, continue with step 6, if not, you can skip to step 7.
7. Modify PDS member `<&HLQ>.VBI.V4R00.INSTLIB(VBIUCNVT)` and submit. This links the new configuration options into the runtime load module for the Converter.
8. To restrict system access by verification of USERID/PASSWORD credentials, skip to step 8. Continue with this step, if you want to restrict access to resources by resource ID or User Written access control.
9. Modify the resource access restriction template to be linked into the server. You must supply either resource class information or the name of a User written application program to be called to authorize an individual bridge execution request. Modify PDS member `<&HLQ>.VBI.V4R00.INSTLIB(VBIUSAFP)` to specify the resource protection desired. Modify the JCL and submit the job.

If you performed this step you must continue with step 8, otherwise skip to step 9.

- o Link the server to include all customization from steps 5 and 7. Modify PDS member `<&HLQ>.VBI.V4R00.INSTLIB(VBIUSRVR)` to customize the Good Morning screen

embedded in this member and link all prior customizations into the run-time server load module.

- Update the CICS definitions either by modifying and running PDS member <&HLQ>.VBI.V4R00.INSTLIB(VBICICSD) or by interactively defining the CICS resources using the CEDA transaction. Make sure to add the XTR1ATFP file definition to your CICS startup JCL.
  - Define the KSDS VSAM File to be used for Terminal Pool
  - ID Management XTR1ATFP Configuration. Modify and submit
  - PDS member <&HLQ>.VBI.V4R00.INSTLIB(VBIVSAM).
10. Copy all load modules from <&HLQ>.VBI.V4R00.SXT1LOAD to a LOADLIB accessible to the target CICS region. Define the:
- DFHBRNSF file for the IBM Linkable Bridge. Newcopy all
  - Load modules or Bring the CICS region down and back up
  - Run the ID Manager Terminal Pool transaction IDMG to define the pools of 3270 Bridge terminals to use.