Installation and Configuration Guide

attachmate®

Verastream® Transaction Integrator

Table of Contents

System Requirements

To install the Transaction Designer, you need:

Requirements for	Supported
Operating System	Windows Server 2003
Processor	500 MHz minimum for Windows
RAM	256 MB
Disk Space	230 MB
Application Server	Microsoft Visual Studio .NET Framework 2003 to generate .NET Web services

To install the Runtime Service, you need:

Requirements for	Supported
Operating System	Windows Server 2003 Server Solaris v. 9 or 10 RedHat Enterprise Linux 5.0
Processor	Intel single or multiple processor RISC single or multiple processor 1.2 GHz suggested for Windows 500 MHz minimum for Windows 450 MHz for Solaris
RAM	512 MB
Disk Space	30 MB for CICS 30 MB for IMS
Java	Sun Java 2 JRE v. 1.6.0_01 (installed automatically on Windows platforms only)

- Uninstall legacy Synapta Services Builder for CICS (IMS) before you install Transaction Integrator. Serious problems may occur otherwise.
- The Java platform does not support 16-color displays. Any component that requires Java will require at least 256-color support.
- The minimum screen resolution setting is 1024 x 768 pixels.
- Host connections via the SNA protocol are not supported.

Overview

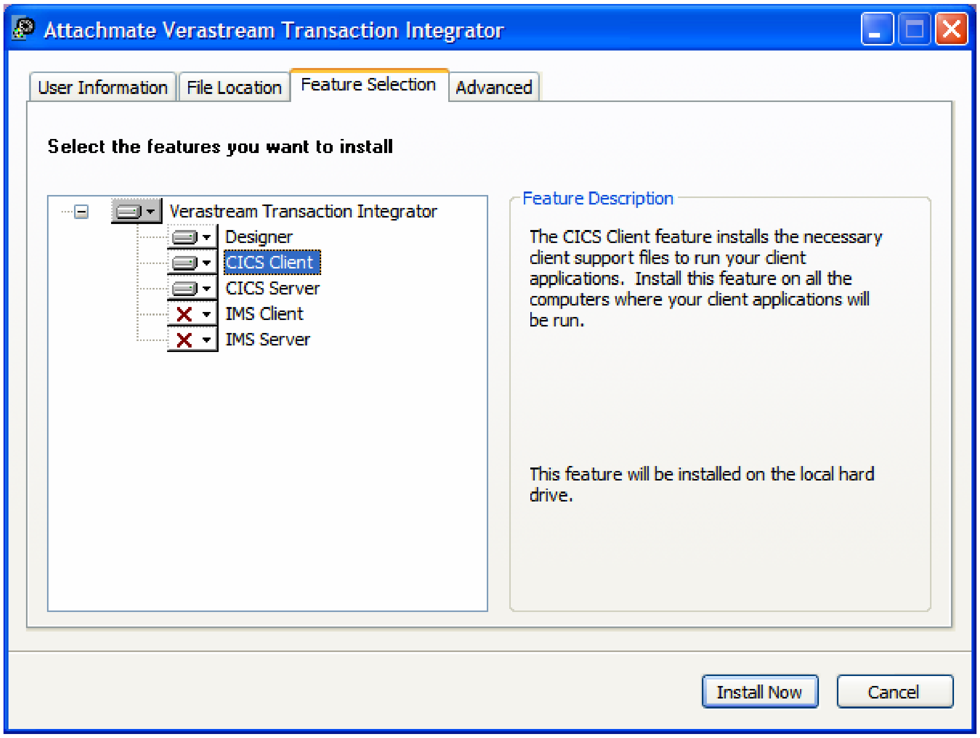
These installation and configuration procedures assume that you are installing all three Transaction Integrator components: Designer, Server, and Client for either your CICS or IMS host. Each of these components can be installed in different locations and independently.

To avoid serious problems, uninstall Synapta Services Builder for CICS (IMS) before you install Transaction Integrator. While the co-existence of these products on the same computer is not supported, uninstalling does not remove existing data files. Legacy Synapta task definitions, host program definitions, and copybooks will still be available to import.

Installing Transaction Integrator

The Transaction Designer can only be installed in a Windows environment. You can install the runtime server and client files in either a Windows or Unix environment. To install multiple servers, simply repeat these instructions for each server.

NOTE: If you are installing multiple runtime servers for use with IMS, a unique system id is required for each server. System ids are specified in the server configuration file. See Configuring Your Runtime Environment section of this document for information.



Select the features you want to install

The CICS Client feature installs the necessary client support files to run your client applications. Install this feature on all the computers where your client applications will be run.

This feature will be installed on the local hard drive.

To install on Windows

1. From the distribution or CD images, click Install.exe and follow the installation program instructions.

2. (Optional) To personalize the installation, click the User Information tab and enter the name and organization.

3. (Optional) To change the default installation folder, click the File Location tab and browse to the folder in which you want to install Transaction Integrator.

4. Click the Feature Selection tab to specify which features of the Transaction Integrator you want to install. These features can be installed on separate computers or on a single computer.

5. (Optional) To change the type of installation, click the Advanced tab. By default, the Transaction Integrator features are installed to the local computer.

Install this	To do this
Designer	Create host transactions, edit copybooks, and produce client operations. Installs Web Builder to generate interface objects.
CICS or IMS Client	Installs the necessary client configuration files to run client applications accessing either IMS or CICS host types. Client files must be installed on the same computer as the designer if you are generating .NET Web services.
CICS or IMS Server	Install the runtime engine to provide host access for your client operations.

NOTE: To connect the runtime engine with a target CICS system on the mainframe using TCP/IP, you must first install the Attachmate TCP/IP connectivity feature on the mainframe. This feature runs as a server inside the CICS Web Support framework and sends HTTP over TCP/IP into CICS, providing the functionality necessary for the runtime engine to communicate with CICS. The Transaction CICS-TCPIP Connectivity Guide (pdf) walks you through this process.

To install on Unix

From the distribution or CD images, click either `vti_cics_4.0.tar.gz` or `vti_ims_4.0.tar.gz` depending on your host type and follow the installation program instructions.

1. FTP the tar.gz files to the target Unix computer in binary mode.
2. After you place the files in a staging area, open them using:

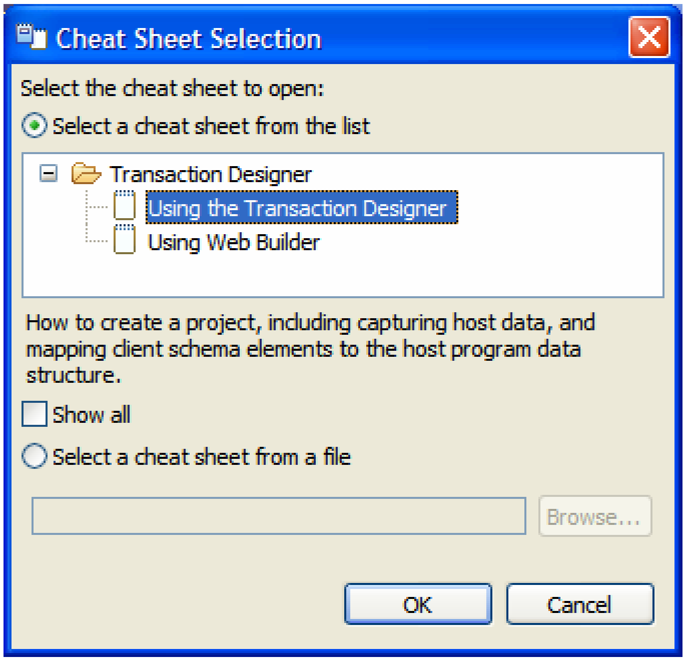
   ```
   Gunzip vti_cics_4.0.tar.gz
   ```

   This creates the file, `vti_cics_4.0.tar`.

3. Extract the .tar file with the command "`tar xvf vti_cics_4.0.tar`". This produces the directory, `cics-4.0`, which contains all the files for the CICS setup.
4. From `cics-4.0/install` (you can get to this location using the cd command), enter chmod `775 *.sh` to work with the shell scripts. (alternatively, you could use chmod `+x *.sh`).
5. Enter sh `setup.sh` to begin the setup process.

After you install the Designer

The first time you open the Designer, you can walk through the Using the Transaction Designer tutorial which is accessible through the Help menu. Click Cheat Sheets > Using the Transaction Designer.



Cheat Sheet Selection

Select the cheat sheet to open:

● Select a cheat sheet from the list

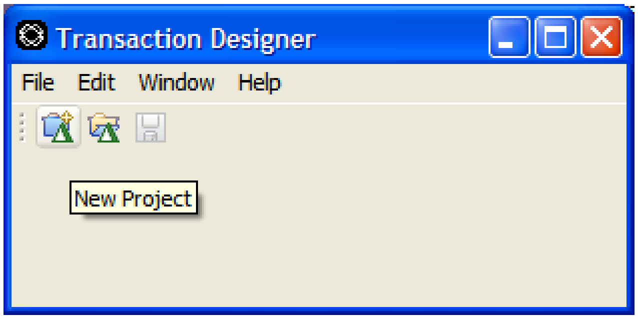
- Transaction Designer
  - Using the Transaction Designer
  - Using Web Builder

How to create a project, including capturing host data, and mapping client schema elements to the host program data structure.

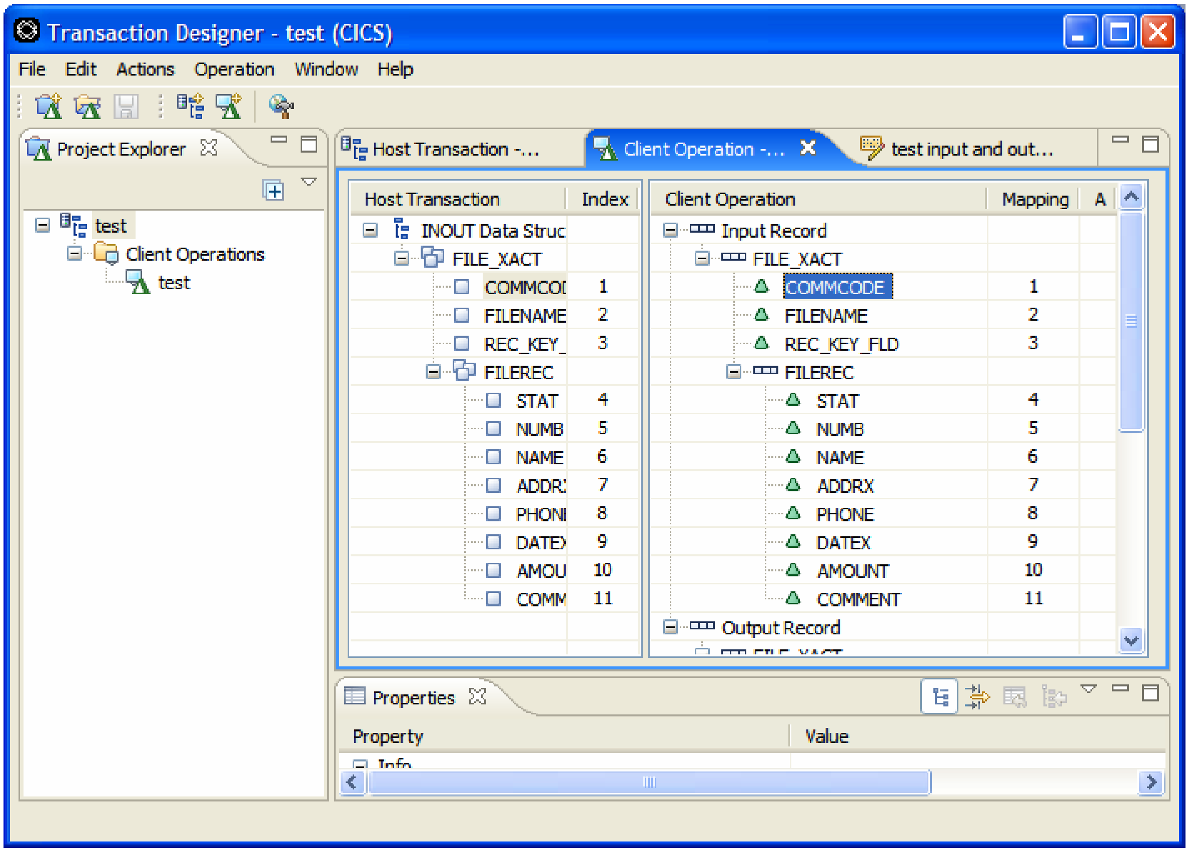
□ Show all

○ Select a cheat sheet from a file

The Transaction Designer is project-based.



1. Open a new project. Click the New Project icon on the toolbar or File > New Project.

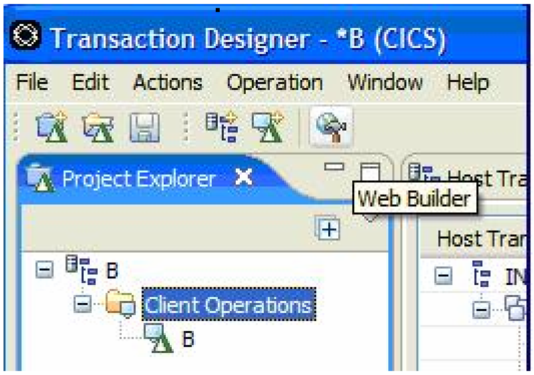


Host Transaction	Index
INOUT Data Struc	
FILE_XACT	
COMMCOI	1
FILENAME	2
REC_KEY_	3
FILEREC	
STAT	4
NUMB	5
NAME	6
ADDR	7
PHONI	8
DATE	9
AMOU	10
COMM	11

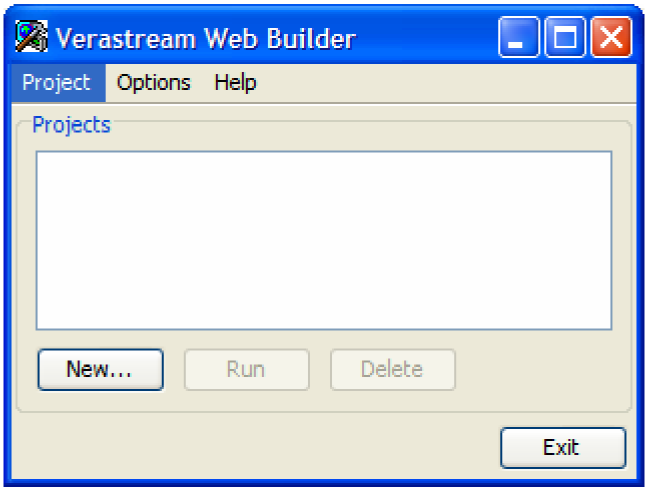
Client Operation	Mapping	A
Input Record		
FILE_XACT		
COMMCODE	1	
FILENAME	2	
REC_KEY_FLD	3	
FILEREC		
STAT	4	
NUMB	5	
NAME	6	
ADDRX	7	
PHONE	8	
DATEX	9	
AMOUNT	10	
COMMENT	11	
Output Record		

2. Declare a Host Transaction, including importing the selected copybook and its import and output layout.

3. Create a client operation. The client operation contains the client schema that describes how clients will interact with your host transaction project. The client schema has one input element and one output element which reference a data structure, defining the mapping of the input and output parameters between the interface and the data structure.



4. Open Web Builder. Click the Web Builder icon on the toolbar or choose Web Builder from the Actions menu. Web Builder uses the client operation to build a reusable interface object.



5. Create a new Web Builder project. Select your projects type, either Java or .NET, and from the drop down list associate the Web Builder project with the Transaction Designer project.

Configure Your Runtime Environment

IMPORTANT: You must configure your runtime environment and start the Runtime Service before attempting to provide host access for your client applications.

The runtime environment includes the mainframe computer (running either CICS or IMS), the runtime server and the client computers. The runtime server and client components can be configured on one server. If you are serving your client applications from an application server, this is considered the "client", or you can have separate servers, and run your client application on separate client computers.

Workflow

The Task Integrator runtime supports TCP/IP host connections and requires configuration. Typically, to configure the runtime environment correctly, regardless of your network configuration, you need to take the following steps:

- Open the system configuration file and make the necessary modifications to provide host access to your client operation.

- Ensure that your client application has access to the correct JAR files by installing the client feature for either CICS or IMS host type.

- Ensure that the class path is set correctly on the computer on which the client application is running. The complete list of the JAR files to add is available in the Configure Client Access to Runtime Servers section of this document. You can also access the samples under the CICSClient or IMSClient directories. Open `run.sh` to see an example of the JAR files used in the client-side of the runtime environment.

- Configure the client configuration file used by the client to access the runtime server.

- Optional. Configure client properties to provide logging, tracing, and other capabilities.

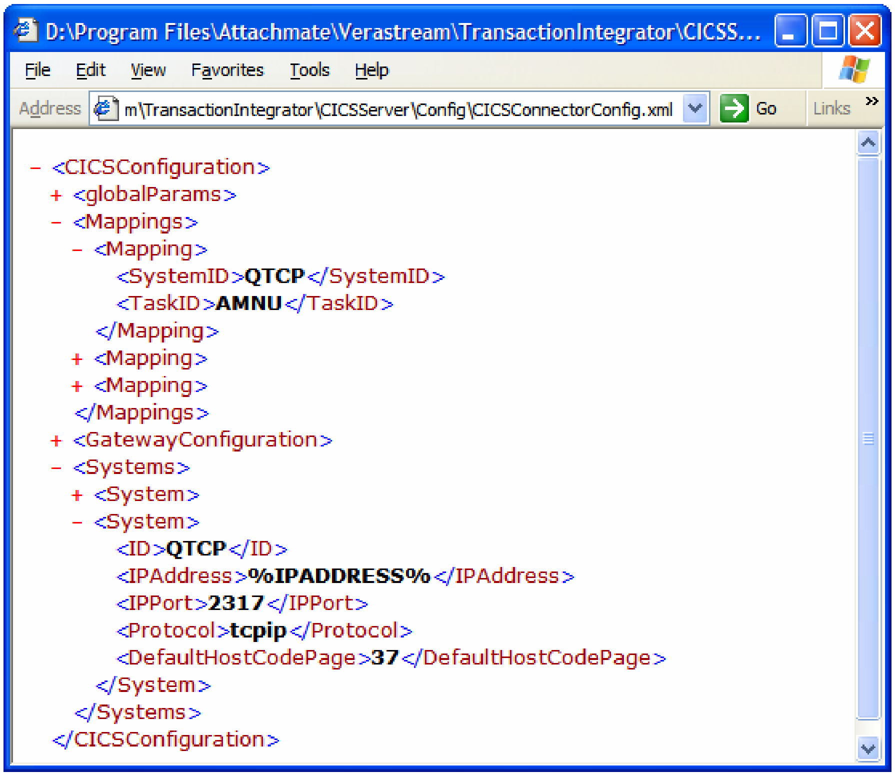
- Test your system by running the samples provided. To test, we will assume that both the runtime and client options have been installed to a single computer. This, of course, is not necessary and you may want to install the client files, as well as the designer, on separate computers.

NOTE: The examples reference a CICS host type. IMS instructions are the same except where noted.

Configure the host connection

From the `Attachmate\Verastream\TransactionIntegrator\CICSServer\Config` directory, open `CICSConnectorConfig.xml` in a text editor. In IMS, the file name is `IMSServer\Config\IMSConnCfg.xml`.

Configuration includes defining a host connection (referred to as a "System") and then creating mapping parameters to support the client operation. This example is a default CICS configuration file.


D:\Program Files\Attachmate\Verastream\TransactionIntegrator\CICSS...
File
Edit
View
Favorites
Tools
Help
Address
m\TransactionIntegrator\CICSServer\Config\CICSConnectorConfig.xml
Go
Links

```
- <CICSConfiguration>
  + <globalParams>
  - <Mappings>
    - <Mapping>
        <SystemID>QTCP</SystemID>
        <TaskID>AMNU</TaskID>
      </Mapping>
    + <Mapping>
    + <Mapping>
    </Mappings>
  + <GatewayConfiguration>
  - <Systems>
    + <System>
    - <System>
        <ID>QTCP</ID>
        <IPAddress>%IPADDRESS%</IPAddress>
        <IPPort>2317</IPPort>
        <Protocol>tcpip</Protocol>
        <DefaultHostCodePage>37</DefaultHostCodePage>
      </System>
    </Systems>
  </CICSConfiguration>
```

Edit the server configuration file using the appropriate values for your system and mapping parameters.

<System> parameters include:

- ID –(CICS) The name of the <system>definition. It is used in the Mappings section of the CICS configuration file to indicate the host CICS system (represented by a System definition) that processes each execution of a client operation matching the <TaskID> of that mapping. ID can be any alphanumeric name that is unique within the <System> tags of the current CICS configuration file.

- ID – (IMS) The ID is a 1 to 4 character alphanumeric name that must be unique across the environment of all VTI IMS servers in your environment. The ID is used by the IMS server as a prefix to a unique client request identifier to the host environment that proceeds each client operation.

- Datastore ID(IMS Only) – the IMS Connect datastore value for the IMS system to which the target IMS Connect will connect in order to execute IMS transactions.

- IP Address – the Internet Protocol address of the CICS Web interface or IMS Connect on the target mainframe.

- IP Port – the port at which the CICS host component is installed or of the target IMS Connect.

- Protocol – TCP/IP is the supported protocol

- Default Host Code Page – the number of the code page being used by the host program. See Host Code Pages for a complete list of supported code pages.

<Mapping> (CICS only) parameters include:

- System ID – An internal identifier referring to the definitions necessary to configure a TCP/IP connection to a particular CICS subsystem.

- Task ID –the name of the client operation. Use the default value, AMNU, to run the provided sample. In an IMS configuration file, client operations are mapped using task id. The <TaskID> entries are the client operations associated with (mapped to) the <System> definition they are associated with.

<globalParams>

Optional. Set the parameters to "true" to log errors, warnings, and information. This information is logged to:

- In a Windows environment:

```
<logfile>C:Documents and Settings\user_name\My
Documents\cicsevents.log</logfile>
```

```
<tracefile>C:Documents and Settings\user_name\My
Documents\cicsevents.log</tracefile>
```

or

- In a Unix environment:

```
<logfile>/home/user_name/log</logfile>
```

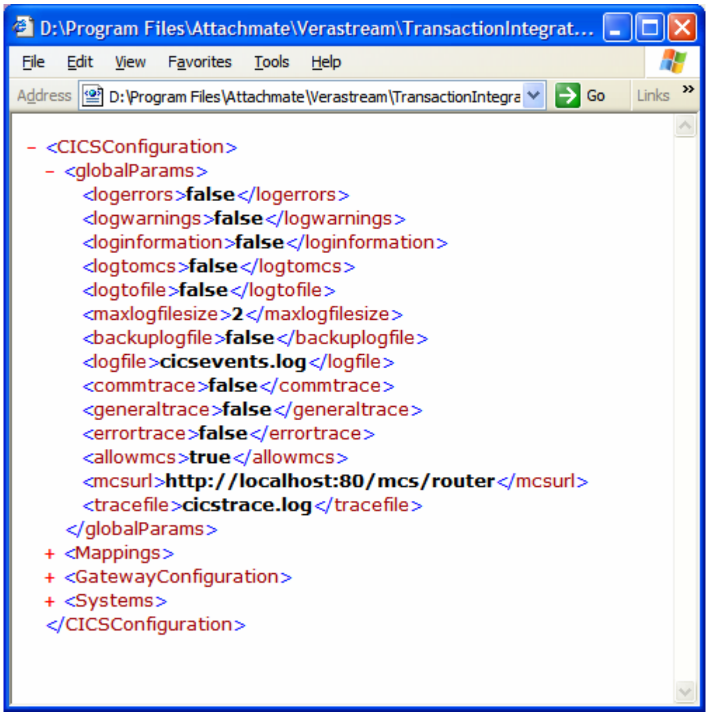
```
<tracefile>/home/user_name/log</tracefile>
```

Where:

Logfile = location of logging events

Tracefile = location of tracing

This example illustrates where you can put the file, however, in both cases, a file with that name is created in the current directory of the running server; CICSServer (or IMSServer for IMS) by default.


D:\Program Files\Attachmate\Verastream\TransactionIntegrat...
File
Edit
View
Favorites
Tools
Help
Address
D:\Program Files\Attachmate\Verastream\TransactionIntegra
Go
Links

```
- <CICSConfiguration>
  - <globalParams>
      <logerrors>false</logerrors>
      <logwarnings>false</logwarnings>
      <loginformation>false</loginformation>
      <logtomcs>false</logtomcs>
      <logtofile>false</logtofile>
      <maxlogfilesize>2</maxlogfilesize>
      <backuplogfile>false</backuplogfile>
      <logfile>cicsevents.log</logfile>
      <commtrace>false</commtrace>
      <generaltrace>false</generaltrace>
      <errortrace>false</errortrace>
      <allowmcs>true</allowmcs>
      <mcsurl>http://localhost:80/mcs/router</mcsurl>
      <tracefile>cicstrace.log</tracefile>
    </globalParams>
  + <Mappings>
  + <GatewayConfiguration>
  + <Systems>
  </CICSConfiguration>
```

Test your connection

Assuming that you have completed the previous instructions, including modifying the server configuration file and starting the runtime server, use the samples provided to verify the configuration file is accurate and your connection is working.

1. From `<host type>Client\samples\AMNU`, open either the ITask or IConnectorAccess directory, depending on what type of interface object you want to run.

2. Depending on whether you are in a Windows or Unix environment, open `run.bat` or `run.sh` in a command prompt window or a console.

3. Supply your username and password. Read the comments provided and follow the instructions to verify that your host connection is working and you are receiving data output.

Configure Client Access to Runtime Servers

If you are planning on running your applications from the same server where you installed the runtime service, then, after you set the classpath, no other configuration is needed.

If, however, you plan to run your applications on a different computer, you need to provide the necessary client support files for your environment on all the computers you will use. The CICS or IMS Client option, available from the install program, installs a CICS or IMS Client directory containing everything you need to run client operations.

Editing your classpath

In addition to installing the client support files, you need to add the following client support files to your Java classpath.

In the directory …\Attachmate\Verastream\TransactionIntegrator\<host type>Client:

- CICSClient.jar
- xerces.jar
- common.jar
- MetaData.jar
- util.jar

In the directory …\Attachmate\Verastream\TransactionIntegrator\Common\mcs:

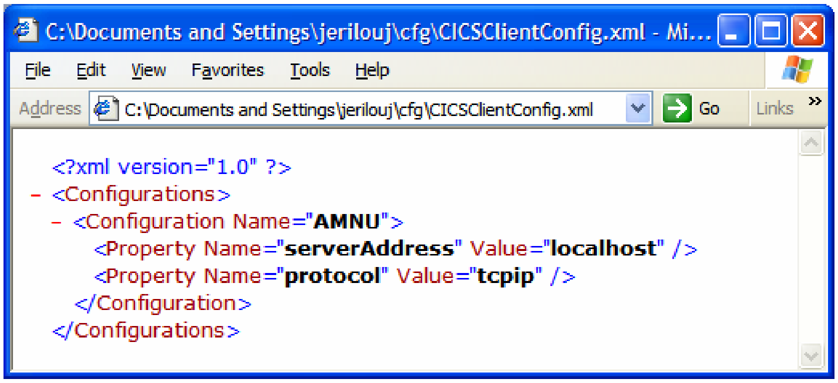
- mcsagent.jar
- mcs-rpc.jar

Configure the Client Environment

Client access to the runtime servers is handled through the <host type>ClientConfig.xml file, which is created automatically the first time a client operation is run on a remote or client system.

The <host type>ClientConfig.xml file is located in the cfg directory, either in the location you specified as a “System” property (Attachmate.home) or in the Java-provided location, user.home. On a Windows system, this is Documents and Settings\user login\cfg.

An example of a default client configuration file


C:\Documents and Settings\jerilouj\cfg\CICSClientConfig.xml - Mi...
File Edit View Favorites Tools Help
Address C:\Documents and Settings\jerilouj\cfg\CICSClientConfig.xml
Go
Links

```
  <?xml version="1.0" ?>
- <Configurations>
  - <Configuration Name="AMNU">
      <Property Name="serverAddress" Value="localhost" />
      <Property Name="protocol" Value="tcpip" />
    </Configuration>
  </Configurations>
```

Where:

serverAddress specifies the location of the runtime server (default location: "localhost"). The default RMI port number for the CICS server is 6800, for the IMS server, 6900. The protocol is TCPIP.

To change the port number, add the port number after the server address as follows, where *xxxx* is the port number:

<Property Name = "serverAddress" Value = "*localhost:xxxx*"/>

The value you specify must match the value found in `CICSServer\cics_runtime.properties` or `IMSServer\ims_runtime.properties`.

Editing the client configuration file to identify remote applications

Open the `<host type>ClientConfig.xml` in a text editor and modify as follows:

- To specify the location of a remote runtime server to serve applications from the processing runtime server, type the server's address as follows:

  ```
  <Property Name = "serverAddress" Value = "<DNS Name> or
  <IPAddress>"
  ```

To specify a custom directory for the client configuration file

Set the System property "Attachmate.home" on the command line that starts the Java Virtual Machine that is running your interface objects as follows:

```
-Dattachmate.home=<c:\mydirectory>
```

Where:

```
<mydirectory>= the name of the directory you are specifying
```

How you pass command line arguments to the Java VM depends on your environment. The Java VM can be part of a Web server for Servlets and JSPs, in a browser for Java Applet, or a stand-alone process if your interface objects are running as part of a Java application.

Set the System property programmatically from within your application as follows:

```
  System.setProperty(“attachmate.home”,<myDirectory>);
```

Your application must make the call before you run interface objects, and it must have the necessary permissions to modify system properties.

Configure Logging and Tracing

As an optional feature, you can set up logging and tracing, both on the server and the client-side.

- Server configuration – In the `CICSConnectorConfig.xml` or `IMSConnCfg.xml` file, depending on your host type, is the <globalParams> parameter. This is where logging and tracing options are set. See Configure Host Connection for more information.

- Client configuration – You can configure client logging and tracing in the `<host type>client.properties` file, located in the start-up directory of your Java Virtual Machine.

Start and stop the Runtime Service

After installing and configuring the runtime server, you must start the service.

In Windows, you can run the service either as a Windows service or in application mode. In UNIX, you can run the service as a daemon in the background.

In a Windows environment

- To run as a service

  From the Control Panel, choose Services, select Runtime Service (for CICS or IMS), then choose Start or Stop. By default, the service starts automatically whenever you start the Runtime Service computer.

- To run in application mode

  From the Start menu, choose Attachmate Verastream > Transaction Integrator>Runtime Service. A command prompt window displays with the message that the Runtime Service was successfully started. All console messages are recorded in the ServiceLog.txt file located in the IMS or CICS Server directory. To stop the Runtime Service, press CTRL + C

In a UNIX environment

From the following directory, for both root and non-root users, use the run-time wrapper script to start, stop, and restart the daemon as indicated:

```
…\Attachmate\verastream\transaction
integrator\<host_type>Server\<host_type>runtime.sh
```

Use this command	To do this
cicisruntime.sh start	Starts the runtime service, which starts in the background and writes its process ID to CICSServer/cicsruntime.pid, and logs output to CICSServer/log/runtime.log
./cicisruntime.sh stop	Stop the runtime service
./cicsruntime.sh restart	Restart the runtime service
./cicsruntime.sh status	Obtain runtime service status

Verify Your Installation

To verify that your installation and configuration are operational, there are samples provided with the Transaction Integrator. The samples include client operations designed and configured to access the standard IBM provided "Installation Verification" transaction available within CICS (program DFH$AXCS) and IMS (transaction IVTNO) on the host.

Update host security credentials for the intended host system, and invoke the samples using run.bat or run.sh on UNIX. If the host configuration is complete and appropriately provided in the runtime server configuration, sample data from the host will be displayed in the client samples.

All sample support files used in these examples are located in **`…\Verastream\TransactionIntegrator\<host_type>Client\samples`**

Create an interface to a program on the host and deploy it to a remote client

Assumptions:

- If you did not install the client and server features on the same computer, you have updated the client configuration as described in Editing the Client Configuration file to Identify Remote Applications..
- The example instructions will use a CICS host type, but can be used with IMS, with minor host-specific changes.
- You have followed the directions in this guide and have configured the runtime server.

Components you will use:

- Transaction Designer
- Runtime and Client Features
- Web Builder

Steps:

1. Specify which program on the host you will be using
2. Indicate the copybooks that define the host program data structure
3. Create the Client Operation you will use to build the interface object
4. Define the mapping of the input and output parameters between the host transaction and the client operation
5. Create the interface object from the client operation
6. Deploy and use the interface object to develop client applications

Step One: Specify which program on the host you will be using

In the Designer

1. Create a new project called Sample.
2. Accept the default project location and select CICS or IMS as the Transaction Type. Click OK.

Step Two: Select the copybook you will use to build the host transaction

1. Choose New Host Transaction from the Actions menu, or from the right-click menu in the Project Explorer frame.
2. Type *sample_transaction* in the Name field of the Declare a Host Transaction, dialog box. If you'd like you can enter a description of the new host transaction.
3. For CICS, type program name, DFH$AXCS in the Program Name field. For IMS, leave the program name field blank.
4. For CICS, select *One shared data structure for both input and output*. For IMS, select *Two separate data structures*. The new host transaction must match the host program data structure. Click Next.
5. Import the copybook you will use to redefine the Host Program Data Structure.
   a. Click Add and for CICS, browse to `…\Verastream\TransactionIntegrator\CICSClient\samples\AMNU\copybooks` For IMS, `…\Verastream\TransactionIntegrator\IMSClient\samples\IVTNO\copybooks`
   b. For CICS, open `AMNU.cbl`. For IMS, use `IVP_in.cbl` for input and `IVP_out.cbl` for output. Click Next.

Step Three: Create the Client Operation you will use to build the interface object

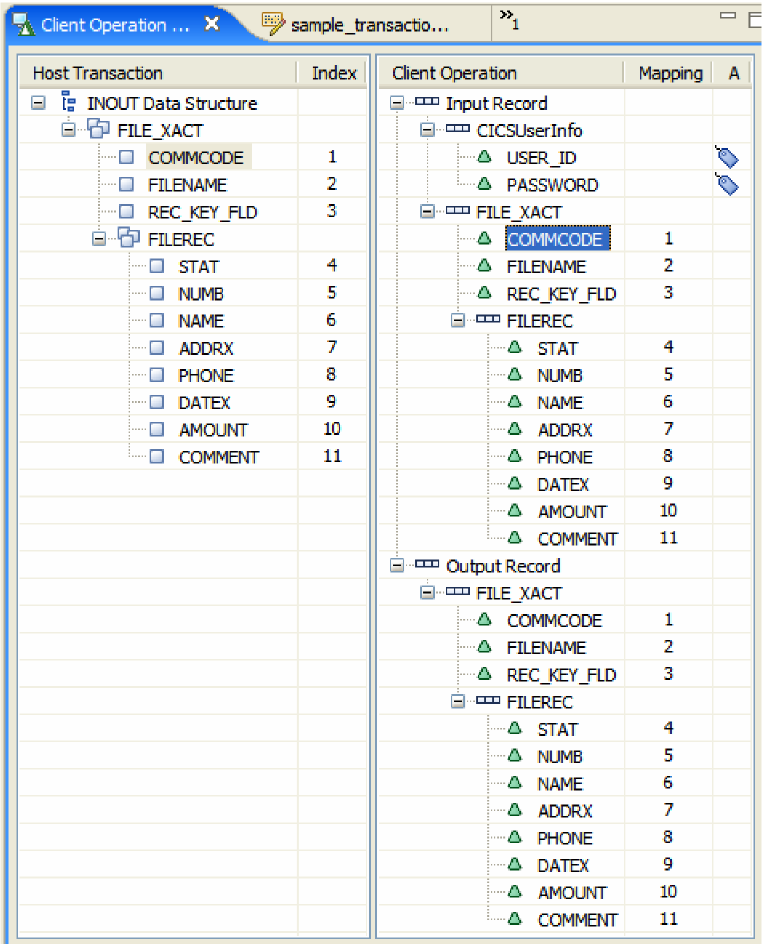
1. Create a new client operation. Enter *sample_client_op* in the Name field.

   By default, an initial client operation is created with elements from each data item in the host program data structure. This is the option we will use for this example. However, you can choose to create an empty client operation that you can populate manually.

2. If required for access to host transactions in your environment, add input elements for the security attributes Username and Password. In the IMS examples, these elements and the additional IMS security element, RACF Group are not shown. These attributes allow the client to supply the operation with valid security credentials while accessing the host program data. Click Finish.

Step Four: Define the mapping of the input and output parameters between the host transaction and the client operation

1. Review the copy of the AMNU.cbl in the Copybook Editor pane. The Copybook Editor gives you the opportunity to modify and correct any discrepancies in the original host program copybook. In our example, we will accept the copybook as is.
2. Select the Client Operation Editor tab. In this view the host transaction and client operation default mapping are visible and can be edited.
3. The Host Transaction data structure Index values map to the Client Operation basic elements. For example,



Host Transaction	Index
INOUT Data Structure	
FILE_XACT	
COMMCODE	1
FILENAME	2
REC_KEY_FLD	3
FILEREC	
STAT	4
NUMB	5
NAME	6
ADDRX	7
PHONE	8
DATEX	9
AMOUNT	10
COMMENT	11

Client Operation	Mapping	A
Input Record		
CICSUserInfo		
USER_ID		🏷
PASSWORD		🏷
FILE_XACT		
COMMCODE	1	
FILENAME	2	
REC_KEY_FLD	3	
FILEREC		
STAT	4	
NUMB	5	
NAME	6	
ADDRX	7	
PHONE	8	
DATEX	9	
AMOUNT	10	
COMMENT	11	
Output Record		
FILE_XACT		
COMMCODE	1	
FILENAME	2	
REC_KEY_FLD	3	
FILEREC		
STAT	4	
NUMB	5	
NAME	6	
ADDRX	7	
PHONE	8	
DATEX	9	
AMOUNT	10	
COMMENT	11	

Note the security attribute icons identifying the user_id and password elements.

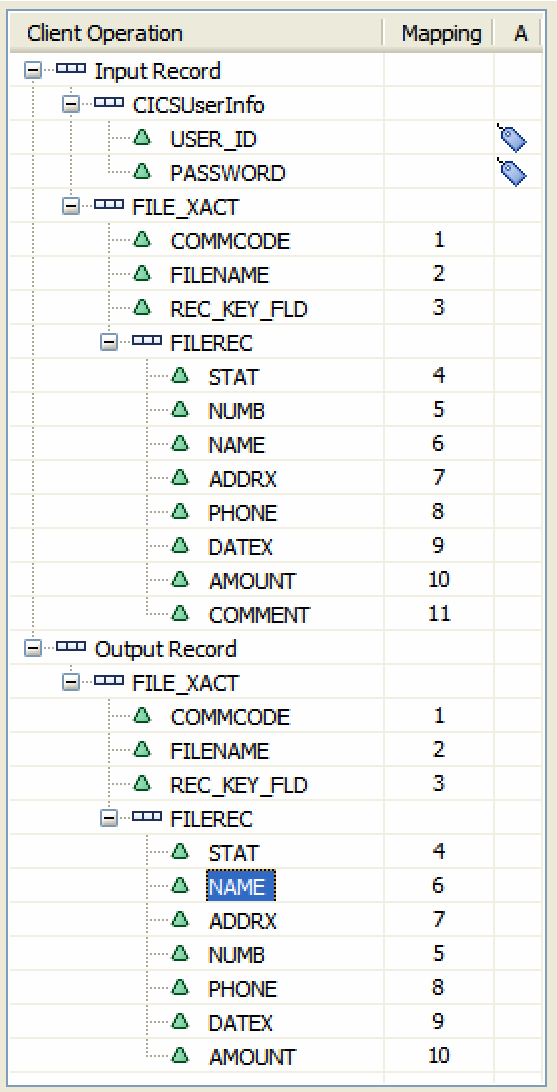
4. Let's rearrange the order of outputs under the Output Record tree and also delete an output element that won't be required by our application.

    a. In the Client Operation Editor, drag and drop the NAME (6) element under STAT (4).

    b. Drag and drop the ADDRX (7) element under NAME.

    c. Select the COMMENT (11) element, right-click and choose Delete Object.

5. Review your new Client Operation Output Record tree:

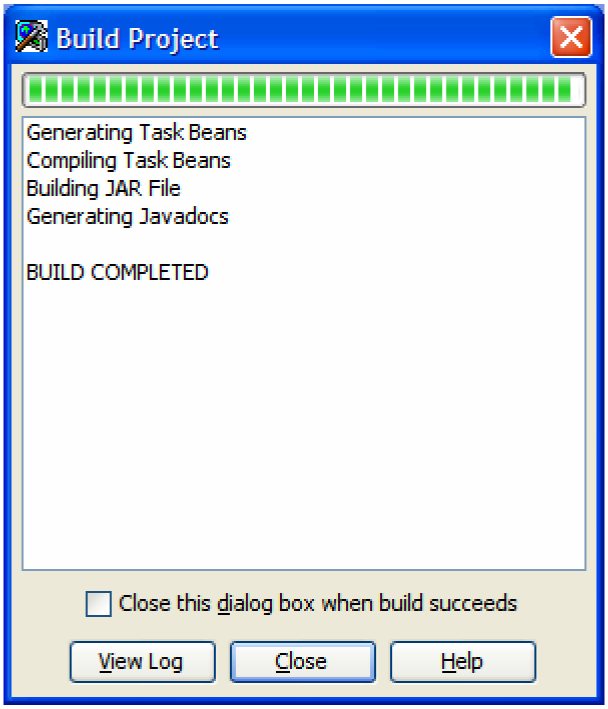


Client Operation	Mapping	A
Input Record		
CICSUserInfo		
USER_ID		🏷
PASSWORD		🏷
FILE_XACT		
COMMCODE	1	
FILENAME	2	
REC_KEY_FLD	3	
FILEREC		
STAT	4	
NUMB	5	
NAME	6	
ADDRX	7	
PHONE	8	
DATEX	9	
AMOUNT	10	
COMMENT	11	
Output Record		
FILE_XACT		
COMMCODE	1	
FILENAME	2	
REC_KEY_FLD	3	
FILEREC		
STAT	4	
NAME	6	
ADDRX	7	
NUMB	5	
PHONE	8	
DATEX	9	
AMOUNT	10	

6. Save your project.

Step Five: Create the interface object from the client operation

1. Open Web Builder by clicking the toolbar icon or selecting it from the Actions menu.
2. Open a new Web Builder project. Select Task Beans as your project type.
3. Verify that the Transaction Designer project is *sample*. The default Web Builder project name matches the designer project name.
4. Click Properties. Verify that the *sample_client_op* client operation is selected. Click Task Beans and type *sample* in the Package Name field. Accept the other default settings. Click OK.
5. Click Build to create your task bean.



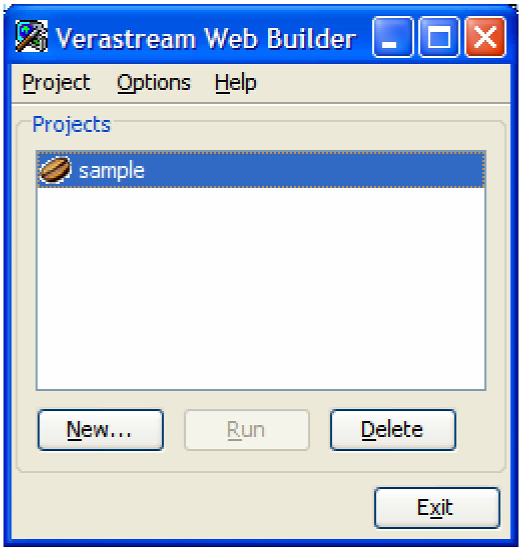
```
Generating Task Beans
Compiling Task Beans
Building JAR File
Generating Javadocs

BUILD COMPLETED
```

☐ Close this dialog box when build succeeds

Step Six: Deploy and use the task bean to develop a client application

Javadocs are available from the Designer online help, under References or under …\Attachmate\Verastream\TransactionIntegrator\Documentation\javadoc

1. Select *sample* from the Web Builder project list.
2. Right-click on the project name. Choose Explore to navigate to the jar file.



3. Copy …\Attachmate\Verastream\TransactionIntegrator\Designer\ ↪WebBuilder Projects\sample\taskbeans\sample.jar to the machine where you will compile your applications and from where they will be served. Sample.jar contains the Java bean and the transaction runtime module classes necessary to run the application.
4. If you are using a development environment (such as Eclipse or NetBeans) to write your Java application, then add the generated jar file and the required client jars to your project. See the Configure Client Access to Runtime Servers section of this document.

   If you are writing your Java application manually, add the generated jar file and required client jars to your classpath.

5. To write your Java application, import the generated Java bean into your application and code calls to set inputs, execute the transaction, and get outputs.

   For an example, see …\CICSClient\samples\AMNU\ITASK\AMNUTask.java.

6. Compile and run your application.