TIBCO®
The Power of Now®

BusinessConnect™ Insight Installation and Configuration

May 2011

TIBCO Software Inc., Foresight Group
655 Metro Place South
Suite 900
Dublin OH 43017

Phone: (614) 791-1600
Fax: (614) 791-1609

Web: http://foresight.TIBCO.com
E-mail: FSSupport@TIBCO.com

Contents

1 Introduction

Intended Audience

This document is intended for the technical team that is installing and configuring BusinessConnect™ Insight (BCI).

To use this guide, you should have:

- Basic knowledge of EDI
- Basic knowledge of Windows system administration
- Basic knowledge of UNIX administration if you are installing parts on UNIX
- Basic knowledge of databases or access to a database administrator

System Requirements

Please see **System_Requirements.pdf** in the TransactionInsight Doc directory.

BCI, TI and Transaction Insight

The BCI product is installed into a directory called **\Foresight\TransactionInsight**. Throughout this document you will see references to TI and Transaction Insight in pathnames (e.g., C:\Foresight\TransactionInsight\4.1.0\ Environments*envname*\java) and program names (e.g., TIAutomator). In all cases, the terms TI and Transaction Insight are referring to the BCI product and the files and components within it.

Validation Requirements

EDI imported into BCI needs certain information in the validation detail file. The following settings will create a detail file that is suitable for importing.

This is the set of APF records that must be set to 1:

STRT	END	ENDS	DTL	EMSG
EDAT	ESEG	VER	SVRTY	ETYPE
SVRTS	ETYPS	GEN	ZREC	STRUS
STRUE	SVALU	EVALU	UID	

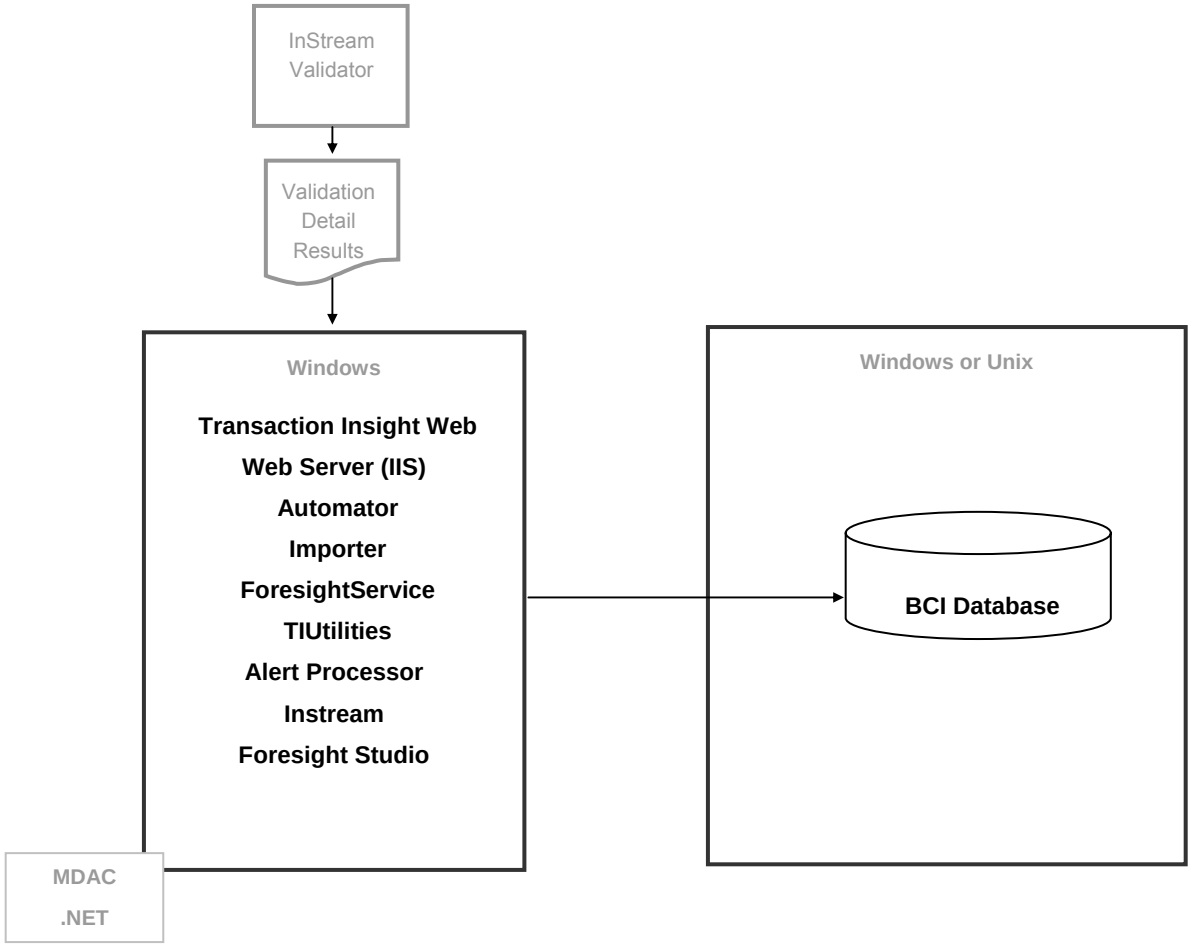
	Guideline	$Dir.ini/fsdir.ini	APF
BusinessConnect Insight Imports X12 210, 810, 850, 856, 856, 860			APF records listed above must be set to 1. If using TIMatcher, Custom User ID, or ScenarioDetector: IDENT=1 Use EDISIM Standards Editor to add DSR marks: ISA1, GSGS, and STST.

Sample Implementations

Since applications are highly configurable, your implementation can be different from these.

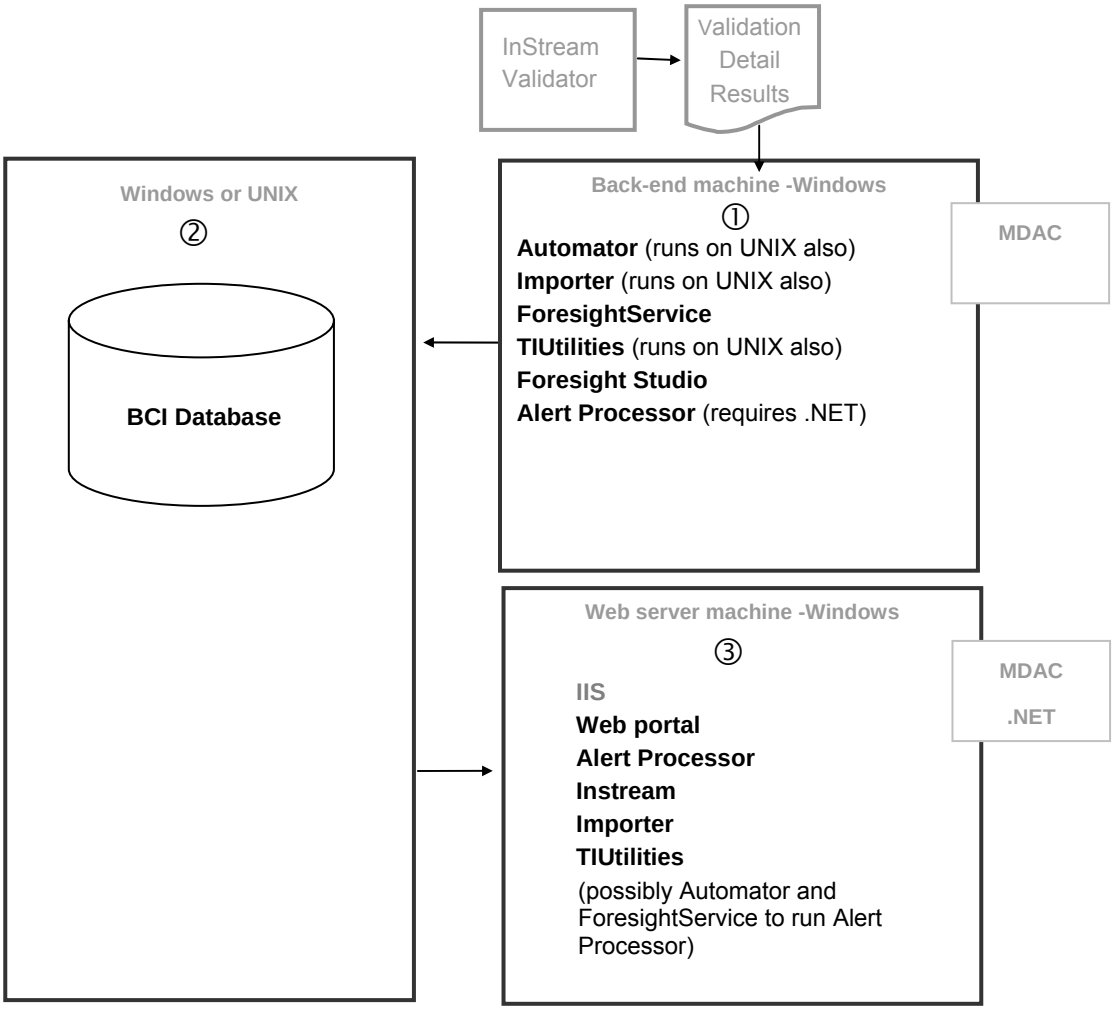
Example Two-Machine Implementation

For development or QA, you might set up a two-machine system like the one below. One machine is a database server and the other contains all portal components.



Example Three-Machine Implementation

You should always set up the databases on their own machine. In this example there are three machines: a database server, a web server and a "back-end" machine.



The gray areas above are necessary but are not part of the installation program.

① ForesightService starts Automator when the computer is turned on. Automator lets you run workflows containing programs such as Importer and TIUtilities.

② Databases are on your SQL Server or Oracle database machine.

③ The portal is a website that shows reports about what is in the databases.

You can run Alert Processor on this machine also, since it requires .NET. As an alternative, you can run Alert Processor on machine ① if you install .NET on it. Please see page 28 for more setup information.

Overview of BCI-Related Programs

Application Name/ Executable file name/ Configuration file	**Default Location**	**Purpose**
AlertProcessor AlertProcessor.exe AlertProcessor.exe.config	C:\Foresight\TransactionInsight\ *version* \Environments*envname*\bin	Proactively monitors the system for events and sends e-mail when an event occurs. See page 28.
Automator ForesightAutomator.exe	C:\Foresight\ForesightAutomator\ *version*\ bin	Runs a workflow. Example: moves EDI files to an input directory and then runs validation.
ForesightService ForesightService.exe FSService.ini	C:\Foresight\ForesightService\ *version*\bin	Runs Automator as a service so that it can automatically start on system startup. See page 31.
Importer Importer.exe Importer.ini	C:\Foresight\TransactionInsight\ *version*\ Environments*envname*\bin	Parses a validation detail results file and populates the database with its values, errors, and statistics. See page 32.
TIMatcher TIMatcher.jar	C:\Foresight\TransactionInsight\4.1.0\ Environments*envname*\java	Lists information about related documents at the bottom of the Document Summary page, including a link to each document.
TIWeb No executable filename Web.config	C:\Foresight\TransactionInsight\ *version*\ Environments*envname*\ TIWeb	Portal user interface. It displays web pages, queries and updates the portal databases based on user action. See page 33.

Application Name/ Executable file name/ Configuration file	Default Location	Purpose
TIUtilities TIUtilities.exe TIUtilities.ini	C:\Foresight\TransactionInsight\ *version*\ Environments*envname*\ bin	Processes each transmission after it is populated into the database. The processes it can carry out are: Populating summarized statistical tables. Deleting expired data. Generating filter values and automatically assigning them to partners. See page 35.
Instream HVInStream.exe DocSplitter.exe (and other Instream programs) $dir.ini or fsdir.ini	C:\Foresight\InStream\Bin	HVInStream.exe: Validates data, creates detail results file for importing into TI. DocSplitter.exe: Splits an application document from a file
Foresight Studio Foresight.Studio.IDE.exe Foresight.Studio.IDE.exe.config	C:\Foresight\ForesightStudio\ *version*\	Creates Automator workflows.
ScenarioDetector ScenarioDetector.jar	C:\Foresight\TransactionInsight\4.1.0\ Environments*envname*\java	Used for Scenario Detection.

2 Preparation before Installing

Staff Skills Needed

Before and during installation, you will need access to:

- EDI business analyst
- Experienced database administrator
- Network administrator with administrative privileges on all machines

Planning Database Creation

Before installation day, the database administrator should communicate with the person who will run the installation program about who will create the databases.

Options include:

- Before the installation program is run, the database administrator can create an empty database for BCI. This can be done after the installation program too.
- The person installing the portal applications chooses "Use DB" and points to the database.
- When finished, the database administrator runs the scripts created by the installation program.
- If the database is SQL Server, the person running the installation program can actually create and configure the database when running the front-end installation.

The database administrator should choose an option above and instruct the person who will run the installation program:

- What database name(s) to use and on what server(s).
- Whether to "Use DB" or "Create" databases if given a choice during installation:

Please see Appendix H: Database Scripts on page 69 for details on running scripts created by the installation program. The Scripts directories have ReadMe.txt files with instructions.

Setting up tnsnames.ora

(Oracle only, Windows and UNIX)

The actual connection is still done using Instant Client directly.

The tnsnames.ora file is a way of specifying a shortcut for an Oracle connection. Instead of using one of the connection string formats (such as the EZ-connect syntax) directly in your configuration file, you can put all that information in your tnsnames.ora file and give it a simple shortcut name like **mydb**.

If you are using a tnsnames.ora file to connect to your database, follow these steps on all portal product machines:

1. Set up a tnsnames.ora file containing information needed to connect to your BCI database. Your Oracle database manager can supply this, or you can copy it from another machine that connects to the same databases. See the example below.
2. Create a new environment variable to tell Oracle where to find tnsnames.ora:
   a. Click New under the Control Panel | System | Advanced | Environment Variables | System variables pane.
   b. For Variable Name, type TNS_ADMIN.
   c. For Variable Value, type the directory path to tnsnames.ora.

      This is the directory path only. Do not include the filename.
   d. Click OK until you return to your desktop.
   e. Restart your machine

Example simple tnsnames.ora file

This example shows a single Net Service Name (ora901_testbox) which points to an Oracle server on machine at IP address **192.168.1.40** with a SID of **ora10**. You should be able to connect to the machine of your choice by changing the HOST= and SID= entries to suite your local environment.

```
# tnsnames.ora Network Configuration File:
C:\oracleinstantclient\tnsnames.ora

ora901_testbox =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.1.40)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SID = ora10)
      (SERVER = DEDICATED)
    )
  )
```

Database Sizes

These are general guidelines and your size requirements will vary.

BCI Database

Plan on more space if you have:

- Large documents
- Transaction filters, especially ones that are not "not search only."
- Number of document fields chosen under **Settings | Document Fields**.

As a rough estimate:

- BCI averages about 2.5 KB of disk space per application document. This is about 2.5.GB of disk space per million documents.
- Saving good data quadruples the database size. When saving both good and bad data, the average size would be about 10 GB/million documents.

Examples

Average disk space needed for one year at 50K transactions per day.

Importing only bad data	*DocsPerDay * Days * 2.5GB = disk space in GB* (50k docs / day) * (365 days) * (2.5 GB / 1,000,000 docs) = about 46 GB
Importing good and bad data	(50k docs / day) * (365 days) * ((2.5*4) GB / 1,000,000 docs) = about 182 GB

Information to Collect

Before running the installation program, collect this information. Keep it with your important portal notes for reference during installation and later.

Look through the prompts shown in Installation Day on Windows on page 13 and be sure you have all the information that you will need.

Item	Get this information from …
Network name of each machine involved, and whether are 32-bit or 64-bit	Network administrator
Environment name for portal (goes in web URL for portal)	Discuss with web server administrator
The "from" e-mail address for portal alerts	Discuss with network administrator
SMTP server name	Network administrator

If using Oracle databases	
User ID and Password for Oracle access or Windows/UNIX login	Oracle database manager
System identifier (SID)	Oracle database manager
One of these: - Name of database machine, port, and Oracle instances for TI - or TNS_ADMIN environment variable and corresponding TNSnames.ora file that contains connection information for the databases	Oracle database manager
If using SQL Server databases	
Network name of the SQL Server machine	SQL Server database manager
Name of existing or new databases for TI	SQL Server database manager
User ID and Password for SQL Server or Windows login username and password	SQL Server database manager
Whether to “create” a new database, or “use” one already created by the database manager	SQL Server database manager

License Files

Get these from your TIBCO Foresight account executive:

- **TILicense.ini** for TI’s bin directory on the server where TIUtilities and Importer will run.
- **license.ini** or **ISLicense.ini** for Instream’s Bin directory

Create a Configuration Plan

It will be helpful if you prepare a diagram of your intended configuration, similar to the sample implementation diagrams on pages 4 and 3. This will help you when running the installation program.

Then look at the components choices you will have during installation and determine which combinations you will be installing on each server. Please see Appendix A: Components Choices during Installation on page 39.

Review **System_Requirements.pdf** in the TransactionInsight Doc directory and be sure that they are all met on the servers that you intend to use.

Please call TIBCO Foresight Support and discuss your intended configuration before installing.

Basic Permissions

Your permission needs will vary, depending on your specific implementation.

- The user running a workflow needs permissions to all folders that the workflow accesses.
- Recommendation: Keep it local for a workflow. For security, simplicity, and speed, keep all folders and applications on the server where the workflow is running.
- The Oracle or SQL Server user in the connection string (sa, for instance) needs permission to create schema/database, create tables, views, stored procedures, read, update.
- ASPNET worker process needs read/write access to folders created for BCI partner groups.

Instream Considerations

Some BCI applications need to have Instream configured to insert FSUIDs (Foresight Unique IDs) in one or both of these locations in the validation detail (DTL) file:

- At the top of the file in a GEN record with error number 11005.
- At each application document in an IDENT record.

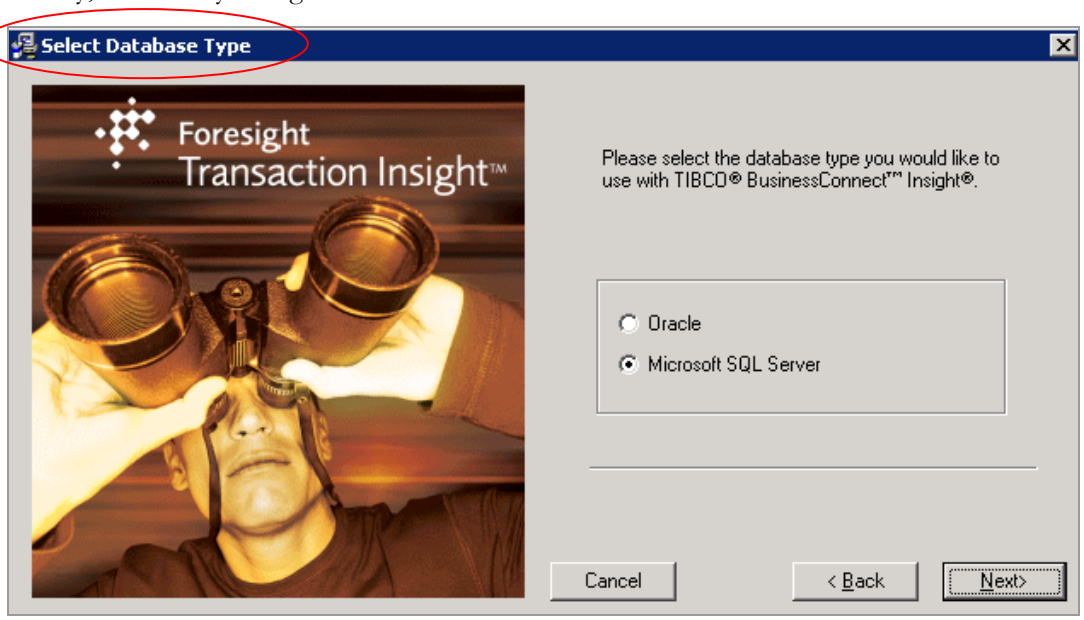
The default installation on Windows has trading partner automation set up to cause Instream to validate with guidelines that put both of these in the detail results file. It copies the files TI_Demo.apf and TI_Demo.csv to Instream's **Bin** folder and the $Dir.ini file is set up to use them.

Please refer to **FSUID_and_AppDocs.pdf** in Instream's Doc directory for further information on which guidelines to use with each BCI application so that these records are inserted into the detail file when they are needed. Also see Appendix I: Instream and TI on page 71.

3 Installation Day on Windows

Before Starting the Installation Program

When installing BCI applications, always look here to verify the specific application and activity, since many dialog boxes look similar:



Running the Windows Installation Program

You will need to run the installer on all Windows machines that are to contain BCI application components. The only exception is the database machine.

1. Have your necessary people (see page 7), list of necessary information (see page 10), and configuration plan (see page 11) handy.
2. Turn off IIS on the web server (iisreset /stop at the command prompt).
3. Turn off ForesightService (under Control Panel's Services).
4. Execute one of these BCI application installation programs:

   TISetUp32.exe on 32-bit systems

   TISetUp64.exe on 64-bit systems
5. You will be asked to make many of the same types of decisions that installation programs typically require.

Choose Components

When you get to the panel where you select components, choose the components that go on this machine. Back-end components (Importer and TIUtilities) always install.

Please see Validation Requirements

EDI imported into BCI needs certain information in the validation detail file. The following settings will create a detail file that is suitable for importing.

This is the set of APF records that must be set to 1:

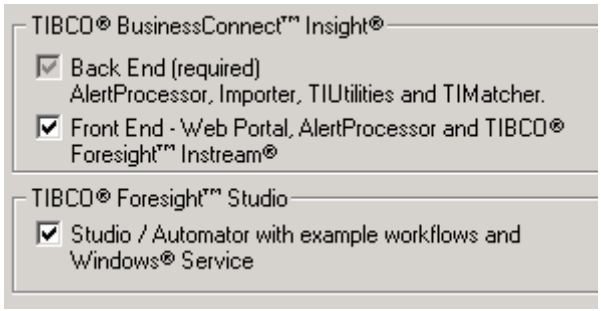
STRT	END	ENDS	DTL	EMSG
EDAT	ESEG	VER	SVRTY	ETYPE
SVRTS	ETYPS	GEN	ZREC	STRUS
STRUE	SVALU	EVALU	UID	

	Guideline	$Dir.ini/fsdir.ini	APF
BusinessConnect Insight Imports X12 210, 810, 850, 856,			APF records listed above must be set to 1. If using TIMatcher, Custom User ID, or ScenarioDetector:

856, 860			IDENT=1 Use EDISIM Standards Editor to add DSR marks: ISA1, GSGS, and STST.

Sample Implementations on page 2 and Appendix A: Components Choices during Installation on page 39.

Choose TI components

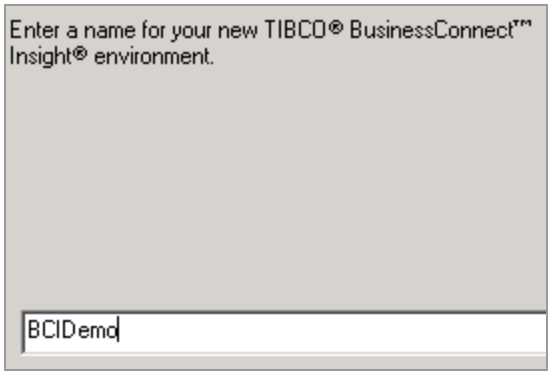


Back End
AlertProcessor (see page 28)
Importer (see page 32)
TIUtilities (see page 35)
TIMatcher (see page 34)

Front End
Web Portal (see page 33)
AlertProcessor (see page 28)
Instream (separate product)

Foresight Studio …
Automator (see page 30)
Foresight Studio (see page 32)
Workflows
Foresight Service (see page 31)

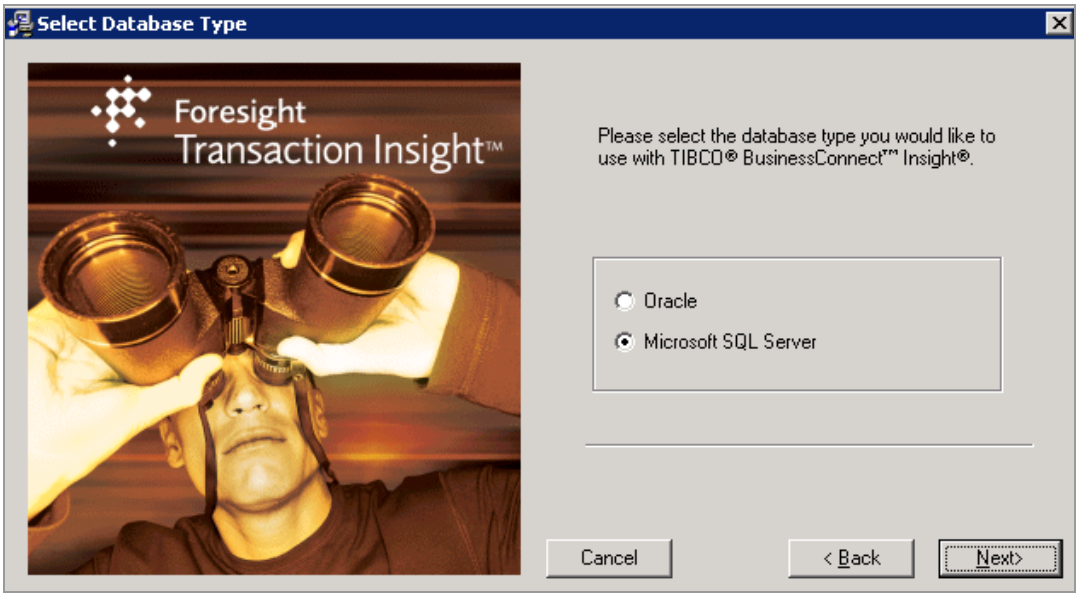
Choose a Name for this Environment



- The name will be part of the directory structure and the names of some of the batch files that start Automator.
- Avoid version numbers in the name since this will continue to be the name through future upgrades.
- Many Automator batch files will start with the environment name you are specifying. For example, if you called your environment BCIDemo, you will see files named BCIDemoOutbound.cmd, BCIDemoMatcherDemo.cmd and so on.
- Directory paths are affected by the environment name as well. Each environment has its own web portal, set of directories, and configuration.

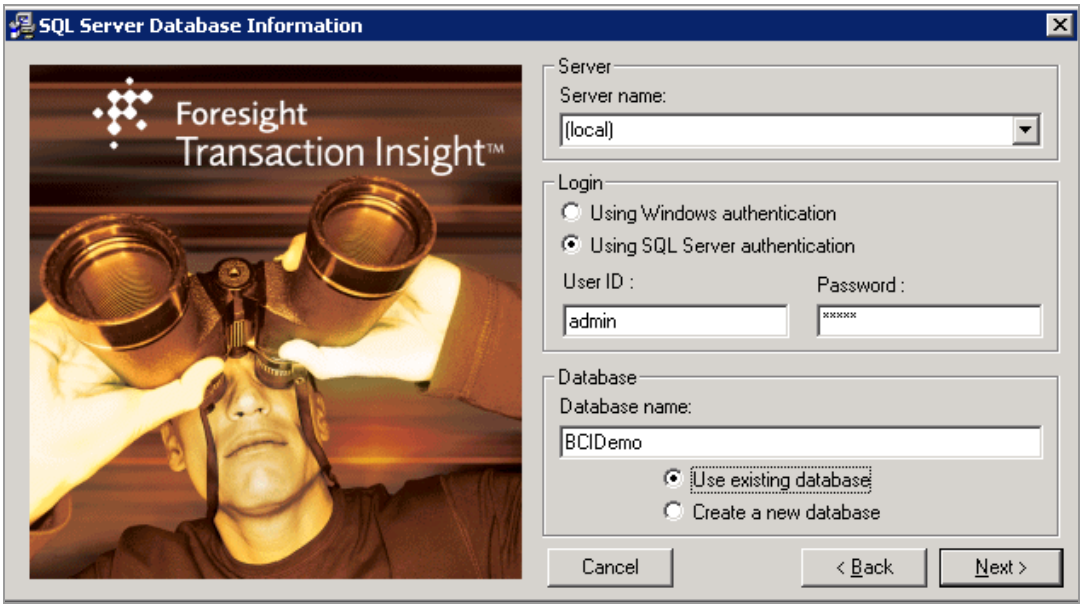
Specify a BCI Database

When you get to the Select Database Type dialog box, choose the type of database that you intend to use:



If you chose Sql Server

When you get to the SQL Server Database Information screen:



Server Name Server (machine) where database is located. `(local)` means the current machine.

Login Select the method of accessing the database.

Using Windows authentication

Choose this if you want to connect to the database using a Windows user ID and password.

Using SQL Server authentication

Choose this if you want to connect to the database using a SQL Server user ID and password.

User ID If you selected SQL Server authentication, type the login ID (`sa`, for example) for SQL Server on the machine where your database will reside. This is the ID that will be in effect at runtime.

Password Type the corresponding SQL Server password. This is the password that will be in effect at runtime.

Database Name Remember that this will be a permanent database, even after you upgrade to new versions of BCI. Therefore, don't include the version as part of the name. Example: `BCIdemo` is better than `BCI10demo`. Do not use spaces or special characters.

Use existing database This is the choice if you already have a BCI database or if you are going to set up the database later by running the scripts that come with the installation program.

Create a new database If you want the installation program to create the database, select this option. After finishing the installation, give your database administrator the scripts described in Appendix H: Database Scripts on page 69.

If you chose Oracle

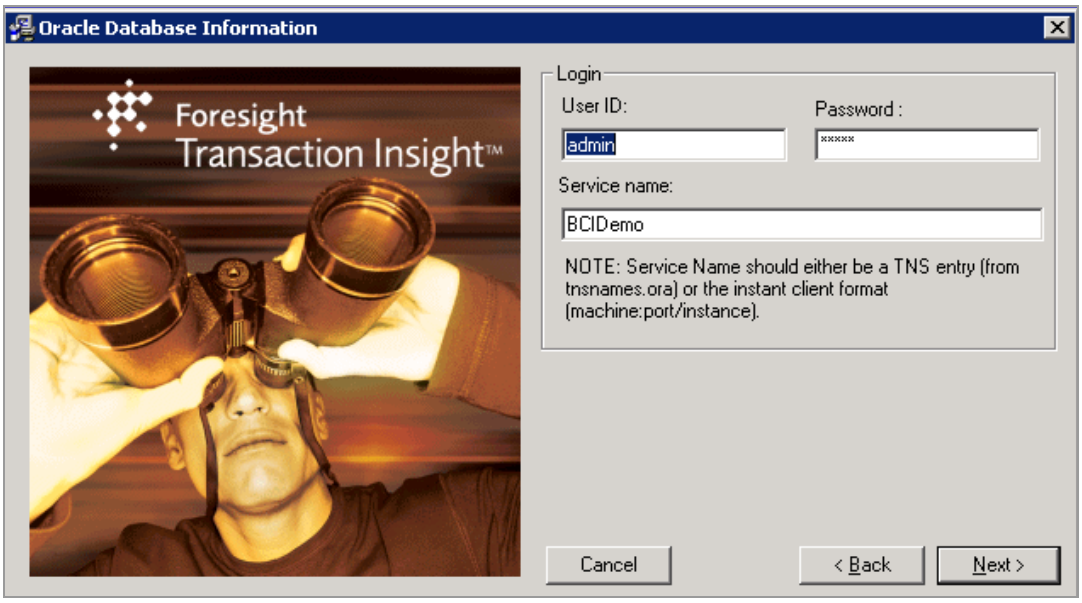
Fill in the dialog box about the Oracle **BCI** database:

User ID User ID for the database (from DBA).

Password Password for the database (from DBA).

Service Name This is the Instant Client format or a TNS Name from the tnsnames.ora file that is to be used when BCI connects to the database.

Example using TNS name for Service Name



After finishing the installation program, you will have database scripts to run to complete your Oracle BCI database. Please see Appendix H: Database Scripts on page 69.

Set up BCI Web Component

The Web Configure dialog prompts for:

Virtual Directory Name — The logical name for the directory containing the web component for the environment you are installing. Avoid using the web component version as part of the name, since this will probably be used in future upgrades.

Smtp Server — Name of the server that is to route and send outgoing e-mail containing BCI web component alerts. Be sure that this server is configured to permit external emailing.

E-mail — The "from" e-mail address from which portal alerts are sent.

Web URL — The URL for the BCI web component. Write down this address. You will need it later to access the web component's login screen, and you will need to send it to BCI users. Depending on how your web server is configured, your users may have to add Default.asp**x** to the end of the URL. In this example, users would log on at:

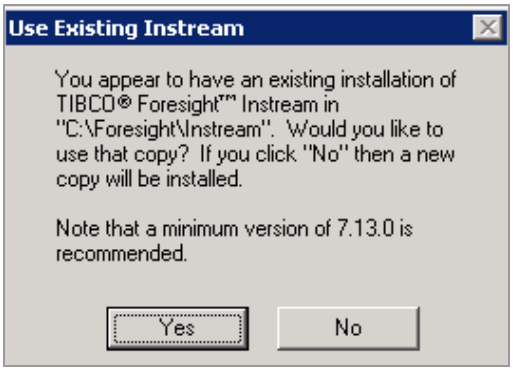
```
http://NA-DUB-DOC/BCIDemo/Default.aspx
```

Specify Instream

Instream will install next.

If you already have Instream installed on this machine, you will be asked if you want to use it with BCI:

- Note the Instream directory in the prompt
- Check Instream to see if it is the version recommended in the prompt:

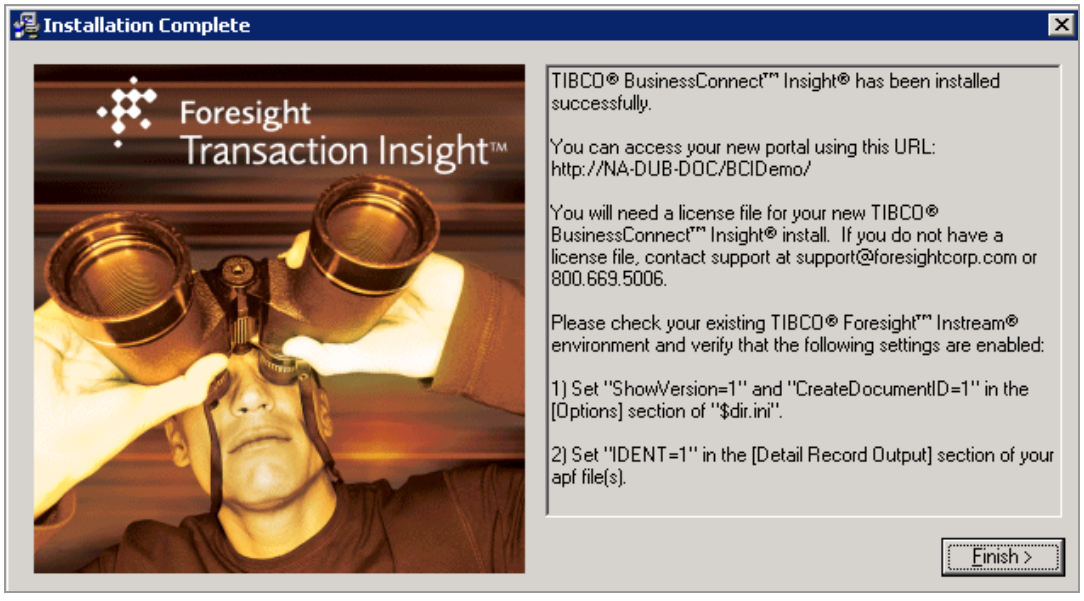


(To check Instream's version, execute the Version script in Instream's Scripts directory)

- Select **Yes** to have BCI use this Instream for validation
- Select **No** to install Instream

Installation is successful

The Installation Complete dialog contains important information:



- The URL to access your BCI Web component.
- A reminder that you must have a license file to use BCI (see License Files on page 11).
- Additional steps to follow to ensure certain Instream settings are enabled.

Click **Finish** when you are done with this dialog.

4 Installation Day on UNIX

Parts that can be Installed on UNIX

- BCI database (Oracle only)
- Importer
- TIUtiliities
- TIMatcher
- ScenarioDetector
- Automator
- Instream
- Workflows

Installing Instream on UNIX

Follow the directions in **InstallNotes.txt** for Instream. Please see Instream Considerations on page 12 for details.

Installing Automator on UNIX

Follow the directions in InstallNotes.txt.

Installing BCI on UNIX

Follow the directions in the TI section of AIX_ReadMe.txt that is distributed with the BCI installation kit.

Steps

1. Run the BCI installation program on Windows and install all parts.
2. Read the BCI readme that came with your operating system (AIX_ReadMe.txt, etc.)
3. Create a directory to contain the TransactionInsight directory on the UNIX machine:

   ```
   mkdir Foresight
   cd Foresight
   ```

4. Copy the UNIX TI archive to the directory you just created:

   ```
   ~/Foresight ls
   -rw-r--r--    ...     <archivename>.tar.gz
   ```

5. Extract the archive. This example will create a `TransactionInsight/`*`version`*`/bin` directory and load files into it:

   ```
   gunzip <archive name>.tar.gz
   tar -xvf <archive name>.tar
   ls -R
     -rw-r--r--       ...             <archive name>.tar
     drwxr-xr-x       ...             TransactionInsight
     ./TransactionInsight:
     drwxr-xr-x       ...             <version>
     ./TransactionInsight/<version>:
     drwxr-xr-x       ...             bin
     ./TransactionInsight/<version>/bin:
     -rw-r--r--       ...             ErrMsgTrans<Instream version>.txt
     -rwxr-xr-x       ...             FSSendMail.sh
     -rwxr-xr-x       ...             Importer
     -rw-r-----       ...             Importer.ini.example
     -rwxr-xr-x       ...             Java
     -rwxr-xr-x       ...             TIUtilities
   ```

6. Copy your new **TILicense.ini** to BCI's bin directory.
7. Go to the **bin** directory and customize Importer.ini as detailed on page 32:

   Check all settings. Update the connectstring line in the [Database] section to point to your BCI database. Example:

   ```
   connectstring=UID=myuid;PWD=mypwd;DATABASE=BIGUNIX1:1521/or10
   ```

8. FSSendMail.sh is used by TIUtilities to send e-mail. You may need to modify this script if:
   - Your sh interpreter is not in /usr/bin directory

     If this is the case, change the top line of this script to point to the location of sh on your system.
   - You prefer to use a different mail program than mailx.

     The arguments supplied to mailx are documented in the script.

9. To run Importer on UNIX, set up `LIBPATH` and `FSIMPORTERINI` environment variables to point to TI's **bin** directory.

   This example Bourne Shell script shows how to set up the environment and run Importer. In this example, a user named **ti** has extracted the archive into their home directory.

```
#!/usr/bin/sh
export LIBPATH=/home/ti/TransactionInsight/<version>/bin
export FSIMPORTERINI=/home/ti/TransactionInsight/<version>/bin
/home/ti/TransactionInsight/<version>/bin/Importer \
-i \
-e /home/ti/TransactionInsight/<version>/DemoData/Tutorial-A.edi \
-r /home/ti/TransactionInsight/<version>/DemoData/Tutorial-A.dtl \
-l /home/ti/TransactionInsight/<version>/DemoData/Tutorial-A.log
```

10. To run TIUtilities on UNIX, set up the `LIBPATH` and `FSTIUTILITIESINI` environment variables to point to TI's **bin** directory..

    This example Bourne Shell script shows how to set up the environment and run TIUtilities. In this example, a user named **ti** has extracted the archive into their home directory.

```
#!/usr/bin/sh
export LIBPATH=/home/ti/TransactionInsight/<version>/bin
export FSTIUTILITIESINI=/home/ti/TransactionInsight/<version>/bin
/home/ti/TransactionInsight/<version>/bin/TIUtilities \
-dbtype "oracle" \
-db "DATBASE=db-server:1521/instance-name;UID=TIDEV;PWD=TIDEV" \
-t \
-url "http://webserver/TIDEV/" \
-smtp "smtp-server.mydomain.com" \
-from ti@mydomain.com
```

11. Run the BCI installation program on a Windows machine to set up the Web component.

12 To set up or update a BCI database on UNIX, customize the UNIX scripts created by the Windows installation program. See Appendix H: Database Scripts on page 69.

13. Install Instream, if you are running it on UNIX and if it has not been installed already. See InstallNotes.txt that comes with UNIX versions of Instream.

14. Install Automator on UNIX, if you are going to run an Automator workflow on UNIX and it has not been installed already. See **Automator.pdf** in the TransactionInsight Doc directory.

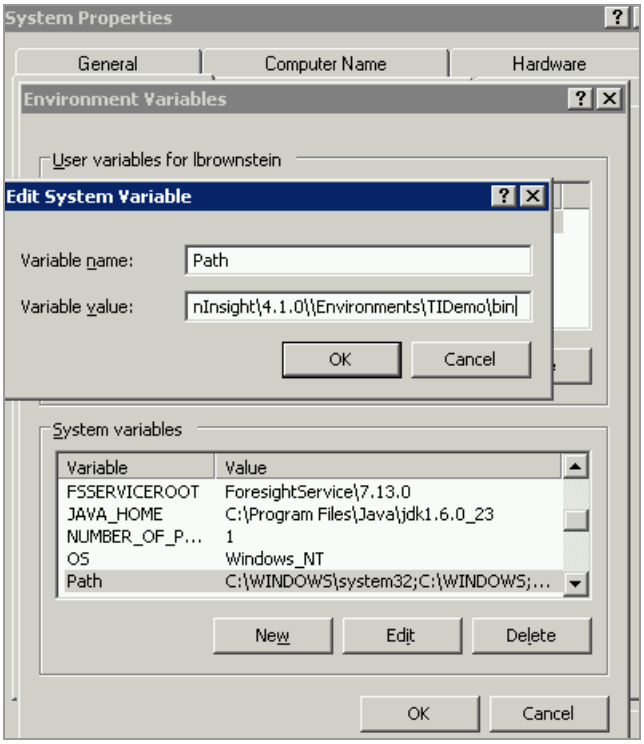
5 General Configuration after Installation

Running Database Scripts

Your database administrator will need to run scripts before you can use your BCI applications. Please see Appendix H: Database Scripts on page 69.

Updating your System Path

Go to **Control Panel | System | Advanced System Settings | Environment Variables** and put your BCI environment's Bin directory in the System path:



Copy and Test License Files

Install License Files

Copy these license files:

Instream:	Copy license.ini or ISLicense.ini to Instream's Bin directory
BCI (TI)	Copy the TILicense.ini to Foresight's TransactionInsight\<*version*>\Environments\<*environment*>\bin

Test License Files

Validate the BCI license with **Start | All Programs | Foresight | Transaction Insight |** <*version*> **| Validate License**.

Test the Instream license by running a simple script like V_810_5040_X_1, which you can find in Instream's Scripts directory. You will get a message about the license failing if it does not work.

6 Configuring other Foresight Programs

AlertProcessor

Executable AlertProcessor.exe

Purpose Proactively monitors the system for events and sends e-mail when an event occurs.

Configuration file AlertProcessor.exe.config

This is a standard .NET XML-based application configuration file.

Keep AlertProcessor.exe and AlertProcessor.exe.config in the same directory.

Default location Front-end or back-end machine. It is an ASP.Net application which will only run on Windows:

C:\Foresight\TransactionInsight*version*\Environments*envname*\bin

Where:

version	Version number.
envname	User-supplied name of the environment set up during installation.

Configuring AlertProcessor.exe.config	
Setting	**Explanation**
DBType:TransactionInsight	**SqlServer** or **Oracle**
DB:TransactionInsight	Application Name TIWebGUI server **(local)** or name of server where DB is located. Database or initial catalog Name of database from installation. UID SQL or Oracle user ID. PWD Corresponding password. For details about using Windows Integrated Security instead of a UID and PWD here, see page 44.
SmtpServer	The SMTP server the application uses to send e-mail notifications.
FromAddress	The “From:” address in e-mails sent by BCI.
WebApplicationUrl	The URL of the web application root. This is used to include URLs (links) in e-mails sent by the application.
smtp deliveryMethod	Uncomment this section if you want to have AlertProcesor send email alert notifications using the IIS SMTP component. With this method, email alerts are queued locally, preventing the loss of information if the target server is unavailable. Consult your web server administrator before using this method, since you need a local IIS server with the SMTP option. To uncomment, remove `<!--` and `-->` Default behavior is to have AlertProcessor send the alert notification emails directly using SMTP.

Example appSettings entries

```
<add key="DBType:TransactionInsight" value="SqlServer" />
<add key="DB:TransactionInsight" value="Application
Name=TIWebGUI;server=FCDOC01;Database=Demo;UID=sa;PWD=a1s2d3f4;"></add>
<add key="SmtpServer" value="relay.foresightcorp.com"></add>
<add key="FromAddress" value="TImanager@foresightcorp.com"></add>
<add key="WebApplicationUrl" value="http://FCDOC01/Demo//" />
```

Automator

Executable ForesightAutomator.exe

Purpose Runs a workflow.

Configuration file XML files in workflows under Foresight\Systems\ *workflow*\Workflows*version*\Automator.

Default location Automator/Studio/Windows Service (ForesightService) machine:

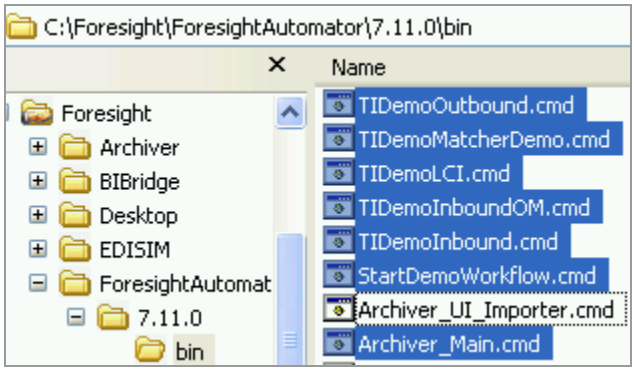
C:\Foresight\ForesightAutomator*version*\bin

Where:

version Automator version.

Documentation **Automator.pdf** in the TransactionInsight Doc directory.

Automator's bin directory contains files to start the workflows that are automatically installed. The TI workflow filenames start with the environment name. In this example, the environment is **TIDemo**:



ForesightService

Executable	ForesightService.exe
Purpose	Lets you run Automator when you are logged out. Look for ForesightService under **Control Panel | Administrative Tools | Services**.
Configuration file	FSService.ini. Keep this in the same directory as ForesightService.exe.
Default location	C:\Foresight\ForesightService*version*\bin on machines with Automator. Where: *version* Foresight Service version.
Documentation	**ForesightService.pdf** in the TransactionInsight Doc directory.

Installs with the Automator / Studio and Windows Service checkbox.

Foresight Studio

Executable Foresight.Studio.IDE.exe

Purpose Creates, modifies, and runs Automator workflows.

Configuration file None

Default location C:\Foresight\ForesightStudio*verson*\bin

Where:

version Studio/Instream version such as 7.11.0.

Documentation **ForesightStudio.pdf** in Foresight Studio's Doc directory.

Importer

This installs with BCI Web (TIWeb).

Executable Importer.exe

Purpose Import EDI data and statistics into the BCI database. This Windows executable uses information from the Instream validation results file and (optionally) from an EDI file.

Configuration file Importer.ini

Determines how Importer connects to the BCI database.

Keep Importer.ini and Importer.exe in the same directory.

Default location C:\Foresight\TransactionInsight*version* \Environments*envname*\bin

Where:

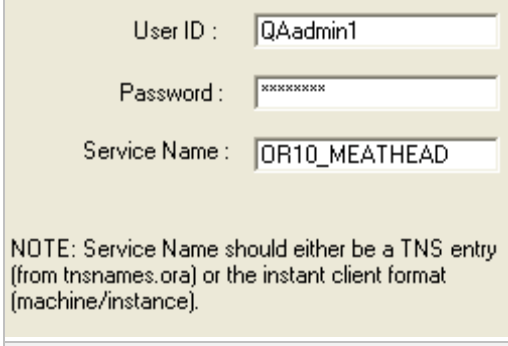
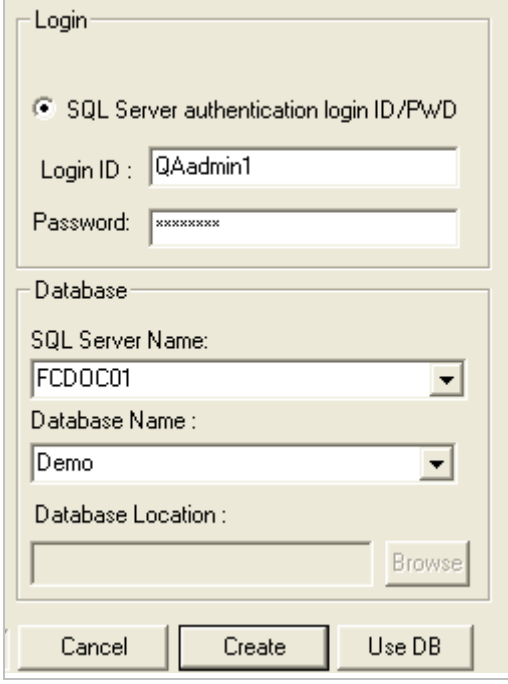
version Version number.

envname User-supplied name of the environment set up during installation.

Documentation **Importer.pdf** in the TransactionInsight Doc directory.

Settings in Importer.ini

Inside Importer.ini, customize **connectstring** under **[Database]** as explained in the file. This will be preconfigured from the information you entered during installation. For details, see **Importer.pdf** in the TransactionInsight Doc directory.

<table>
<tr><th colspan="2">Oracle Example</th></tr>
<tr><th>Example Installation Information</th><th>Resulting Connectstring</th></tr>
<tr><td> 
</td><td>connectstring=DATABASE=OR10_MEATHEAD; UID= QAadmin1;PWD= w2e3r4t5</td></tr>
<tr><th colspan="2">SQL Server Example</th></tr>
<tr><th>Example Installation Information</th><th>Resulting Connectstring</th></tr>
<tr><td> 
</td><td>connectstring=DRIVER={SQL Server};SERVER=FCDOC01;DATABASE=Demo; UID=QAadmin1;PWD= w2e3r4t5

or

connectstring=DRIVER={SQL Server}; SERVER=FCDOC01;DATABASE=Demo; Trusted_Connection= yes</td></tr>
</table>

Instream

Please see **Appendix I: Instream and TI** on page 71 for details.

TIMatcher

Executable TIMatcher.jar (run by TIMatcher.bat)

Purpose Matches related documents in the BCI database. When the BCI user displays a Document Summary page, a list of matching documents appears at the bottom with links.

Configuration file TIMatcher.config.

Default location The BCI environment's bin directory.

Documentation **TIMatcher.pdf** in the TransactionInsight Doc directory.

Web.config settings

Please see Appendix L: Web.config Settings on page 78.

TIWeb

Executable none

Purpose Web application that lets users view statistics and administer application level security. It uses information in the BCI database.

Configuration file Web.config. This file may contain a username and password to the database and must be protected from access via the web.

Other configuration ASPNet User must have full read/write access to all folders in the TIWeb directory. This includes any folders created during BCI execution (for example, TIWeb's Temp directory).

Default location Web.config is found in the virtual directory of the web application. By default, this is:

C:\Foresight\TransactionInsight*version*\Environments*envname*\TIWeb

Where:

version	Version number.
envname	User-supplied name of the BCI environment specified during installation

Documentation **BCI_Admin***nnn***.pdf** and **BCI_User***nnn***.pdf** in the TransactionInsight Doc directory.

Please see Appendix L: Web.config Settings on page 78.

TIUtilities

Installs with BCI Web. There is no separate TIUtilities installation program.

Executable TIUtilities.exe

Purpose After Importer.exe imports a transmission into the BCI database, TIUtilities.exe processes it by:

- Populating summarized statistical database tables.
- Deleting data from the BCI database after certain administrator-specified conditions are met (see Expiring data below).
- Generating filter values and automatically assigning them to partners using the information given on the Filters page. See **BCI_Admin*nnn*.pdf** in the TransactionInsight Doc directory for details.

Configuration file Any XML file that runs TIUtilities in Systems*systemname*\Workflows\1.0*workflow*

Default location C:\Foresight\TransactionInsight*version* \Environments*envname*\bin

Where:

version	Version number.
envname	User-supplied name of the environment (or site) where BCI was installed

Documentation **TIUtilities.pdf** in Transaction Insight's Documentation directory.

Settings in TIUtilities XML files	
Setting	**Explanation**
See: **Overview of Component XML Files** on page 55 **TIUtilities_5MinUpdate.xml** on page 65 **TIUtilities_6HrUpdate.xml** on page 67 **TIUtilities_12HrUpdate.xml** on page 68 For details, see **TIUtilities.pdf** in Transaction Insight's Documentation directory.	

Expiring data

The TIUtilities -e command line parameter expires (deletes) data from your database according to your selections on the BCI's **Settings | Data Expirations** page.

If you are going to use this method to delete data, set it up right after installation and then run TIUtilities -e every day. Do not wait until your database is very large and then try to expire a large range of days at once. This will cause your database to run out of memory.

Consider using the TI Purge command line utility rather than Data Expirations. TI Purge is a faster and more thorough way to remove data from your BCI database. Please contact TIBCO Foresight Support for information.

There is no "unexpire."

If you selected SqlServer and Create

BCI created its new, empty database. This database will be current up to the last full release.

- If you are installing a base release, your database is ready.
- If this is a patch, upgrade the database to the patch level by following the directions with the upgrade.

You also have database scripts in Transaction Insight's **DBScripts\SqlServer** directory. These can be run to create a database on another machine, for example. See the Readme.txt in that directory. These scripts will give you a database compatible with the base release, so remember to run the patch script(s) also.

If you selected SqlServer and Use DB

If you selected **Use DB**, you have database scripts in Transaction Insight's **DBScripts\SqlServer** directory. These can be run to create and configure the BCI database. See the Readme.txt in that directory. These scripts will give you a database compatible with the base release.

If you selected Oracle

The installation program does not actually create a database. You have database scripts in Transaction Insight's **DBScripts\Oracle** directory. These can be run to create and

configure the BCI database. See the Readme.txt in that directory. This will give you a database compatible with the base release.

Replacing Licensing Files

You have two license files to replace.

- **license.ini** for Instream
- **TILicense.ini** for all web component applications.

These are available from your TIBCO Foresight account representative

license.ini

To use the proper license.ini file:

1. Open Web.config (in Foresight's TransactionInsight\version\ Environments\envname\TIWeb directory) on the web server machine and search for InStreamDirectory to look up the path to Instream. You can change this to another Instream location if necessary.
2. Put the license.ini file in Instream's Bin directory.

   If this is a new installation and the directory already contains a license.ini file, it is not valid.

TILicense.ini

Copy the TILicense.ini file that your TIBCO Foresight account representative gave you into your **TransactionInsight***version***Environments** *envname***bin** directory. This goes on the server that is running TIUtilities and Importer.

Protecting your Database

- Back up your database frequently and be sure the backups can be used to restore.
- **Never, ever, make manual changes to your BCI database**. Manually changing your database can irrevocably damage it, and TIBCO Foresight may not be able to help you recover.
- Set up permissions that protect it against such changes from others at your company.
- Use TIBCO Foresight-provided scripts, and call TIBCO Foresight if you have database issues or questions.

Starting and Stopping IIS

To start or stop Internet Information Services, go to the command line on the BCI machine and type one of these:

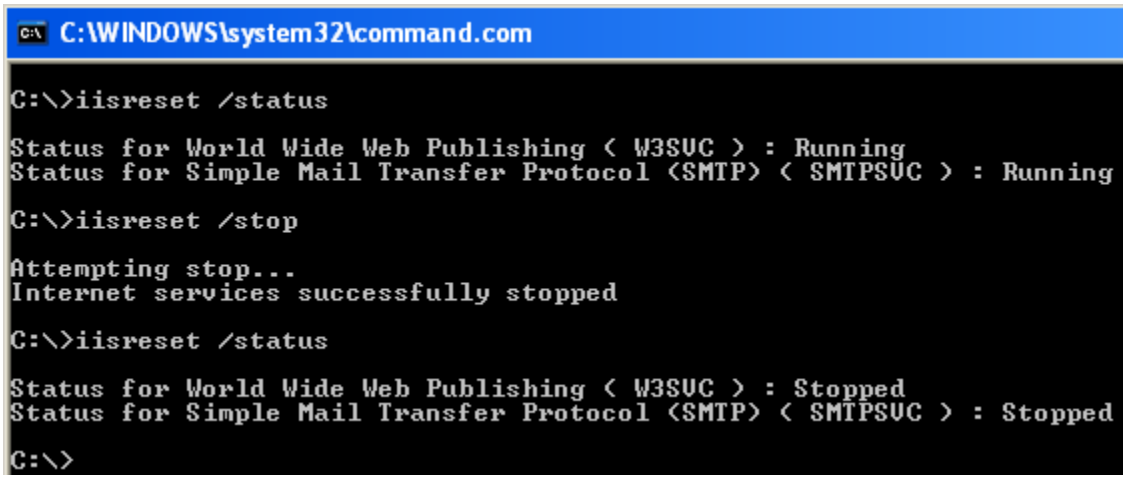
```
iisreset /start

iisreset /stop

iisreset /status
```

Example

This example checks the status of IIS (World Wide Web Publishing) and finds it running. It then stops it and checks the status again.

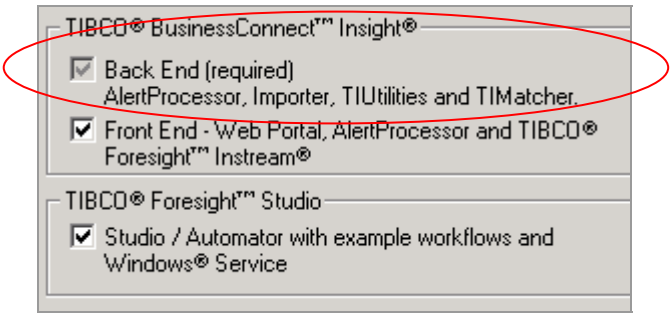


7 Appendix A: Components Choices during Installation

BCI Components

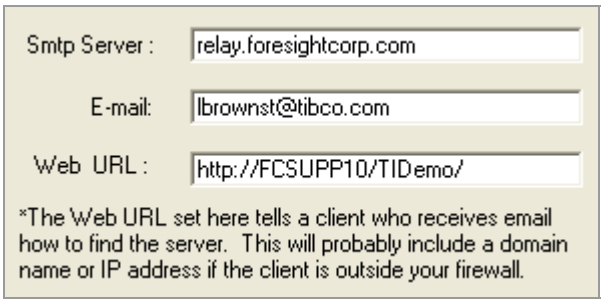
Back End

Used when importing, summarizing and matching data in BCI. This can (and should) be used on a machine that is separate from the web server.



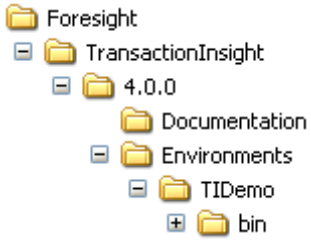
Information to enter:

1. Name for environment.
2. Information for TI Web and TIUtilities, and TIMatcher:



What was Installed

Directories



TI's bin directory contains

- AlertProcessor
- Importer
- TIUtilities
- TIMatcher

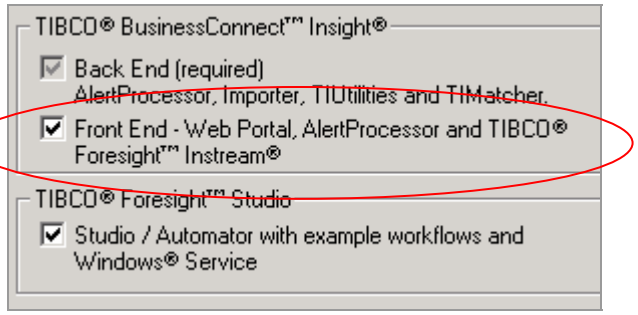
Databases

- None

Front End

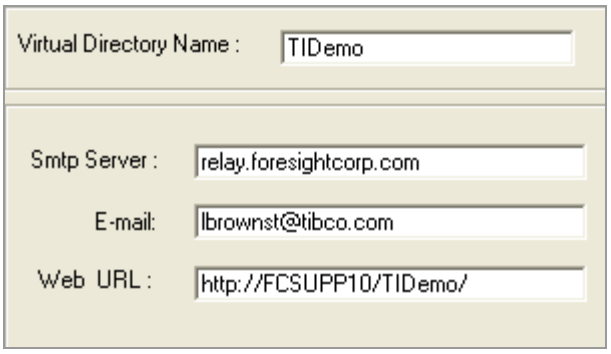
(Requires Back End)

The web portal (or GUI) used when interacting with BCI. Your web browser connects to this server to perform BCI operations such as viewing documents, transmissions and statistics.



Information to enter:

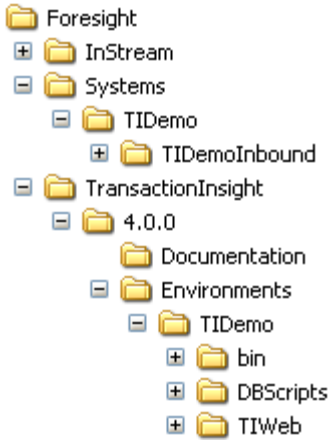
1. Name for environment.
2. Location of existing Instream or location for new installation of Instream.
3. Information for TI Web:



What was Installed

(Versions may differ.)

Directories



TI's bin directory contains

- AlertProcessor
- Importer
- TIUtilities
- TIMatcher

Databases

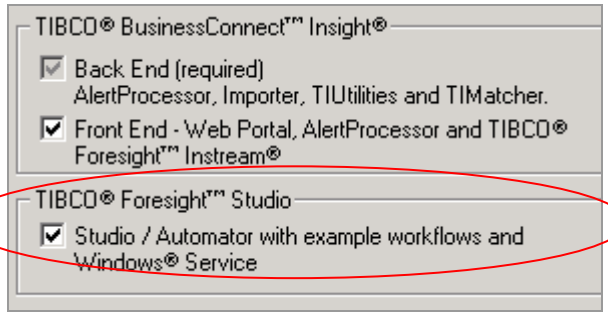
- TI database

Foresight Automator / Studio and Foresight Service Components

These components orchestrate the daily operations of TI. Foresight Service runs Automator which in turn runs a workflow containing the various BCI components.

Where to install:

- Put these components on your back end machine to run Importer, TIUtilities, TIMatcher, AlertProcessor, and other programs.
- You may also want them on your front end machine to help with miscellaneous tasks such as moving EDI data files across your network.

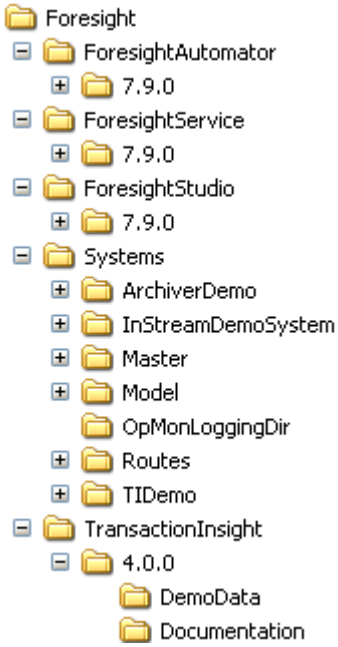


Information to enter: Name for environment.

What was Installed

(Versions may differ.)

Directories



Databases

- None

8 Appendix B: Windows Integrated Security

A typical installation creates a user ASPNET and gives it access to the BCI database. The password for ASPNET is stored in the Web.config file on the web server

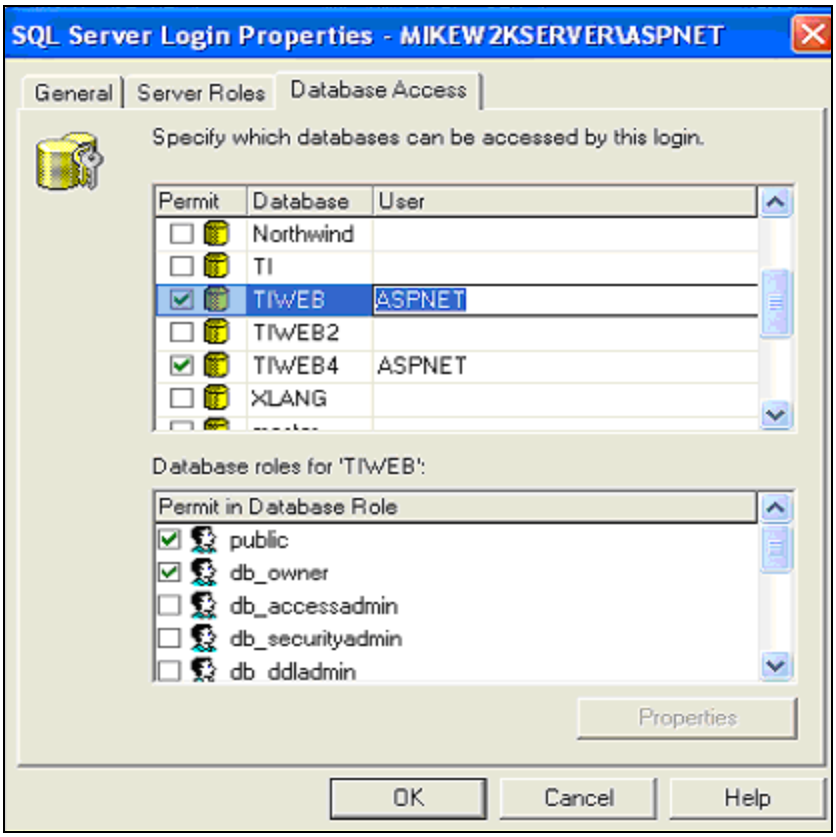
You can avoid storing this sensitive information in Web.config by using Windows Integrated Security. In that case, the username will be the one that is running the web server.

You will need to give that username access to the BCI database:

1. Edit the XML file Web.config on the web server machine.

   Look for <appSettings> and replace the UID and PWD entries with Integrated Security=true

2. On the database machine, check SQL Server Login to be sure that the user that will run the web server has owner privileges for the BCI database. In this example, the BCI database is called TIWEB and the user running the web server is ASPNET.


SQL Server Login Properties - MIKEW2KSERVER\ASPNET
General
Server Roles
Database Access
Specify which databases can be accessed by this login.
Permit
Database
User
Northwind
TI
TIWEB
ASPNET
TIWEB2
TIWEB4
ASPNET
XLANG
Database roles for 'TIWEB':
Permit in Database Role
public
db_owner
db_accessadmin
db_securityadmin
db_ddladmin
Properties
OK
Cancel
Help

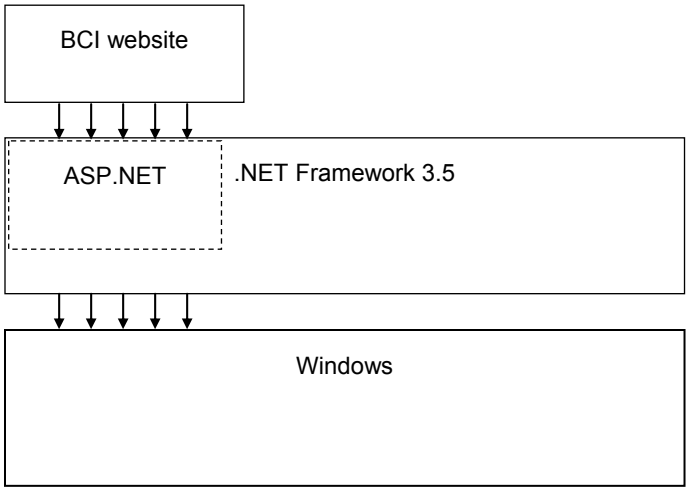
9 Appendix C: .NET

Overview of .NET and BCI

Microsoft's .NET Framework 3.5 must be installed on the web server before you install BCI.

The .NET Framework is Microsoft's managed code development platform for creating secure, high-quality software applications.

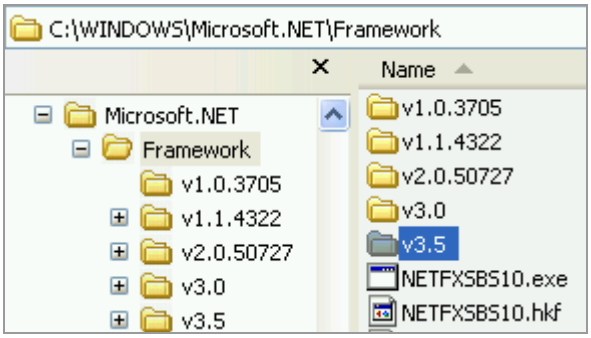
BCI is built using ASP.NET, a component of .NET Framework for developing web sites, web applications, and web services.



Checking for .NET 3.5

.NET redistributable package is required on the BCI web server.

To see if .NET version 3.5 or later is installed, look in your Windows or WINNT directory for **Microsoft.NET\Framework\v3.5**:



Installing .NET 3.5

You can download the **.NET Framework Version 3.5 Redistributable Package** from http://www.microsoft.com/downloads and install it.

10 Appendix D: Network Load Balancing

Overview

You can set up multiple servers that work together to provide better performance and reliability, while appearing to be a single system to your BCI users.

In Windows, you can use Microsoft Network Load Balancing (NLB), where each host runs separate copies of the server application. Incoming requests are distributed among the clustered servers.

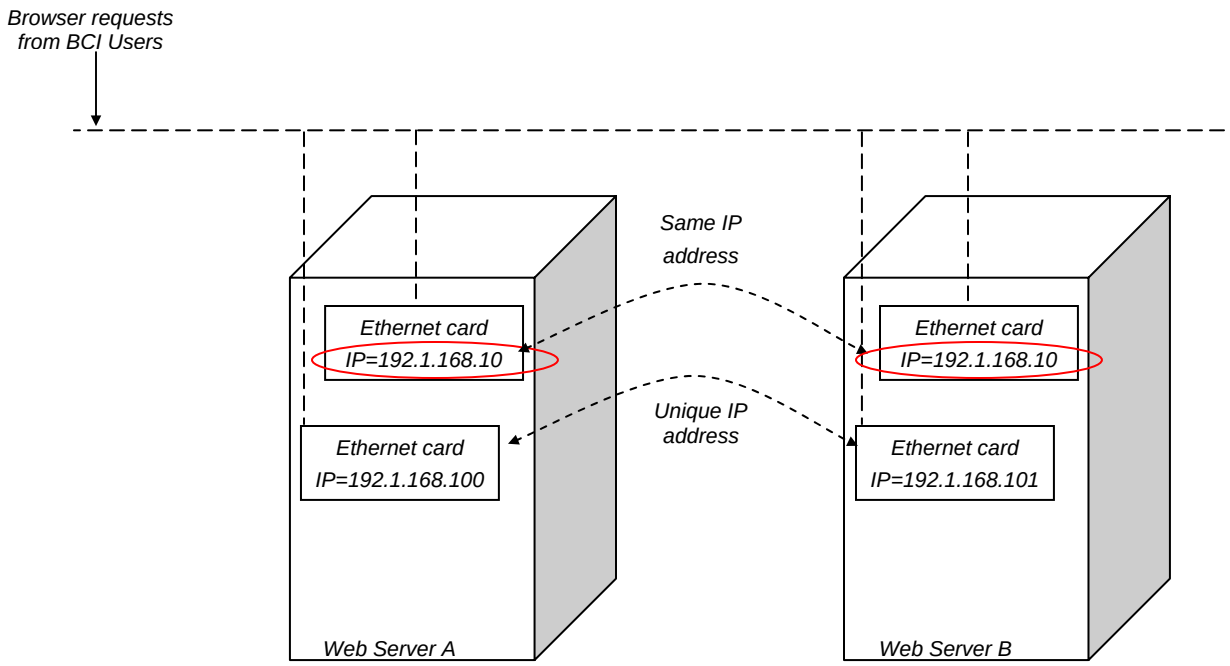
When a host fails or goes offline, active connections are lost but the new incoming load is automatically redistributed among the other hosts. A drawback: If a server has failed but is still responding to the network, the NLB system will continue to send users to it.

This method requires two network cards on each machine. One card on each machine must be capable of NLB.

All of the computers in the cluster have the same cluster IP address – the one that maps to the Internet address used by BCI browsers.

Each server also retains its own unique, dedicated IP address.

You can cluster up to 32 servers, which must all be located on the same subnet.



Preparing to Set up NLB

Important

Set up NLB before installing BCI

1. Install all BCI prerequisites on the machines.
2. Request a cluster IP address and create a DNS record for it.
3. Install IIS on each machine:

   Log on as the domain administrator on both web server machines.

   Go to Control Panel's Add/Remove Programs and choose Add/Remove Windows Components.

   Select Application Server (or IIS) / Details.

   Select ASP.net, IIS, and Com+.
4. Run IIS Administrator under Control Panel | Administrative Tools | Internet Information Services (IIS) and ensure that ASP.NET is enabled.
5. Set up NLB on each machine (see below)

After setting up NLB, you are ready to run the BCI installation program.

Setting up Network Load Balancing

1. Set up the NLB cluster network card on each machine:

a. Under **Control Panel | Network Connections**, select the card to be used for NLB (suggestion: rename it NLB to avoid confusion later) and choose **Properties**.

b. Make sure only Network Load Balancing and Internet Protocol are checked.

c. Edit the properties of Network Load Balancing and set up the three tabs as follows:

   Cluster Parameters – set up information for the clustered IP address including its IP address, subnet mask, and full Internet name. For Cluster operation mode, select **Unicast**. This information will be the same on all clustered machines.

   Host Parameters – set up information for the local IP address, including IP address, subnet mark, and priority. Priority is a unique integer assigned to each machine in the cluster. It determines the order in which requests are delivered.

   Port Rules – check the settings. Single affinity is recommended so that users can reconnect to the same session if they are disconnected.

2. Return to the NLB properties and edit the properties of **Internet Protocol**.

   Select **Use the following IP addresses**.

   Insert the NLP cluster IP, subnet mask, and default gateway.

   Insert the DNS server addresses.

3. Verify the setup by going to the command line and running **ipconfig /all.** You should see the dedicated IP and clustered IP addresses.

Connecting to the NLB Cluster

Direct browsers to the shared IP address or the Internet address that points to it

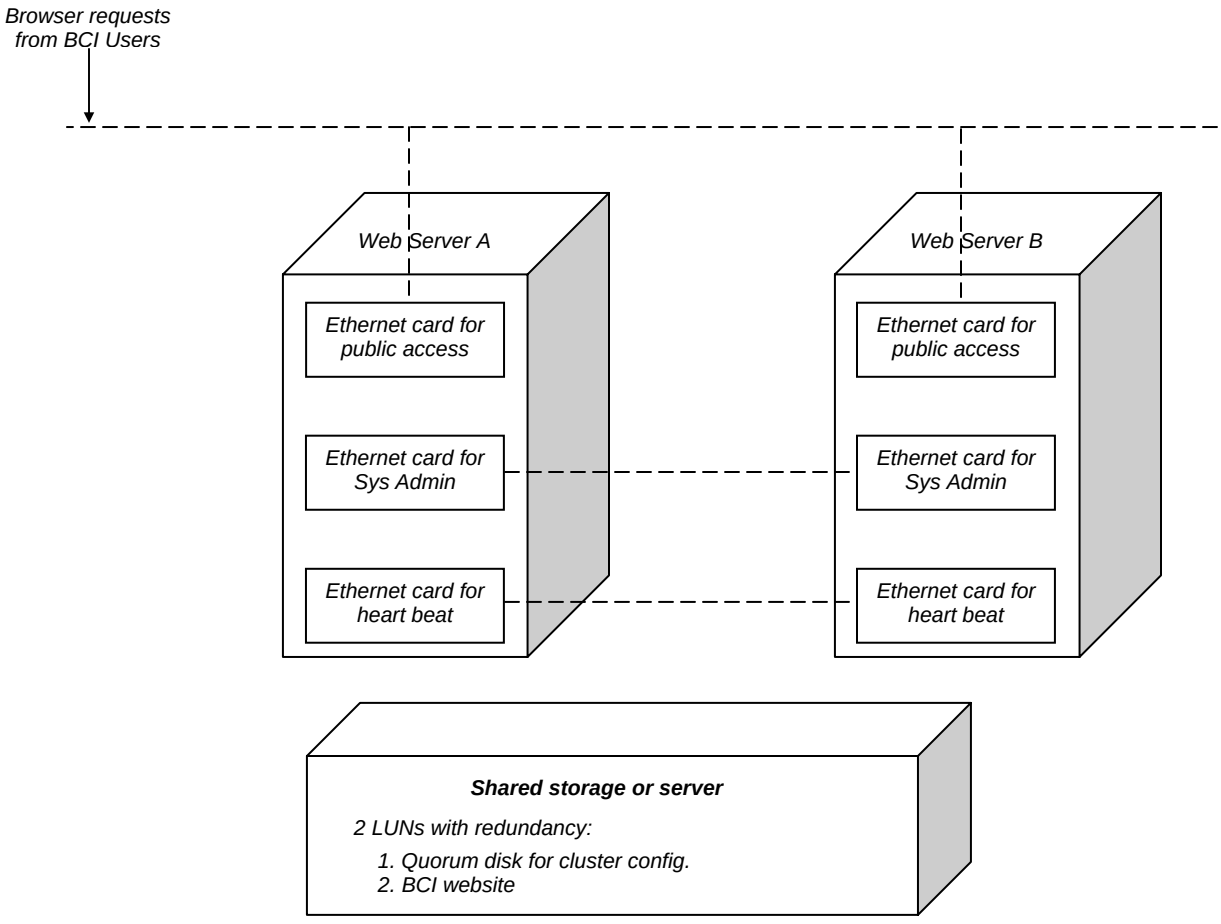
.

11 Appendix E: Network Clustering

Overview

Microsoft Windows Server 2003 clustering provides failover – when the primary server fails, a backup server takes over with no loss of data or functionality.

Example Configuration



When web server A is active, web server B is not responding to browser requests, but it is tracking all actions by web server A.

When web server A fails, web server B immediately takes up where A left off.

On the public Ethernet card, each server has its own IP address.

In addition, the active machine has a cluster IP address and a BCI cluster IP address. These two addresses are attached to the active server and move when a fail over occurs.

Example

Web server A is the primary server for handling browser requests from BCI users. Web server B is the backup.

- Both have unique permanent machine IP addresses for the public access Ethernet connection.

  In this example, A has 192.168.1.180 and B has 192.168.1.181.

- In addition, whichever server is active has two additional public IP addresses. When the active server fails, the cluster software automatically moves these to the other server.

  In this example, the addresses 192.168.1.179 and 192.168.1.175 will be on the active server and not on the passive one.

	IP addresses for server A	**IP addresses for server B**
Machine IP address	192.168.1.180	192.168.1.181
Cluster IP address	192.168.1.175 -- points to www.*ourcompany*.com	
BCI cluster IP address	192.168.1.179 – points to www. BCI.*ourcompany*.com	

12 Appendix G: Workflow XML Files

Workflow Files Overview

The BCI installation gives you a set of XML files that define the flow of files through Instream and BCI. They install in Foresight\Systems*system*\Workflows*version*\Automator.

This appendix describes XML files that ship in the sample workflows.

We recommend that you use TIBCO Foresight Studio to modify workflows, but you can also edit the files directly with a text editor like Notepad or with an XML editor. Back up your workflow before making changes in this directory.

Once a workflow has been run, each XML file has a corresponding Error, In, Out, and Process directory:

- Workflows
  - 1.0
    - InStreamDemoWorkflow
      - DocSplitterStep
        - Error
        - In
        - Out
        - Process
      - DocumentSplitReportFolder
        - Error
        - In
        - Out
        - Process
      - InboundDocumentFolder
        - Error
        - In
        - Out
        - Process

When the workflow is running, files move through these directories as follows:

1. A file arrives in a component's In directory.

2. The component moves it to Process, takes action, and creates output files in the Process directory.

3. From there they are moved:

   a. Those that have a destination in the workflow are moved to the **In** directory of the next component in the workflow. This can be defined with a <Target> tag or in the <Task> tag in the XML file.

   b. Those with no destination go to the current component's **Out** directory.

   c. If the component returns an error, all files in the current component's **Process** directory go to the **Errors** directory.

When debugging workflow changes, you can determine how far the files progressed through the workflow by looking in these directories.

If these directories are deleted, they are rebuilt when Automator starts the workflow again.

Important: Do not delete the XML files.

Variables

These variables are available for use in workflow XML files:

Type of Variable	How Defined …	Use in …
Automator Variables	Built in to Automator; values are assigned as Automator processes each component	Any workflow run by Automator
Automator Global System Variables	In WorflowGlobals.xml in Foresight's Systems directory	Any workflow XML file in any system listed in the directory containing WorkflowGlobals.xml
Individual Component Variables	In an individual XML workflow component file	The same individual XML workflow file

Please see **Variables_for_Workflows.pdf** in Studio's Doc directory for details.

XML Files

How to Edit XML Files

The easiest and most accurate way to edit the XML files is by opening the workflow with TIBCO Foresight Studio and making changes there.

You can also use a text editor or XML editor. Be sure that TIBCO Foresight Studio is closed before editing the files this way.

Important Note

This chapter provides all-inclusive information about the XML files contained with BCI. Not all elements described here will apply to your particular system.

workflow.xml

This file lists each component that is used in the workflow.

Overview of Component XML Files

Example

This is a partial component XML file for a workflow:

Component name is InStream

```
  <?xml version="1.0" encoding="utf-8" ?>
- <FSDoc>
- <Description>
  <Title>Foresight Studio Schedule Item</Title>
  <Type>Schedule</Type>
  <Version>0.9.9.9</Version>
  </Description>
- <Schedule>
  <Id>InStream</Id>
  <Type>InStream</Type>
- <Connection>
  <Id>Dtl</Id>
  <Type>PatternOut</Type>
  <Required>False</Required>
  <ExpectedFileType>Detail</ExpectedFileType>
  <Target>EDIReport</Target>
  <Target>ImporterM</Target>
  <Target>RespGen</Target>
  <Target>DocSplitter</Target>
  <Param>DTL</Param>
  </Connection>
.
.
.
- <Connection>
  <Id>EDI</Id>
  <Type>SingleFileIn</Type>
  <Required>False</Required>
  <ExpectedFileType>EDI</ExpectedFileType>
  </Connection>
- <Property>
  <Id>Guideline</Id>
  <Value />
  <DefaultValue>PDSA837I</DefaultValue>
  <Description>Guideline to use for validating.</Description>
  <EditType>TextEntry</EditType>
  <PrependValue>-g</PrependValue>
  <AppendValue />
  <NotUsedValue />
  <Required>False</Required>
  </Property>
- <Property>
  <Id>SetupFile</Id>
```

```
  <Value />
  <DefaultValue />
  <Description>Setup (*.apf) file for InStream.</Description>
  <EditType>TextEntry</EditType>
  <PrependValue>-s"</PrependValue>
  <AppendValue>"</AppendValue>
  <NotUsedValue />
  <Required>False</Required>
  </Property>
.
.
.
- <Trigger>
  <Id>FileToValidate</Id>
  <Type>FileAvailable</Type>
  <Param>In</Param>
  </Trigger>
- <Task>
  <Id>RenameToEDI</Id>
  <Type>Executable</Type>
  <SuccessCode>0</SuccessCode>
  <Param>cmd</Param>
  <Param>/c "ren "%FSPROCDIR%%FSINFILE%%FSINEXT%"
"%FSINFILE%.edi""</Param>
  </Task>
  <Task>
  <Id>HVInStream</Id>
  <Type>Executable</Type>
  <SuccessCode>100</SuccessCode>
  <Param>C:\Foresight\InStream\Bin\HVInStream.exe</Param>
  <Param>-i"%FSPROCDIR%%FSINFILE%.edi"</Param>
  <Param>-o"%FSPROCDIR%%FSINFILE%"</Param>
  <Param>%Guideline%</Param>
  <Param>%SetupFile%</Param>
  <Param>%UseTableFileServer%</Param>
  <Param>-r</Param>
  <Param>-a</Param>
  </Task>
- <Layout>
  <XPos>400</XPos>
  <YPos>550</YPos>
  <ImageIndex>2</ImageIndex>
  </Layout>
  </Schedule>
  </FSDoc>
```

Defines when the component activates (when a file arrives in the component's In folder).

*.Component first executes a **ren** command to change the file extension to "edi"*

Component next executes InStream using variables defined by Foresight and variables set up with Properties tags

Major Elements

Description	Automator ignores this element. It is for your information.
Schedule	Contains information needed by Automator to run the workflow.
Connection	Defines how data flows between this component and other components in the workflow.
Property	Defines a variable for use in the Task or Trigger element.
Trigger	Defines when the component activates.
Task	Defines the command that the component runs. A component can have one or more tasks.

Layout	Ignored by Automator. It is used by Studio to lay out the workflow display.

Connection Element Details

This element defines the input and output connections for this component.

Id	Name of the connection.
Type	File type of the files that are moved in or out. See the Input Connection Tab and Output Connection Tab sections of **ForesightStudio.pdf** in Studio's Doc directory for details.
Success Code	Code that indicates success. For Instream products, this is 100. For other products, refer to their documentation.
Parameters	The file type to be moved to the next component for Output connections.

Property Element Details

The Property element sets up a variable that is used in the Task or Trigger element. Please see **Variables_For_Workflows.pdf** in Studio's Doc directory for details.

Trigger element details

The Trigger element defines what activates the component. See the Trigger Tab section of **ForesightStudio.pdf** in Studio's Doc directory for details.

Type	How the component activates, one of these:
	
	`FileAvailable`
	`FileNameMatch`
	`SimpleTimer`

Parameters	This defines when causes activation for the Type and varies by component.

Task element details

The Task element defines what the component does when it activates

Id	The name of the task.
Type	Several types available. See **ForesightStudio.pdf** in Studio's Doc directory.
SuccessCode	Return code that the application uses for success.

Parameters	For executables, the first parameter is the command line. Additional Param elements supply the command-line parameters to the executable. For other types, the parameter meanings vary.

Alerter.xml

What it does	Runs Alerter (AlertProcessor.exe) on the BCI database to detect situations under which BCI alerts should be sent out, and then sends out the e-mail alerts.
Configuring	The administrator for BCI sets up alerts on BCI's **Alerts** page. The BCI database and other information is identified in **AlertProcessor.exe.config**. See AlertProcessor on page 28.

Elements of Interest	
Tag	**Description**
Trigger	30-second timer.
Task	Path to AlertProcessor.exe

Bad.xml

What it does	Moves files with file type **bad** from its **In** directory to Foresight\Systems*system*\Root\EDIBad. Typically, this moves DocSplitter invalid output files.

Elements of Interest	
Tag	**Description**
Directory	Where the file is moved.
Trigger	Arrival of a file in its **In** directory (Foresight\Systems*system*\Workflows*version*\ Automator\Bad\In).
Task	The Move command. Directory is defined earlier in the XML file. Other variables are described on page 54.

DocSplitter.xml

What it does

Runs DocSplitter to split an EDI file into valid and invalid EDI.

Produces a valid file with extension **god**, an invalid file with extension **bad**, and an XML report with extension **XML**.

Elements of Interest	
Tag	**Description**
Connection	Extensions of files coming in to the DocSplitter component: **Good** – sends files with extension **god** to the Good component. **Bad** – sends files with extension **bad** to the Bad component. **Report** – sends files with the extension defined in the %ReportExtension% variable to the **ResultsOut** component. DTLIn and EDIIn take in files.
Trigger	Arrival of a detail file with extension **dtl** and an EDI file with extension **edi** in its **In** directory (Foresight\Systems*system*\Workflows*version*\Automator\DocSplitter\In).
Task	The DocSplitter command. Variables in Caps and lower case are described earlier in the XML file. Other variables are described on page 54. The DocSplitter command format is described in **DocumentSplitterTechnicalManual.pdf** in Instream's Doc directory.

EDIIn.xml

What it does Moves files with extension **edi** from Foresight\Systems\ *system*\Root\In to the Instream component.

Properties of Interest	
Tag	**Description**
Connection	**Out** – sends files with the extension in variable **FileExtension** to the RenameToEdi component. **Directory** – defines where to find the input files.
EDIInDir	A variable containing "`Foresight\Systems\`*`system`*`\Root\In`", used in Task.
Trigger	Timer using the value defined in the **Frequency** variable.
Task	The Move command. The **Directory** variable is defined earlier in the XML file. Other variables are described on page 54.

EDIReport.xml

What it does Takes in a Detail file and an EDI file and produces a HTML report that shows all segments and the errors that go with them.

Elements of Interest	
Tag	**Description**
Connections	Takes in files with type **EDI** and **DTL**. Creates files with type **HTML** (EDI report)
Trigger	Arrival of a file with extension **dtl** and another file with extension **edi** in Foresight\Systems*system*\Root\In.
Task	Executes **EDIDiagnostician** to create the HTML file. Other variables are described on page 54.

EDIResp.xml

What it does Takes in a file and moves it to Foresight\Systems*system*\Root\Out.

Elements of Interest	
Tag	**Description**
Connections	EDIResp accepts any file.
Directory	Where the file is moved: Foresight\Systems*system*\Root\ Out.
Trigger	Arrival of a file in its **In** directory.
Task	Executes a **Move** command to move the file to Foresight\Systems*system*\Root\Out. Directory is defined earlier in the XML file. Other variables are described on page 54.

Good.xml

What it does Moves files from its **In** directory to Foresight\Systems*system*\Root\EDIGood. Typically, this moves DocSplitter valid output files.

Elements of Interest	
Tag	**Description**
Directory	Variable specifying where the file is moved: Foresight\Systems*system*\Root\EDIGood.
Trigger	Arrival of a file in its **In** directory (Foresight\Systems*system*\Workflows*version*\ Automator\Good\In).
Task	Executes a **Move** command to move files to the path specified with the Directory variable. Other variables are described on page 54.

ImporterM.xml

What it does Imports validation detail and EDI files into the BCI database.

Elements of Interest	
Tag	**Description**
Connections	**DTLIn** – accepts files with extension DTL. **EDIIn** – accepts files with extension EDI. These files remain in ImporterM's **Out** directory.
Direction	Variable that specifies whether the data is considered i**nbound** or **outbound** from the point of view of BCI.
Trigger	Arrival of a file with extension **dtl** and another file with extension **edi** in ImporterM's **In** directory.
Task	Executes **Importer**. For details on Importer's command line format, see **Importer.pdf** in TransactionInsight's Documentation directory. Direction is defined earlier in the XML file. Other variables are described on page 54.

InStream.xml

What it does Validates an EDI file and produces reports. Renames the EDI file to have file extension EDI.

Elements of Interest	
Tag	**Description**
Connections	**Detail** – sends files with extension DTL to several components. **Sum and Rpt** – there are no target components for files with extension SUM or RPT so they remain in Instream's Out directory. **SourceEDI** – sends files with extension EDI to several components. **EDI** – accepts files with any extension.
Guideline	Specifies the guideline to use for validation.
SetupFile UseTableFileServer	These will not appear on the command line being assembled under Tasks, once the variables are evaluated. This is because Value and DefaultValue are both empty, and Value matches NotUsedValue. ↓
Trigger	Arrival of a file in Instream's **In** directory.
Task	Renames the EDI file extension to EDI and executes Instream. Variables are defined earlier in the file or are described on page 54. For details, see **InstreamValidationTechnicalManual.pdf**, in Instream's Doc directory.

RenameToEDI.xml

What it does Changes file extension to **edi**.

Elements of Interest	
Tag	**Description**
In and FromName	Accepts files with any extension.
Out and ToName	Outputs files with extension changed to EDI to the next component (Instream).
Trigger	Arrival of a file in RenameToEDIi's **In** directory.
Task	Executes a **ren** command that renames all file extensions to edi.

RespGen.xml

What it does Executes Response Generator and copies output to a specified directory.

Elements of Interest	
Tag	**Description**
Connections	Response Generator takes in a validation detail report (extension Detail)
DeleteZeroLengthResults ApplicationReceiverID ApplicationSenderID ApplicationVersion …	These will not appear on the command-line since their Value and NotUsedValue match.
Other properties	A number of other properties are defined and will be used as parameters on the Response Generator command line unless their Value and NotUsedValue match.
StartingST02 StartingGS06 InterchangeControl-NumberStart	Control numbers start with 1.
Trigger	Arrival of dtl file.
Task	Executes Response Generator. See **ResponseGeneratorTechnicalManual.pdf**, in Instream's Doc directory.

ResultsOut.xml

What it does Moves all files from its In directory to a specified directory.

Elements of Interest	
Tag	**Description**
In	Takes in any file extension.
Directory	Variable specifying where output goes: Foresight\Systems*system*\Root\Out
Trigger	Arrival of any file in its **In** directory.
Task	Executes a **move** command.. Directory is defined earlier in the XML file. Other variables are described on page 54.

TIUtilities_5MinUpdate.xml

What it does Assigns tasks to teams, sends corresponding e-mails, updates the summary information on the Errors and Transmissions pages.

Elements of Interest	
Tag	**Description**
EnableLogToConsole	This will not enable console logging because Value matches NotUsedValue. This corresponds to TIUtilities' **–tc** command-line parameter.
LogfileDirectory	The log goes to the component's Process directory. This corresponds to TIUtilities' **–td** command-line parameter.
LoggingLevel	Errors are logged. This corresponds to TIUtilities' **–tl** command-line parameter.
UpdateSummaryTables	Updates statistical tables in TI. This corresponds to TIUtilities' **–s** command-line parameter.
SummarizeDocuments	This will not update summary tables since Value and NotUsedValue match. This corresponds to TIUtilities **–sd** command-line parameter but not set.
SummarizeErrors	The component updates the top error counts on the Errors page. This corresponds to TIUtilities' **–se** command-line parameter.
Summarize Transmissions	This updates the statistics at the top of the Transmissions page. This corresponds to TIUtilities' **–st** command-line parameter.
GeneratePartnerFilters	This will not generate partner filters since Value and NotUsedValue match. This corresponds to TIUtilities' **–f** command-line parameter.
ExpireOldData	This will not expire old data, since Value and NotUsedValue match. It corresponds to TIUtilities' **–e** command-line parameter.
EMailSenderAddress	The "from" address for BCIe-mails. This corresponds to TIUtilities' **–from** command-line parameter.
SmtpServer	The SMTP server for e-mails. This corresponds to TIUtilities' **–smpt** command-line parameter.

Elements of Interest	
Tag	**Description**
DBConnectString	The connection string to access the database. See your database administrator for these settings. Example if DBType is ODBC: DRIVER={SQL Server};SERVER=(local); DATABASE=TI231SQL;UID=sa;PWD=sa Example if DBType is Oracle: DATABASE=ora901_bigbadhp;UID=TI231QA;PWD= TI231QA This corresponds to TIUtilities' **–db** command-line parameter.
DatabaseType	The type of database used by TI, one of these: **odbc** **oracle** This corresponds to TIUtilities' **–dbtype** command-line parameter.
URL	The URL to access the BCI web portal Example: `http://KAVERTEST/TIDemo//` This corresponds to TIUtilities' **-url** command-line parameter.
IntervalSeconds	The component runs every 300 seconds.
Trigger	This component executes on a timer. The frequency is defined with IntervalSeconds.
Task	The TIUtilities command. Variables are defined earlier in the XML file.

TIUtilities_6HrUpdate.xml

What it does Generates dynamic partner filters. This is best done infrequently and at a time when the database is not busy.

Elements of Interest	
Tag	**Description**
EnableLogToConsole	This will not enable console logging because Value matches NotUsedValue. This corresponds to TIUtilities' **–tc** command-line parameter.
LogfileDirectory	The log goes to the component's Process directory. This corresponds to TIUtilities' **–td** command-line parameter.
LoggingLevel	Errors are logged. This corresponds to TIUtilities' **–tl** command-line parameter.
Other properties	Properties where Value matches NotUsedValue are not included on the command line.
GeneratePartnerFilters	This generate dynamic partner filters. This corresponds to TIUtilities' **–f** command-line parameter.
DBConnectString	The connection string to access the database. See your database administrator for these settings. Example if DBType is ODBC: DRIVER={SQL Server};SERVER=(local); DATABASE=TI231SQL;UID=sa;PWD=sa Example if DBType is Oracle: DATABASE=ora901_bigbadhp;UID=TI231QA;PWD= TI231QA This corresponds to TIUtilities' –db command-line parameter.
DatabaseType	The type of database used by TI, one of these: **odbc** **oracle** This corresponds to TIUtilities' **–dbtype** command-line parameter.
IntervalSeconds	This component runs every 21600 seconds.
Trigger	This component executes on a timer. The frequency is defined with IntervalSeconds.
Task	The TIUtilities command. Variables are defined earlier in the XML file.

TIUtilities_12HrUpdate.xml

What it does Updates the summary statistics on the Document, Document Volumes, and Success Rate pages.

Elements of Interest	
Tag	**Description**
EnableLogToConsole	This will not enable console logging because Value matches NotUsedValue.
LogfileDirectory	The log goes to the component's Process directory. This corresponds to TIUtilities' **–td** command-line parameter.
LoggingLevel	Errors are logged. This corresponds to TIUtilities' **–tl** command-line parameter.
SummarizeDocuments	The component updates the statistics on the Document, Document Volumes, and Success Rate pages. This corresponds to TIUtilities' **–sd** command-line parameter.
DBConnectString	The connection string to access the database. See your database administrator for these settings. Example if DBType is ODBC: DRIVER={SQL Server};SERVER=(local); DATABASE=TI231SQL;UID=sa;PWD=sa Example if DBType is Oracle: DATABASE=ora901_bigbadhp;UID=TI231QA;PWD= TI231QA This corresponds to TIUtilities' –db command-line parameter.
DatabaseType	The type of database used by TI, one of these: **odbc** **oracle** This corresponds to TIUtilities' **–dbtype** command-line parameter.
IntervalSeconds	This component runs every 43200 seconds.
Other properties	Properties where Value matches NotUsedValue are not included on the command line.
Trigger	This component executes on a timer. The frequency is defined with IntervalSeconds.
Task	The TIUtilities command. Variables are defined earlier in the XML file.

13 Appendix H: Database Scripts

Knowledge Required

This section is written for a database administrator.

Setting up *new* Databases

Caution These scripts are only for setting up a NEW database.

The installation program creates scripts for setting up new databases for TI.

You have two methods for creating the databases:

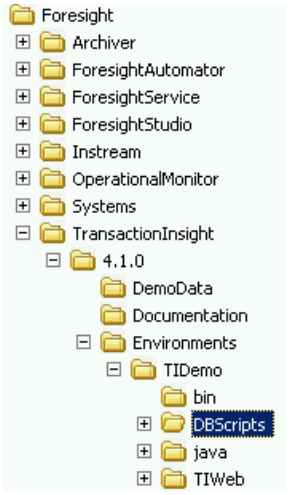
- For SQL Server, the installation program will actually create and customize the database right from the installation program, if you select "create" database during installation. You will not have to run any scripts.
- You can run the scripts yourself, before or after running the installation program. In this case, you should tell the installation program to "use" the database.

To run the scripts yourself:

1. Give the exact names and server locations to the person who will be installing the BCI applications with instructions to "use" the databases rather than "create" them.
2. Create empty databases for BCI.

3. Run the scripts listed in the Database Scripts chart below. They are under these directories:

TI DB Scripts



Database Scripts		
Step	**Script**	**Notes**
TI Database Scripts		
1	CreateDB_4.1.0.sql	(If BCIdatabase does not exist) This can be used to create the empty BCI database. Edit the script and adjust the database name if desired.
2	CreateSchema_4.1.0.sql	This creates the tables and adds basic information needed to run TI: root user, root team, etc.
3	ProcsAndViews_4.1.0.sql	This adds stored procedures, functions, and views.
4	TI_ErrorDefinition_UpdateAnd InsertMissing_MSSQL_4.1.0.sql or … TI_ErrorDefinition_UpdateAnd InsertMissing_ORACLE_4.1.0.sql	This inserts the error messages for the Instream version being used. Open the script and check the version. The error message scripts for different Instream versions are available at http://foresight.TIBCO.com. Click Downloads, login, click Instream and then Instream for Windows and look for “Error Message Update For Transaction Insight to Instream*n.n.n*.” If your version is not there, contact TIBCO Foresight support.

14 Appendix I: Instream and TI

Executable	HVInStream.exe
Configuration files	APF files, guidelines, various INI files, CSV files for partner automation
Default location	Selected during installation
Documentation	InstreamValidationTechnicalManual.pdf in Instream's Doc directory

Validation Instream

Before being imported into BCI, EDI is validated with Instream and the detail file sent to Importer.

These items are used to populate drop-down lists on the forms and on the Code Lookup page.

Updating error messages in the BCI database

This section tells you how to end up with the proper error messages within BCI for the version of Instream you are using.

To install the validation error messages into the BCI database, execute the version of this script that matches the Instream that you are using:

TI_ErrorDefinition_UpdateAndInsertMissing_MSSQL.sql.

If you are using the version of Instream that ships with this installation of TI, use the script that is included with the installation. If you are using another version of Instream:

1. Go to http://foresight.TIBCO.com, click **Support | Product Downloads Login**, and log in. If you do not have a login, contact TIBCO Foresight Support.

2. Choose Instream | Instream for Windows.

3. Download the update under **Error Message Update For Transaction Insight to Instream** *n.n.n*, where *n.n.n* is your version of Instream.

Follow the directions in Readme.txt.

If you are using a version that is not available on the website, contact TIBCO Foresight Support for your version's Error Message Update for Transaction Insight packet. It contains a readme with step-by-step instructions.

Every time you update Instream, follow this process.

Your own Custom Error Messages

If you will be importing documents that have been validated with your own guidelines, and those guidelines use custom error messages, please read **ManagingCustomErrorMessages.pdf** in Instream's Doc directory. You will need to get an Oracle or SQL Server script to run after you have finished setting up your database. This will add your custom error messages into TI.

15 Appendix J: Troubleshooting

General Login Page Problems

- Be sure that IIS is running (see page 36).
- You may have to add Default.aspx to the URL.
- Open Web.config for that environment (under TI's Environments*envname*\ TIWeb directory) and check the WebApplicationUrl setting.
- In Control Panel, choose **Administrative Tools | Internet Information Services** and be sure that ASP.NET is set to Allowed.

HTTP Error 403 – Access Denied

If users outside of your domain are being denied access with a HTTP Error 403, be sure that these match:

- IIS username and password for anonymous access
- The web server's local Windows username and password for anonymous access

This should be done by a network administrator. The general areas to check are as follows.

To set the IIS anonymous username and password:

1. Go to Control Panel | Administrative Tools| Internet Information Services and expand the local computer and Web Sites.
2. Right-click on the BCI website (usually Default Web Site) and choose Properties.
3. Choose the Directory Security tab and edit Anonymous access and authentication control.
4. Turn off Allow IIS to control password.
5. Note the username and type a password for the local machine's anonymous access.

To set a matching username/password for the local machine's anonymous username and password:

1. Go to Control Panel | Administrative Tools| Computer Management and expand Local Users and Groups.
2. Click Users.
3. Make sure the Internet Guest Account has the same username as the one in IIS.
4. Right-click on the Internet Guest Account and chose Set Password. Type the same password.

Problems with Global.aspx

To fix problems with Global.aspx:

1. From the command prompt, under WINDOWS or WINNT, go to the Microsoft.NET\Framework\v2.0.50727.
2. Run this: aspnet_regiis –iru
3. Run this: aspnet_regiis –ga ASPNET
4. Run this: aspnet_regiis -s W3SVC/1/ROOT/SampleApp1

   Where SampleApp1 is your BCI website

Example: For BCI website http://TI-TEST/TIprototype, this command would be:

aspnet_regiis -s W3SVC/1/ROOT/TIprototype

You can look up your BCI website in Web.config:

<add key="WebApplicationUrl" value="http://TI-TEST/TIprototype//" />

Problems Starting TIUtilities

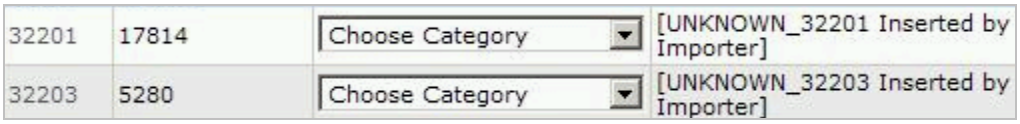
See **TIUtilities.pdf** in the TransactionInsight Doc directory.

Importer Problems

See **Importer.pdf** in the TransactionInsight Doc directory.

Custom Error Message Problems

Custom error messages (in the 32000-32999 range) that have not been defined in the BCI database will display as UNKNOWN on the Top 10 Errors page:



You will need to import the custom error messages into your BCI database. Please see **ManagingCustomErrorMessages.pdf** in Instream's doc directory.

On other pages, custom errors will display as they appear in the imported DTL file.

Database Problems

Never do This

Never modify your DB directly in any way, including with a script. This can cause problems that cannot be fixed. Contact TIBCO Foresight Support.

If you wish to extract data from the database, the BIBridge tool is meant to accomplish this. Contact your TIBCO Foresight account executive or TIBCO Foresight Support for details.

Improving Database Performance

Databases use "statistics" to determine the best algorithm for any operation and thereby affect database performance. Statistics should be "gathered" regularly for optimal performance. These example commands will manually update statistics for your database and may improve performance.

SQL Server example

```
-- Update statistics for ALL tables in current DB
EXEC sp_updatestats
```

Oracle example

```
-- Update statistics for ALL tables in schema "SOMESCHEMA"
begin
dbms_stats.gather_schema_stats(ownname=> 'SOMESCHEMA' , cascade=>
TRUE);
end;
/
```

16 Appendix K: Customizing Web Page Styles

Style Overview

You can customize the look of your BCI web pages.

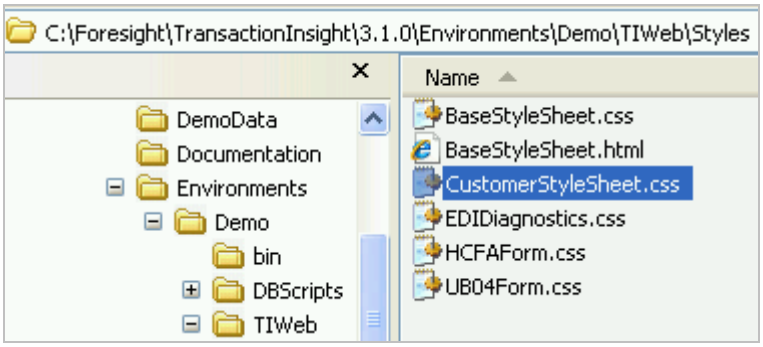
This is a global change for all web pages in the same environment.

To make style changes, you will need:

- A designer to determine what the pages should look like
- A person with knowledge of CSS
- A person who understands the BCI directory structure

Using Style Sheets

The file that you will edit to make style changes is CustomerStyleSheet.css in the Environment's TIWeb\Styles directory:



Any changes that you make to this file will override the default settings in the other CSS files. It contains many examples, commented out by /* and */

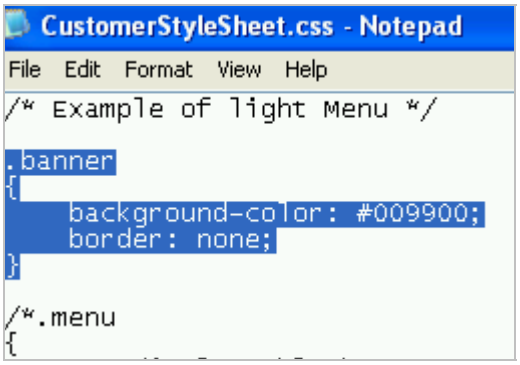
Using files in this directory:

- You can view the contents of the other CSS files in that directory, but do not change them.
- BaseStyleSheet.html is a chart for your reference. It shows what and how CSS styles are used in TI.

Example

To change the banner color to green:

1. Edit CustomerStyleSheet.css.
2. Add this text to an area that is not commented out:



3. Save.
4. Log in to BCI or refresh the page if already logged in.

17 Appendix L: Web.config Settings

Settings in Web.config file

Please see TIWeb on page 34 for overview information.

Important Note

This chapter provides all-inclusive information about Web.config file settings. Not all elements described here will apply to your particular system. Work with your TIBCO Foresight support representative if you have questions.

Settings in Web.config

The web application uses the standard ASP.NET configuration file format. Your customizations are in the <appSettings> section.

<appSettings> section These are user-configurable and specific to TI	
Setting	**Explanation/Value**
Auditing	If `true`, the event log will track many user actions. This greatly increases the size of your event log. See the Audit event type under Settings | Event Log in the BCI Web component.
CollectorDirectory	Directory containing the Importer.exe used after revalidation of corrected data.
CustomDefaultPage	If on, you can add information to the main page of the web user interface. The changes go in CustomDefault.html. See Appendix K: Customizing Web Page Styles on page 76.

<appSettings> section These are user-configurable and specific to TI	
Setting	**Explanation/Value**
DaysOfPasswordPeriod	**No longer needed**. See Password changing, expiration and deactivation in the **BCI_Admin*nnn*.pdf** in the TransactionInsight Doc directory.
DaysOfReusePasswordAging	**No longer needed**. See Password changing, expiration and deactivation in the **BCI_Admin*nnn*.pdf** in the TransactionInsight Doc directory**.**
DB:TransactionInsight	Connection string to the BCI database.
DBType	Value is `SqlServer` or `Oracle`.
DisplayCommonAdministrationPortal	If set to true, Common Administration will appear on the portal for those who have a role with Admin permission.
DisplayTransactionInsightPortal	If set to true, BCI will appear on the portal for those who have a role with Statistics and Files permissions.
DocumentFileSize	Determines the maximum number of characters allowed in a document. If the document contains more characters than this number, an error message alerts that the document is too large to be displayed. The default value is 10000. Note that setting this to a large number will cause the page to load slowly or not at all.
ExclusionSetting	If set to `on`, error exclusion is enabled. This excludes an admin-created list of error numbers from force through.
FormErrorCategoryThreshold	Minimum error level to display on a form. Value can be: Ignore Information Warning Error (the default)
FormViewerCharacter	This lets you choose the character to use for blanking out form fields for users who have been restricted by confidentiality or field permissions. It can be a keyboard character or extended ASCII character. Example: "FormViewerCharacter" value="*" > "FormViewerCharacter" value="█" > (Turn on Num Lock; use Alt plus the ASCII code on the numeric keypad) See Confidentiality and Access to Form Fields in the BCI Admin Guide.

<appSettings> section These are user-configurable and specific to TI	
Setting	**Explanation/Value**
FromAddress	The “From:” address in e-mail sent by TI. Please be sure that this is a properly formatted e-mail address; otherwise, e-mail cannot be sent. See E-Mails in the BCI Admin Guide.
IncludeForesightDocument InformationOnSubmission	For use with external systems. Leave this set to false unless told otherwise by TIBCO Foresight.
IsTop10ErrorTypeSearchOn	If `true`, the Errors page can be filtered by type or severity, and you can now see counts for both errors and warnings.
MaxDocSearchDateRange	Maximum date range when using Search, a value of up to 180 days. If omitted, this defaults to 31 days. TIBCO Foresight does not recommend using settings greater than 31 days since this may greatly reduce performance throughout TI. Example: `<add key="MaxDocSearchDateRange" value="15"/>`
MaxLoginAttempts	Number of failed login tries before a user is locked out. If omitted, this defaults to 3. Example: `<add key="MaxLoginAttempts" value="4" />`
PartnerCacheRefreshTime	If UsePartnerCache is true, this sets the refresh rate for the All Partners list. Time is in milliseconds. (There are 600000 milliseconds in 10 minutes).
PartnerFilterRefreshInterval	Sets the refresh rate for generated values in partner filters. Default is 12 hours. Example: `<add key=" PartnerFilterRefreshInterval " value="6" />` This example causes the generated values in the partner filters to refresh every 6 hours. Please see Setting up Partner Filters in **BCI_Admin*nnn*.pdf** in the TransactionInsight Doc directory.

<appSettings> section These are user-configurable and specific to TI	
Setting	**Explanation/Value**
ShowAllCutoff	This controls whether the Partners, Senders, and Receivers pages will have an “All” entry in the drop-down list to “Show __ records per page” at the bottom left. Default is to omit “All” in these lists. This prevents performance problems if you have a lot of partners. To include an “All” choice in these partner lists, add a ShowAllCutoff option to Web.config. This displays the “All” selection if the list contains under a certain number of partners. Example: `<add key="ShowAllCutoff" value="50" />`
ShowDocumentSummaryDetails	This will display the following document information on tabs at the bottom of the Document Summary page: ▪ Transaction filters values ▪ Extended field values ▪ External identifier values This must be added manually. Default is false. Example: `<add key="ShowDocumentSummaryDetails" value="true" />`
ShowSaveLoginBox	This controls whether the Remember Login box appears on the login page. Default is true. Example: `<add key="ShowSaveLoginBox" value="false"/>`
SmtpServer	The SMTP server that BCI uses to send e-mail. See E-Mails in the BCI Admin Guide.
UsePartnerCache	If `true`, when the webserver starts, a separate thread loads the BCI All Partners list. When a user with a role that has All Partners logs in, BCI uses the partner list that is already created. This speeds up logon and the partner configuration page. Please see also PartnerCacheRefreshTime.
ValidateDebug	Leave this set to `0` unless a TIBCO Foresight technical support person directs you to do otherwise.
WebApplicationUrl	Base URL for BCI e-mail links. Example: `http://SERVER21/TIDemo/` See Messages in the BCI Admin Guide.

<system.web> settings Caution: Use care when changing anything in this section See Microsoft documentation for details http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cpguide/html/cpconconfiguringclientapplication.asp	
timeout	Web.config has two timeout settings, which are Microsoft .NET entries: ▪ One sets the time in minutes before a user is logged out (should be equal to or greater than the session timeout): <forms name=".ASPXAUTH" loginUrl="default.aspx" protection="All" **timeout="60"**/> ▪ The other one sets the time in minutes before a user's session is discontinued: <sessionState mode="InProc" . . . **timeout="20"**/>

Application Settings in Web.config –SQL Server Example

```
<add key="DBType:TransactionInsight" value="SqlServer" />
<add key="DB:TransactionInsight" value="Application Name=
TIWebGUI;server=duelmach;initial catalog=TIDEV;UID=QAadmin2;
PWD=w1e2r3t4y5;"/>
<add key="SmtpServer" value="relay.foresightcorp.com" />
<add key="FromAddress" value="donotreply@foresightcorp.com"
<add key="WebApplicationUrl" value="http://www.Kavercorp.com/TI4.0.16.0
Web/" />
<add key="SubmitDirectory" value="c:\Projects\" />
<add key="InStreamDirectory" value="c:\foresight\instream\bin" />
<add key="CollectorDirectory" value="C:\Projects\TransactionInsight\
Collector\bin" />
<add key="ValidateDebug" value="0"></add>
<add key="997Matching" value="false"></add>
<add key="CustomDefaultPage" value="off"></add>
<add key="FormViewerCharacter" value="" ></add>
<add key="ExclusionSetting" value="off"></add>
<add key="ForceDirectory" value="c:\Inbound\Force\" />
<add key="FormMode837P" value="New"/>
<add key="FormMode837I" value="New"/>
<add key="FormErrorCategoryThreshold" value="Error"/>
<add key="IncludeForesightDocumentInformationOnSubmission" value="true"/>
<add key="UsePartnerCache" value="true"/>
<add key="PartnerCacheRefreshTime" value="600000"/>
```

Application Settings in Web.config –Oracle Example

The application settings are similar to the SQL Server example, except for DBType and DB. Example:

```
<add key="DBType:TransactionInsight" value="Oracle" />
<add key="DB:TransactionInsight" value="Data Source=ora10_M1;
Pooling=true;User ID=QAadmin1;Password=w1e2r3t4y5;Persist Security
Info=true;"/>
```

Connection String example in Web.config – Oracle Instant Client

```
<add key="DB:TransactionInsight" value="Data Source=localhost:1521/xe;
Pooling=true;User ID=QAadmin1;Password=w1e2r3t4y5;Persist Security
Info=true;"/>
```

Where:

`localhost:1521`	system and port
`xe`	Oracle SID
`User ID=Qaadmin1`	Oracle schema name
`Password= w1e2r3t4y5`	Oracle schema password

Encrypting Password Information in Web.config

To encrypt Web.config's appsettings section, which includes the database password:

1. Back up the unencrypted Web.config to another location, since the encrypted file will only work on the current machine.
2. Go to the directory containing Web.config (the environment's TIWeb directory).
3. Execute this at the command line:

   C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727\aspnet_regiis.exe -pef "appSettings"

   The period at the end is part of the command.

   Look for this message:

   ```
   Encrypting configuration section...
   Succeeded!
   ```

   The appsettings section should now be encrypted.
4. Give ASPNET user access:

   ```
   C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727\aspnet_regiis -pa
   "NetFrameworkConfigurationKey" "ASPNET"
   ```

   Look for this message:

   ```
   Adding ACL for access to the RSA Key container...
   Succeeded!
   ```

To unencrypt:

```
C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727\aspnet_regiis.exe
-pdf "appSettings"
```