



**SUNGARD** PUBLIC SECTOR  
7i SERVER ADMINISTRATION

IFAS  
Integrated Financial &  
Administrative Solution

## NOTICE

SUNGARD PUBLIC SECTOR BI-TECH LLC MAKES NO REPRESENTATIONS OR WARRANTIES, ORAL OR WRITTEN, EXPRESS OR IMPLIED, WITH RESPECT TO THE SYSTEM, SERVICES, SOFTWARE, DOCUMENTATION, OPERATING ENVIRONMENT, ANY OTHER SOFTWARE OR SERVICES PROVIDED HEREUNDER OR ANY OTHER MATTER ADDRESSED HEREUNDER, AND SUNGARD PUBLIC SECTOR BI-TECH LLC EXPLICITLY DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, INCLUDING THE IMPLIED WARRANTIES OF TITLE, MERCHANTABILITY AND FITNESS FOR A SPECIFIC PURPOSE. SunGard Public Sector Bi-Tech LLC shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance or use of this material. This documentation is proprietary and confidential information of SunGard Public Sector Bi-Tech LLC. Copying, reproduction or distribution is strictly prohibited. All rights reserved.

Copyright © 2008 by

SunGard Public Sector Bi-Tech LLC

890 Fortress Street

Chico, CA 95973

Should you wish to make a comment, or if you find an error in the text, please contact us via email:

[doc@bi-tech.com](mailto:doc@bi-tech.com)

## Document Change Log

<b>Version</b>	<b>Date</b>	<b>Change Description</b>
7.9	March 2008	7.9 Version

# Contents

<b>1</b>	<b>7i Server Administration .....</b>	<b>9</b>
1.1	Introduction .....	9
1.1.1	Design.....	11
1.1.2	Scalability.....	11
1.1.3	7i Network Architecture.....	12
1.1.4	Presentation Layer.....	13
1.1.5	Business Logic Layer.....	13
1.1.6	IFAS 7i IQ Folder Mapping .....	13
1.1.7	IFAS Server Configuration .....	14
1.1.8	GUI Tool .....	14
1.1.9	Test the Screens.....	15
<b>2</b>	<b>Administering Connections .....</b>	<b>16</b>
2.1	System Catalog.....	16
2.2	Connection Manager.....	16
2.2.1	General Tab.....	18
2.2.2	Database Tab .....	19
2.2.3	Telnet Host Tab .....	20
2.2.4	Workflow Tab.....	22
2.2.5	App User Tab.....	24
2.2.6	Jobs Tab .....	25
2.2.7	User Tab .....	27
2.2.8	WWW Servers Tab .....	28
2.2.9	Login Domains.....	29
2.2.10	LDAP Tab .....	29
2.2.11	Archive Tab.....	32
2.2.12	Report Server Tab .....	34
2.2.13	Special Tab.....	35
2.2.14	Driver Options Tab.....	36

<b>3</b>	<b>Administrative Console Screens.....</b>	<b>37</b>
3.1	Getting Started .....	38
3.1.1	Introduction .....	38
3.1.2	Settings.....	40
3.1.3	Connection Manager .....	43
3.1.4	Start Page.....	49
3.2	First Time Administrative Console Use .....	51
3.2.1	Initial setup.....	51
3.3	Configure Local Server .....	54
3.3.1	Server Config.....	54
3.3.2	Data Processing Service.....	55
3.3.3	Queue Processors .....	57
3.3.4	Applications .....	58
3.3.5	Scheduled Maintenance .....	60
3.3.6	Tracing.....	62
3.4	Job Manager .....	63
3.4.1	What is a Job? .....	64
3.4.2	IFAS Job Management .....	67
3.5	Monitor Servers.....	73
3.5.1	General Status.....	77
3.5.2	CDD Report Queue .....	78
3.5.3	Request Broker .....	84
3.5.4	Status Log.....	85
3.6	Database Logging Worksheet.....	89
3.6.1	Database Logging.....	89
<b>4</b>	<b>Other 7i Administration Tasks.....</b>	<b>98</b>
4.1	Screen Compile.....	98
4.2	Defaults Rules (NUUPDF) .....	99
4.2.1	Entity List .....	101
4.2.2	VBScript Editor .....	103
4.2.3	XML Editor .....	113
4.2.4	BT20 Tree View .....	121

<b>5</b>	<b>Network Level Security .....</b>	<b>122</b>
5.1	IIS Port Selection .....	123
5.2	Secure Sockets Layer (SSL).....	124
5.2.1	Supporting Nucleus Defaults when SSL is used.....	127
5.3	Broker Security Policy for 7i .....	127
5.4	Firewall Considerations .....	129
5.5	Integrated Windows Authentication.....	131
5.5.1	How “Integrated Windows Authentication” works with IFAS 7i/Dashboard .....	131
5.5.2	Getting “Integrated Windows Authentication” enabled in 7i .....	131
<b>6</b>	<b>IFAS Windows 2003 Advanced Server Configuration .....</b>	<b>141</b>
6.1	Overview .....	141
6.2	Hardware Recommendations.....	142
6.3	Win2K Recommendations.....	142
6.3.1	Dual NIC for IFAS 7i .....	152
6.3.2	7i Server #1 .....	153
6.3.3	7i Server #2 .....	156
6.4	Optional.....	159
6.5	Unix Server Load Balancing over multiple Network Cards.....	160
<b>7</b>	<b>Posting Preferences and User Defaults Conversion.....</b>	<b>161</b>
7.1	The map2vb Program .....	161
7.2	Using map2vb .....	161
<b>8</b>	<b>CDD.net Configuration .....</b>	<b>165</b>
<b>9</b>	<b>Remote Configuration .....</b>	<b>166</b>
9.1	Remote Configuration .....	166
9.1.1	General Service Configuration.....	166
9.1.2	Queue Processor Configuration.....	167
9.1.3	CDD Reports .....	168

<b>10 VBScript Editor .....</b>	<b>170</b>
10.1    Menu Options.....	171
<b>11 XML Editor .....</b>	<b>172</b>
11.1    Menu Options.....	172
<b>12 XSL Tester .....</b>	<b>174</b>
12.1    Top Panel.....	175
12.2    Bottom Panel .....	175
<b>13 Troubleshooting.....</b>	<b>176</b>
13.1    Request Logging.....	176
13.2    Proper start/stop of web services.....	177
13.3    Service Packs .....	177
13.4    Database Issues .....	178
13.5    OpenLink.....	178
13.5.1    Version 4.2.....	178
13.5.2    Multi vs Single-Threaded Agents .....	179
13.5.3    ODBC Tracing .....	179
13.5.4    Receive Timeout.....	179
13.6    Do's and Don'ts.....	179
13.6.1    Do's.....	179
13.6.2    Don'ts.....	180
13.7    Web Server Settings for Performance .....	180
<b>14 Customizations .....</b>	<b>182</b>
14.1    Custom Business Rules.....	182
14.2    Server VB Script .....	184
14.3    Adding Reports to Screens.....	189
14.4    Adding Screen Links to Screens.....	190
14.5    Error Catalog.....	191
14.6    Custom Tab Orders in 7i.....	196

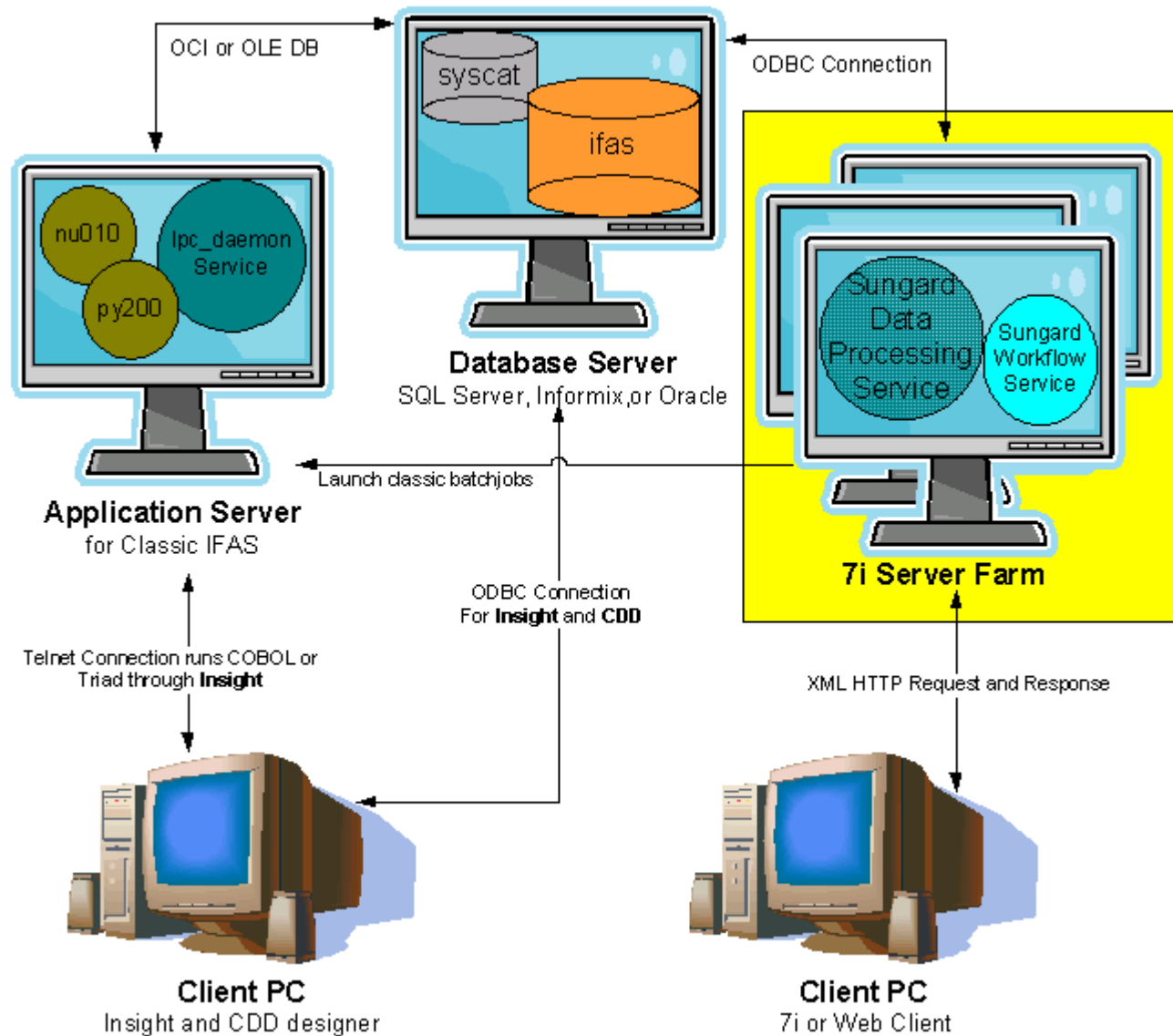
<b>15 Registry Settings .....</b>	<b>201</b>
<b>16 FAQs .....</b>	<b>203</b>



# 1 7i Server Administration

## 1.1 Introduction

This document discusses the administration of the 7i Server, or Network Load Balanced Farm of 7i Servers.



This manual is associated with the 7i Server, or Network Load Balanced (NLB) farm of 7i Servers, highlighted in yellow in the above diagram. For details on Database Server administration, see your Database Vendor documentation. For details on the Application Server (Classic IFAS Server), see the Application Server Administration Guide. For information on initial setup, see the PC Install Guide.

## 1.1.1 Design

The goal for the 7i product was to create a system that has the following characteristics:

Preserve all functionality of existing processes of version 4.3

- Scalable
- Available
- High-throughput
- Run on Web farm
- Thin-client, three-tier architecture

The key to providing the scalability, availability and high-throughput lies with the ability to run the software on a web farm (also known as Network Load Balancing, or NLB) in a true multi-tier manner. You can break the multi-tier design into 3 components:

**Presentation Layer:** The presentation layer is hosted in Internet Explorer 6 and is comprised of DHTML and element behaviors and J Script. “Edge” applications (Employee Online, Applicant Online, etc) use the 7i server architecture but support multiple-browsers.

**Business Logic Layer:** The business logic layer is on a Windows server and provides data related services: data fetches, updates, inserts, deletes, validation, etc. Business rules are all run on this layer with the information that is submitted from the client tier.

**Database Layer:** The Databases supported are Informix or Oracle on a UNIX server, or SQL Server on a Windows server.

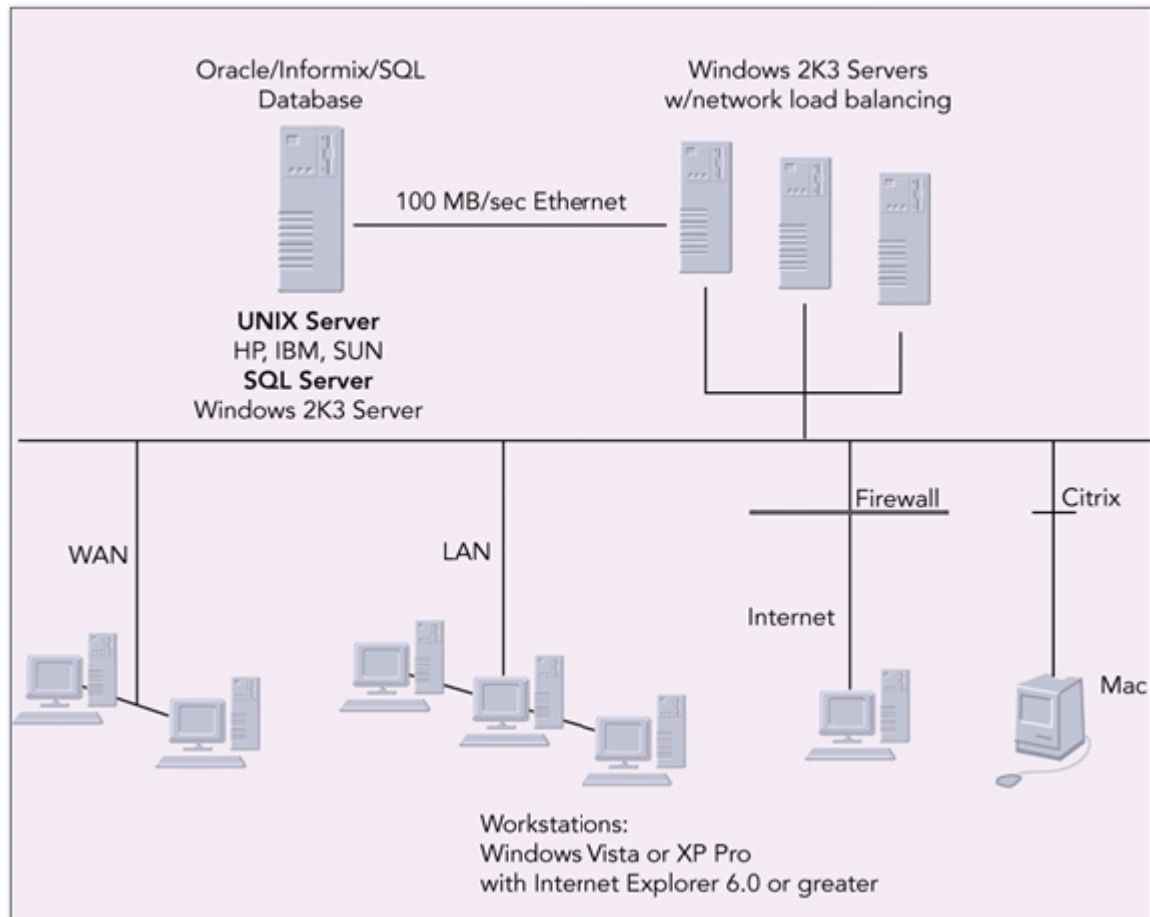
## 1.1.2 Scalability

To build a scalable application, 7i uses an asynchronous processing model. 7i achieves this through the use of Microsoft Message Queue as the communication medium between the ISAPI client end point and the process that performs the work on behalf of the client.

Given that the workload on the server cannot be predicted, it is necessary to queue requests that cannot be satisfied with available resources. Queuing requests and processing them in an asynchronous manner, 7i attempts to keep a relatively static pool of available work contexts. This reduces the amount “thrashing” that would occur if threads were created as needed without limit.

### 1.1.3 7i Network Architecture

The following provides another perspective of the 7i network architecture.



The three tiers of the 7i architecture are shown with the Database and 7i Servers above the horizontal line, and the various options in the client tier below the line.

## 1.1.4 Presentation Layer

The presentation layer is implemented with Active Server Pages (asp) that make extensive use of DHTML and HTML Components and J Script. The client browser communicates with the Windows server farm with XMLHttp requests. These XMLHttp requests are sent to the farm and any server in the farm may service the request. The browser session is tied only to the farm, and not a specific server in the farm, although the network load-balancing component can be configured to preserve server affinity.

## 1.1.5 Business Logic Layer

There are four primary components on each Windows Server on the farm.

1. The 7i software
2. Microsoft Message Queues
3. IIS
4. Network Load Balancing

The 7i Software includes an ISAPI DLL called btwebrqb.dll on each 7i server in the NLB farm. ISAPI DLLs are loaded with Microsoft's Internet Information Services (IIS) manager. Our btwebrqb.dll consumes 7i web requests and quickly writes them to a Microsoft Message Queue. There is a 7i service running on each server that reads requests from these message queues, and dispatches them to our business logic. This business logic services the request and places the response on a response Message Queue. The btwebrqb.dll ISAPI DLL picks up the response from the Message Queue and sends it back to the requesting web page.

## 1.1.6 IFAS 7i IQ Folder Mapping

With IFAS 7i, the traditional terminal-based Interactive Inquiries are being replaced with Click Drag & Drill reports. However, many users will find that they are used to the standard menu location of the IQ Reports (GLIQ, APOHIQ, etc.). To facilitate this need we have added an option to the JCL that allows IFAS menu masks to be mapped to CDD Folders. The syntax of the mapping statement is “[CDDfolder://<folder\_id>]” where “<folder\_id>” is the object ID of the desired folder. Remember that folder ID is not the same as the description typically shown to users. The folder ID can be identified with a simple query of the us\_secobj\_mstr. (ex: select us\_so\_id from us\_secobj\_mstr where us\_so\_desc = “GL Reports”)

If the folders are not present for a specific IFAS database they can be recreated using the Administrative Console application. (See the “Load Base Security Structure” section of the Administrative Console documentation.)

### 1.1.7 IFAS Server Configuration

After installing the software on each computer, the next step is to configure the Data Processing Service. This service is responsible for launching a number of other processes that do the actual message processing. For each connection that is active seven processes will be launched to handle incoming requests from client applications. The seven processors process messages that are dispatched to seven queues:

- High-priority IFAS messages

- Low-priority IFAS messages

- High-priority Nucleus messages

- Low-priority Nucleus messages

- High-priority Daemon messages

- High-priority Tools messages

- Report requests

### 1.1.8 GUI Tool

The PC Installer will create a folder on the 7i web servers desktop named "7i Tools". In this folder are several useful tools for managing the 7i server. Several of these options will allow the compiling of the 7i screens either singly, by subsystem, by connection, or every screen for every connection enabled on the server.

"Compile Single 7i Screen" will allow the compilation of a single screen from a list of all screens available. The user will be prompted for the Subsystem and the screens in that subsystem to compile. If more than one connection is enabled on the 7i server there will be a prompt to select which connection to compile.

"Compile Single 7i SubSystem" will allow the compilation of all screens in a single subsystem for a single connection. The user will be prompted for the Subsystem to compile. If more than one connection is enabled on the 7i server there will be a prompt to select which connection to compile.

Compile 7i Screens for One Connection" will allow the compilation of all screens in all subsystems for one connection. If more than one connection is enabled on the 7i server there will be a prompt to select which connection to compile. If there is only one connection enabled on the 7i server then it is assumed that this is the connection to compile and no prompt will occur.

Compile 7i Screens for All Connections" will allow the compilation of all screens for all connections enabled on a 7i server. Before the compilation begins a restart of the Data Processing Service will occur.

### **1.1.9 Test the Screens**

Now that the screens in the Nucleus subsystem, it's time to do a test run of NUUPCD. Open an instance of Internet Explorer 5.5 and browse to:

<http://localhost/IFAS7/screens/nucleus/nuupcd.asp>

Depending upon the connection definition, you may be prompted for a user name and password.

## 2 Administering Connections

Client based products like CDD and Insight, and the Administrative Console discussed in the next chapter make use of the Connection Manager interface to connect to the system. The Connection Manager is used to configure the connection, as well as to select what connection to log into when launching client based products like CDD, Insight, and the Admin Console. Most of the settings in the Connection Manager are stored in a set of tables called the System Catalog, or syscat.

### 2.1 System Catalog

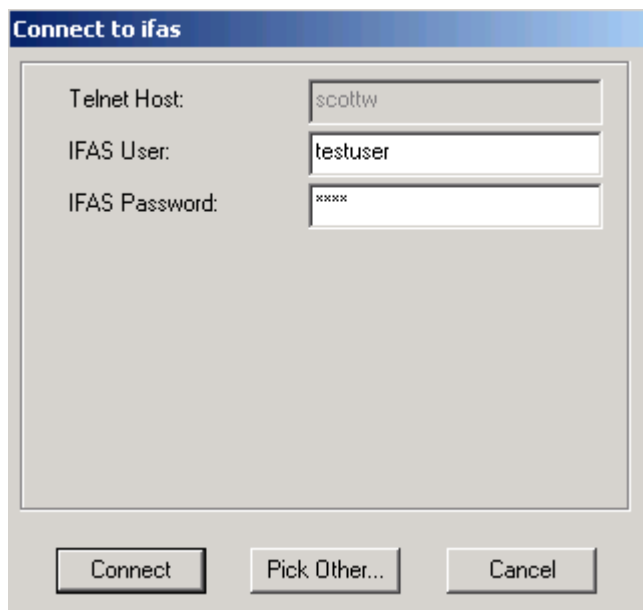
The System Catalog is a set of database tables that can be inside of an ifas database alongside the financial tables, or in a database of its own. It has 2 primary functions: 1) it is the primary storage mechanism for the Insight and CDD PC Products, and 2) it stores the connection information of IFAS databases accessible from within the PC Products and 7i. Since the System Catalog can house connection information for an infinite number of IFAS databases, you should have only one syscat database defined for any given entity to simplify connection information and reduce administrative requirements. In order to use the PC products and the Administrative Console on a given client PC, an ODBC connection to the system catalog is required on that PC.

### 2.2 Connection Manager

The Connection Manager launches automatically as part of the login process for CDD, and the Administrative Console. From it, you may select from one of the existing connections. Only if you are launching the Admin Console can you change connection information or add a new connection.



When IFAS PC Products are initially installed, much of this information is populated automatically for the initial connection. Therefore, you only need to modify it to make adjustments, or add new connections for test, etc. Once a connection is defined, it is stored in the syscat for all users. A connection does not need to be created on each client PC that will use these applications. Upon launching, the following login dialog appears:



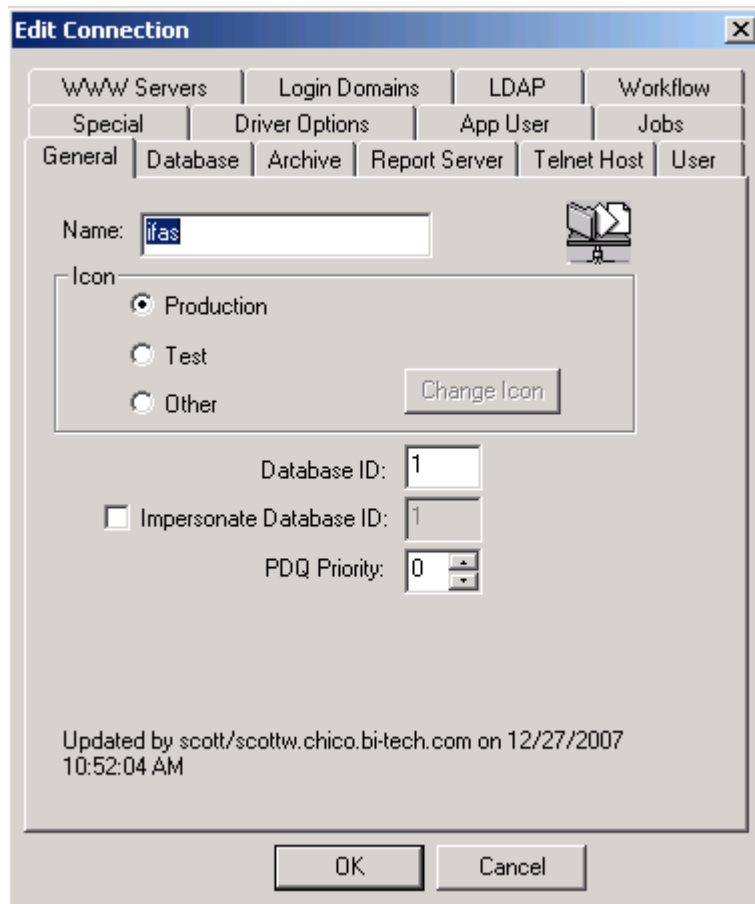
The Connection Manager is entered by selecting “Pick Other...”.



Right-click an existing connection and select “Edit Properties” option to display the Edit Connection page. Following is discussion on each of the tabs on this page, in order of importance.

## 2.2.1 General Tab

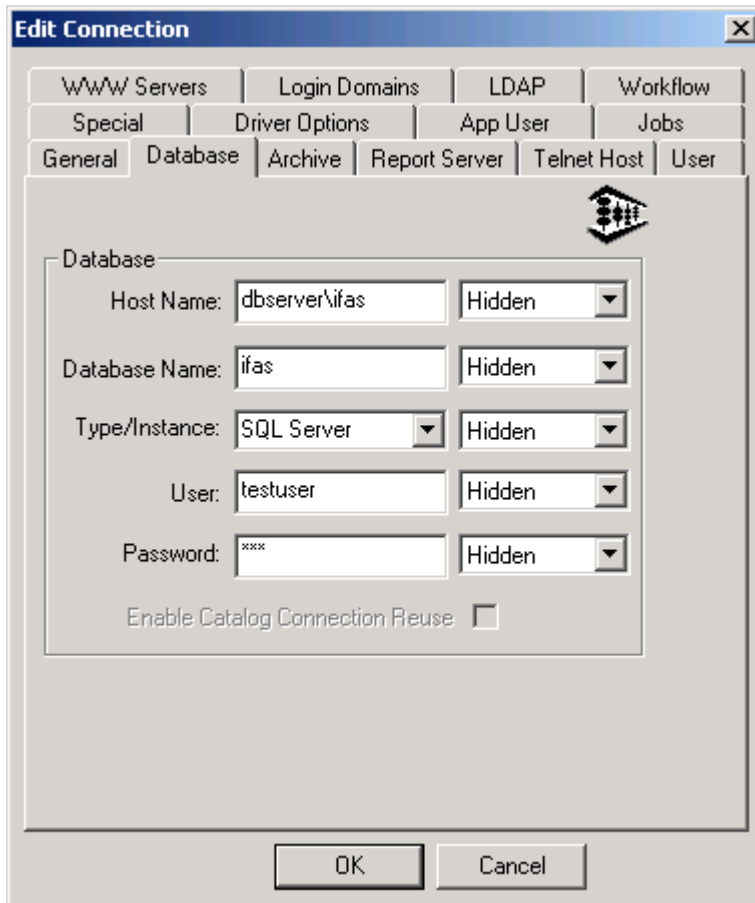
The "General" tab enables you to select the name and icon for your connection or make changes. Note that you can select the connection icon as "Production," "Test," or "Other" depending upon what icon you wish to associate with the connection.



The Database ID is automatically populated, and represents a unique identifier for the database connection being defined. The Impersonate and PDQ Priority are rarely changed, and are for special circumstances.

## 2.2.2 Database Tab

This tab is where information specific to the database connection is defined. The Connection Manager assembles an ODBC connect string with the information entered here, and stores it in the syscat to be shared across the enterprise.



The screenshot shows the 'Edit Connection' dialog box with the 'Database' tab selected. The dialog has a title bar with 'Edit Connection' and a close button. Below the title bar are several tabs: 'WWW Servers', 'Login Domains', 'LDAP', 'Workflow', 'Special', 'Driver Options', 'App User', 'Jobs', 'General', 'Database', 'Archive', 'Report Server', 'Telnet Host', and 'User'. The 'Database' tab is active, showing a 'Database' section with a tree icon. The fields in this section are: 'Host Name' (text box with 'dbserver\ifas'), 'Database Name' (text box with 'ifas'), 'Type/Instance' (dropdown menu with 'SQL Server'), 'User' (text box with 'testuser'), and 'Password' (text box with '\*\*\*\*'). Each of these fields has a 'Hidden' dropdown menu to its right. At the bottom of the 'Database' section is a checkbox labeled 'Enable Catalog Connection Reuse' which is currently unchecked. At the bottom of the dialog are 'OK' and 'Cancel' buttons.

**Host Name:** The name or IP Address of the Database Server to connect with.

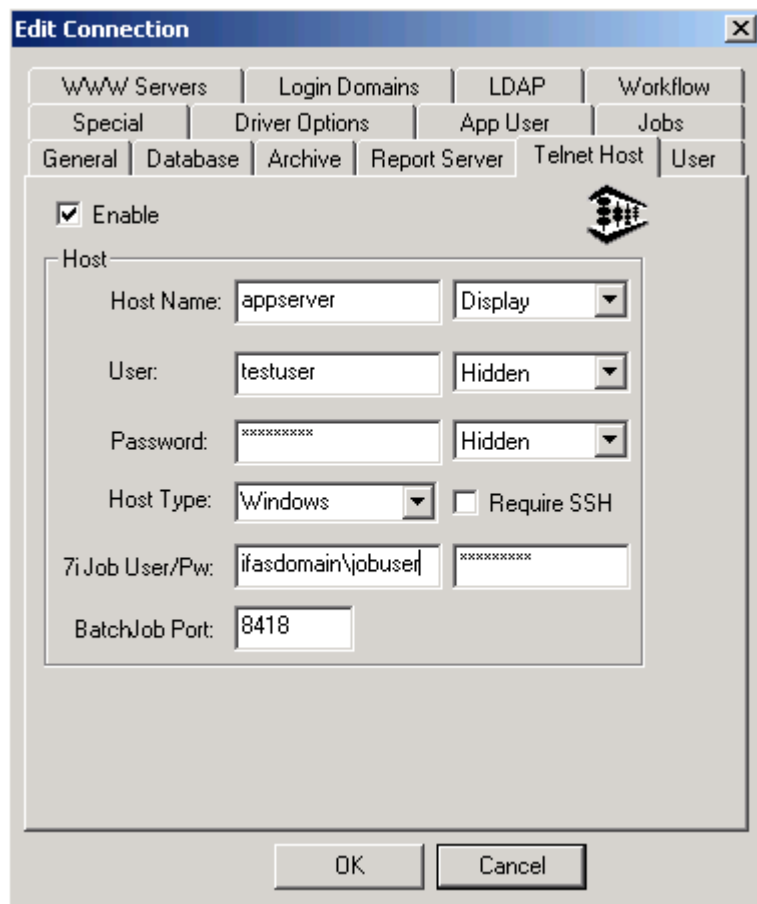
**Database Name:** The name of the database you are connecting to.

**Type/Instance:** Controls the type of database you are connecting to: Oracle, Informix, or SQL Server. If using the OpenLink ODBC driver to connect with Informix or Oracle, this must match the name of your Domain Alias from the oplrqb.ini file. This field is case sensitive.

**User:** This is a user defined at the database level if you are not using a Trusted for of authentication.

## 2.2.3 Telnet Host Tab

This tab is where the properties for the application server are set. This may or may not be the same information as used on the Database tab.



The screenshot shows the 'Edit Connection' dialog box with the 'Telnet Host' tab selected. The dialog has a title bar with a close button. Below the title bar are several tabs: WWW Servers, Login Domains, LDAP, Workflow, Special, Driver Options, App User, Jobs, General, Database, Archive, Report Server, Telnet Host (selected), and User. The 'Telnet Host' tab contains the following fields and controls:

- Enable
- Host section:
  - Host Name: appserver (text field), Display (dropdown menu)
  - User: testuser (text field), Hidden (dropdown menu)
  - Password: \*\*\*\*\* (text field), Hidden (dropdown menu)
  - Host Type: Windows (dropdown menu),  Require SSH
  - 7i Job User/Pw: ifasdomain\jobuser (text field), \*\*\*\*\* (text field)
  - BatchJob Port: 8418 (text field)

At the bottom of the dialog are 'OK' and 'Cancel' buttons.

**Host Name:** Name or IP Address of application server.

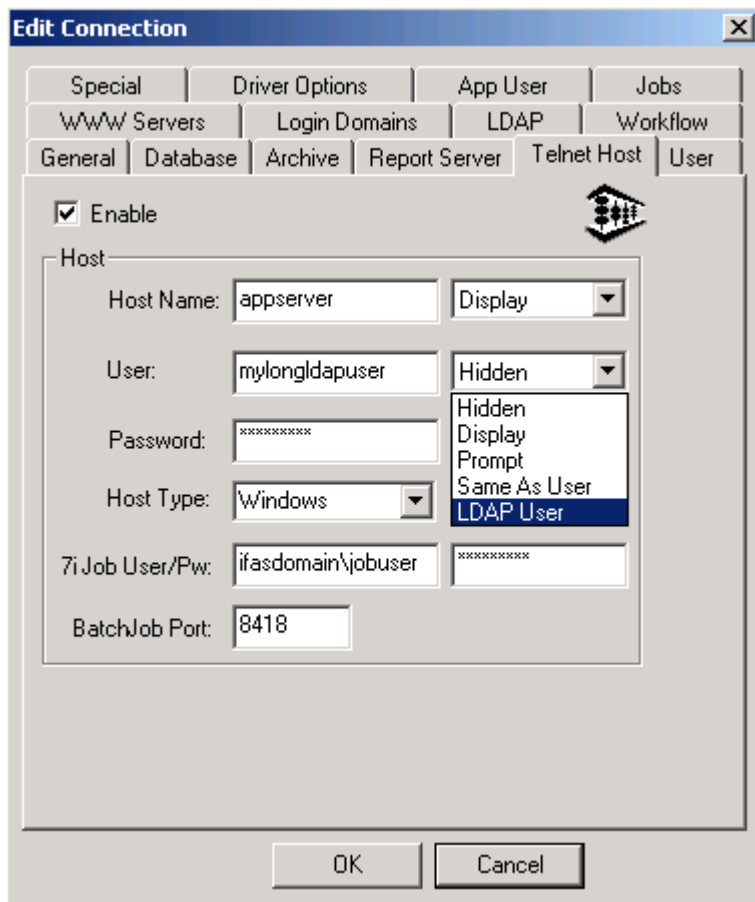
**User:** A user defined at the OS level.

**Host Type:** If you use the SQL Server database, Host Type is Windows. If Informix or Oracle, it will be Unix.

**7i Job User/Pw:** Name and password of a user defined to run jobs. Pertains to SQL Server environments only, since Unix Application Servers use the bt\_jobs script to run jobs.

**Batchjob Port:** The port number (defined in /etc/services or the equivalent Windows file) defined for job running.

NOTE: The User field can also be a long LDAP user name, as shown below.



The screenshot shows the 'Edit Connection' dialog box with the 'User' field dropdown menu open. The 'User' field is set to 'mylongldapuser'. The dropdown menu is open, showing options: Hidden, Display, Prompt, Same As User, and LDAP User. The 'LDAP User' option is highlighted. The 'Host Type' is set to 'Windows'. The '7i Job User/Pw' is set to 'ifasdomain\jobuser'. The 'BatchJob Port' is set to '8418'. The 'Enable' checkbox is checked. The 'Host Name' is 'appserver' and the 'Host' dropdown is set to 'Display'. The 'Password' field is masked with asterisks. The 'OK' and 'Cancel' buttons are at the bottom.

Special	Driver Options	App User	Jobs
WWW Servers	Login Domains	LDAP	Workflow
General	Database	Archive	Report Server
Telnet Host	User		

Enable

Host

Host Name: appserver Display

User: mylongldapuser Hidden

Password: \*\*\*\*\*

Host Type: Windows

7i Job User/Pw: ifasdomain\jobuser \*\*\*\*\*

BatchJob Port: 8418

OK Cancel

## 2.2.4 Workflow Tab

This tab is used for setting up the information needed for Workflow's email functionality.

The screenshot shows the 'Edit Connection' dialog box with the 'Workflow' tab selected. The dialog has a title bar with a close button. Below the title bar are several tabs: 'Special', 'Driver Options', 'App User', 'Jobs', 'General', 'Database', 'Archive', 'Report Server', 'Telnet Host', 'User', 'WWW Servers', 'Login Domains', 'LDAP', and 'Workflow'. The 'Workflow' tab is active and contains the following fields:

- Outgoing E-Mail:** SMTP Server: EXCH2K3
- Incoming E-Mail:**  IMAP4 Server: [ ] IMAP4 Folder: [ ]  POP3 Server: [ ]
- E-Mail Account:** Workflow User: mydomain\testuser Password: [ ] Workflow E-Mail: test.user@myorg.com
- Error Handling:** Send Errors To: test.user@myorg.com On:  Error  No Transition

At the bottom of the dialog are 'OK' and 'Cancel' buttons.

**SMTP Server:** This defines the E-Mail Server for outgoing Workflow Messages. It is assumed the E-Mail Server utilizes an SMTP protocol for outgoing E-Mail messages.

**POP3 Server:** If there is also an incoming E-Mail Server and it utilizes a POP3 protocol instead, this field defines the E-Mail Server. If the E-Mail Server is the same as the SMTP server, leave this field blank.

**Workflow User:** This field defined the Workflow E-Mail account. This is the account that E-Mail Actions will use as the sender of the E-Mail messages. E-Mail Responses will be sent to the Workflow account and picked up by the WF Engine when processing Email responses.

**Password:** This field defines the password for the Workflow User E-Mail account. This password is required by the WF Engine when opening the Workflow User E-Mail account for processing responses.

**Workflow E-Mail Address:** This field defines the fully qualified Workflow User E-Mail address.

**Send Errors To:** If this field is defined, then WF Model Instance records that are processed with an Error or No-Transition state result in the defined E-Mail Address receiving a notification from Workflow automatically regardless of what the Error or No-Transition Activities Action is. The following checkboxes control the type of activity that receives the automatic notification.

**On Error:** Checking this checkbox results in the automatic notification when an Error condition occurs.

**On No-Transition:** Checking this checkbox results in the automatic notification when a No-Transition condition occurs.

## 2.2.5 App User Tab

The App User tab allows you to define an IFAS User and Password to be used for a specific application. The user names used here should be previously defined in NUUPUS. The current settings supported are for Workflow, Employee Online, and Timecard Online.

The screenshot shows the 'Edit Connection' dialog box with the 'App User' tab selected. The 'Application Users' section contains the following fields:

Application	Login	Password
Workflow	<input type="text"/>	<input type="text"/>
Emp Online	<input type="text"/>	<input type="text"/>
Time Online	<input type="text"/>	<input type="text"/>
7i Portal	<input type="text"/>	<input type="text"/>

**Workflow:** If Workflow is to be used with this system connection, it is necessary to fill in the Workflow Login and Password fields on the App User Tab. The login/password will be used by the WF Engine when interacting with the IFAS system. This ID will need to be defined in NUUPUS with blanket insert/update/delete authority regarding tables. This is because the WF Engine needs to be able to process whatever activities are defined in the WF Models with regards to the IFAS system.



**Emp Online:** If you are using Employee Online, check with your implementer to see whether use of the EO App User is appropriate. An alternate configuration is to define a separate connection for EO. Using the same connection saves Web server resources but ties EO more tightly to the 7i screens. If it is decided to use the EO App User, set the User value to EOADMIN with the matching password. Setting this App User will allow EO to use this connection without having to prompt a user for their IFAS name and password. If the single connection configuration is in use then the Emp Online entry must be set to user EOADMIN. No other user name may be used at this time.

**Time Online:** Currently, the setting for Time Online is for future use and can be left blank.

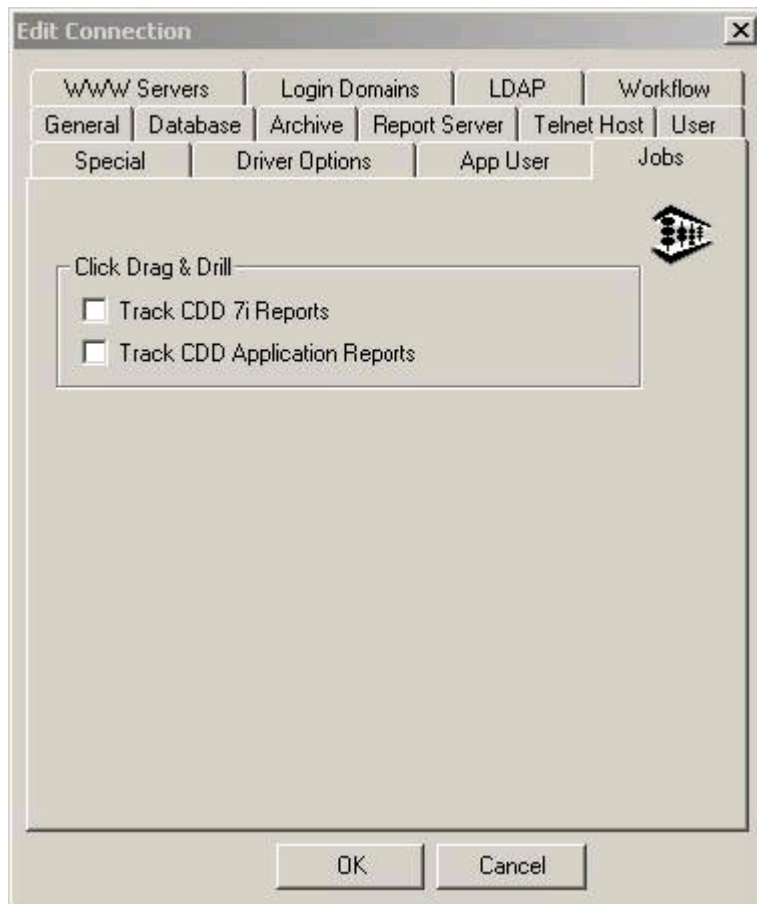
**7i Portal:** This can be set to the user and password for Portal Applications.

## 2.2.6 Jobs Tab

This tab is used to specify how, or if, CDD Reports will be tracked through our Job Manager.

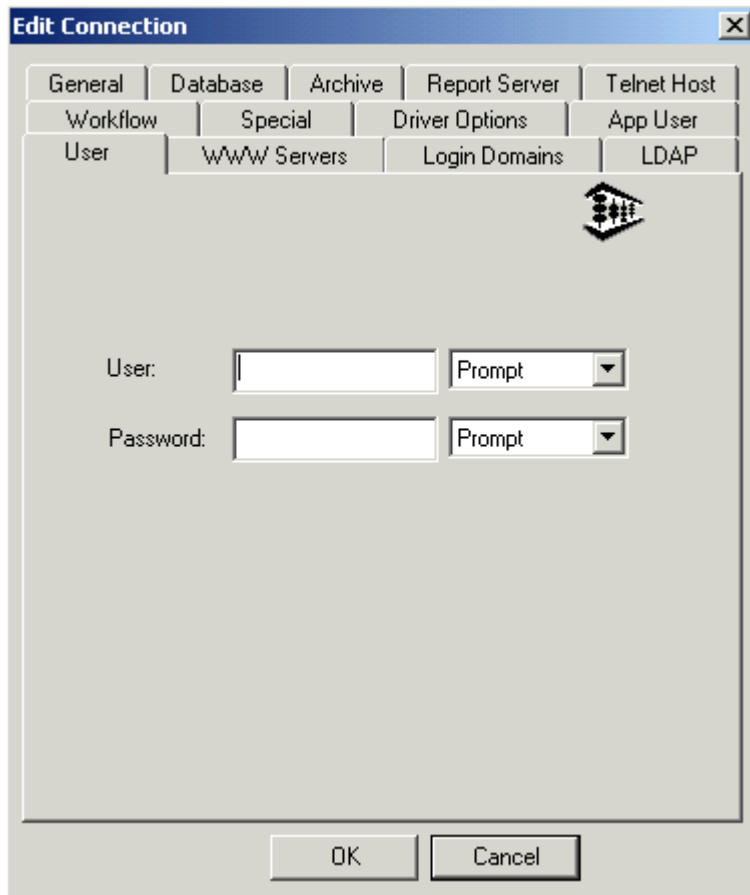
The first option is specific to run through the web (7i Reports). If checked, these 7i reports will be tracked through our Job Manager and their progress and output is accessible from the Job Status page.

The second option is the same concept, but specific to CDD reports run in the Client Server environment. These would be reports run by report designers or others who have CDD installed directly on their workstation.



## 2.2.7 User Tab

This tab is where the IFAS user is entered. The user/password can be hardcoded if desired by using the “Hide” option

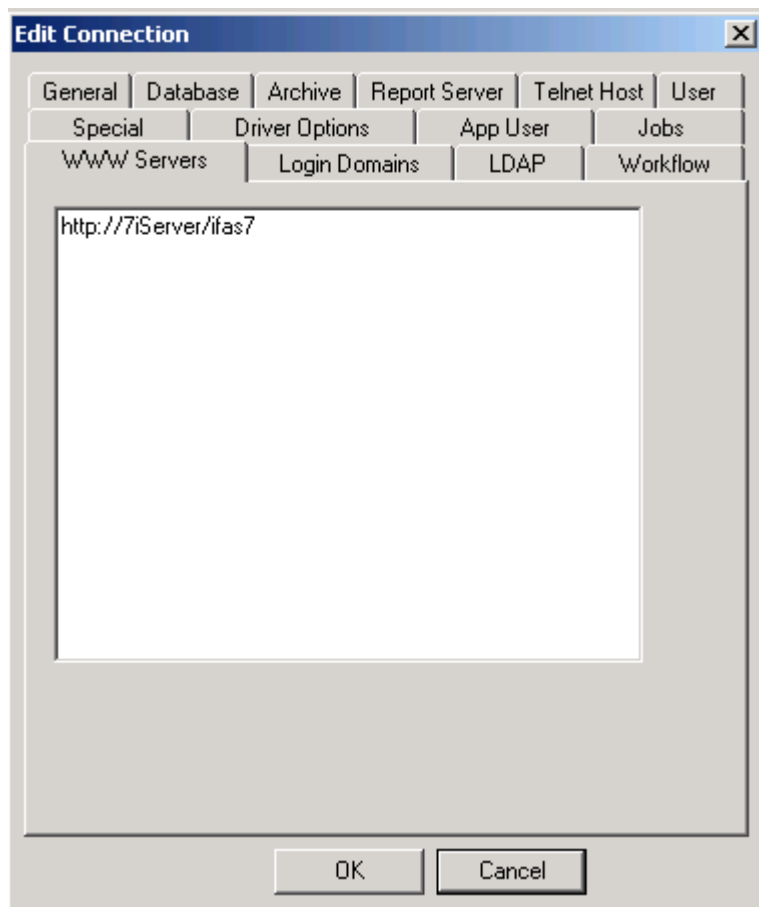


The screenshot shows the 'Edit Connection' dialog box with the 'User' tab selected. The dialog has a title bar with a close button. Below the title bar are several tabs: General, Database, Archive, Report Server, Telnet Host, Workflow, Special, Driver Options, App User, User (selected), WWW Servers, Login Domains, and LDAP. The 'User' tab contains a 'User:' label, an empty text input field, and a dropdown menu with 'Prompt' selected. Below that is a 'Password:' label, an empty text input field, and a dropdown menu with 'Prompt' selected. At the bottom of the dialog are 'OK' and 'Cancel' buttons.

This setting is usually left as seen above, so that users are prompted for a user and password when connecting.

## 2.2.8 WWW Servers Tab

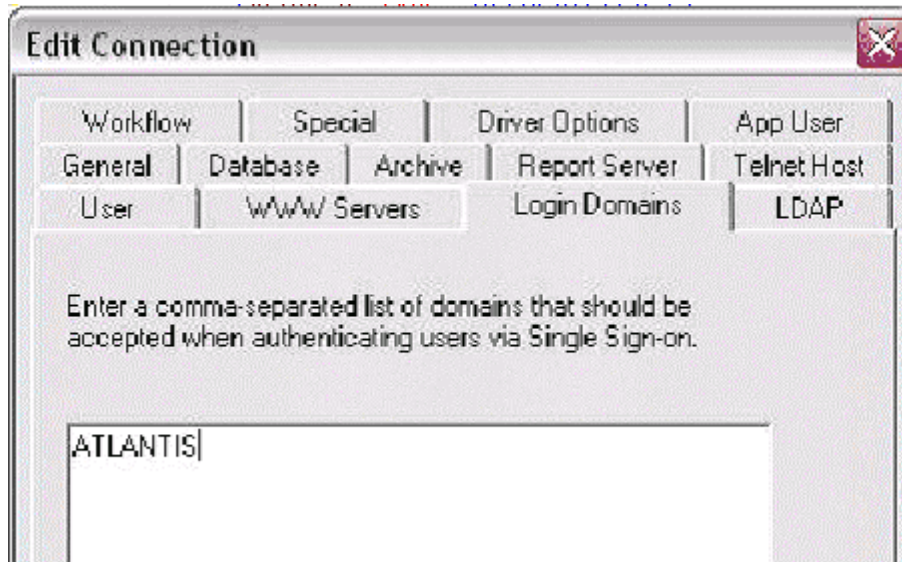
The URL of the 7i server is entered on this tab.



Verify that the URL is correct, and in the correct format: `http://<yourservename>/virtualroot name` (Where virtual root is the name entered during install, commonly IFAS7). Do not use "local host" as the server name. If SSL is in use, https will be used instead of http. See the Chapter entitled Network Level Security for more information on SSL.

## 2.2.9 Login Domains

In the IFAS Connection Manager, the domain name needs to set on the connection's 'Login Domains' tab.



### 2.2.10 LDAP Tab

The PC products support authentication from an external LDAP Server (iPlanet, Active Directory, etc). Once configured, the login process will check the LDAP server first and authenticate with the provided password. If successful, authorization information (functional and database security) will be loaded from IFAS based on the LDAP user provided. If unsuccessful, no specific LDAP error message will be provided and the next authentication step is Nucleus.

The necessary LDAP API routines required are standard only with Windows2000 or greater. LDAP authentication is not supported with earlier Windows platforms.

The screenshot shows the 'Edit Connection' dialog box with the 'LDAP' tab selected. The 'Enable' checkbox is checked. The 'Server' field contains 'your-ldap-server', the 'Port' is '389', and the 'SSL' checkbox is unchecked. The 'User' field is 'ldap' and the 'Password' is masked with asterisks. The 'DN' dropdown shows 'CN=Users,DC=chico,DC=bi-tech,DC=com'. The 'Locate Attribute' dropdown is 'sAMAccountName', the 'Bind Attribute' is 'CN', and the 'IFAS Attribute' is 'SBIUser'. There are 'Test...', 'OK', and 'Cancel' buttons at the bottom.

**Enable:** Check this box to enable the LDAP processing described above.

**Server:** Enter the name of the LDAP server; this name must be resolvable by all PC clients and 7i servers using this connection.

**Port:** The standard LDAP ports are 389 (non-encrypted) and 636 (with SSL). Specify the appropriate port.

**SSL:** Check this box to request the authentication logic to first try SSL and then the standard protocol. If checked, the Port above is the SSL port and port 389 is used for the fallback check.

**User ID:** Enter a LDAP ID that has search permissions on the directory.

**Password:** Enter the Password for the User above.

**DN:** Enter the Base Distinguished Name used for authentication for the User ID above. If the drop-down list is requested, the LDAP server will be queried for published naming contexts. They may or may not be appropriate for the ID. Some example Base DN's are shown here:

Using Microsoft Active Directory: cn=Users,DC=chico,DC=bi-tech,DC=com

Using Sun's iPlanet: "ou=people,dc=Bi-Tech,dc=PublicSector,dc=SunGard,dc=Com"

**Locate Attribute:** Used in the LDAP Authentication process, the Locate Attribute should be a unique value for each Person in the Directory (e.g., sAMAccountName for Active Directory or uid for Sun's Directory Server).

**Bind Attribute:** Authenticate the user against the server by building the full Distinguished Name of the user based on Bind Attribute value.

**IFAS Attribute:** If the User's ID in IFAS is different from the value identified by the 'Locate Attribute', use the IFAS Attribute to identify the proper value to use. For example, if the ID entered is 'Dwight.David.Eisenhower' but his ID in IFAS is simply 'DDE', then a new element can be added to the schema to track this value. The name of the attribute is identified in the 'IFAS Attribute' field.

## Sample of ActiveDirectory configuration for LDAP Authentication

User, Password, Locate Attribute, Bind Attribute, and IFAS Attribute are used as follows during the authentication process:

**Step 1:** Bind to the LDAP server using the defined User, Password, and DN identified on this page. This user only needs read-only access to the Server the ability to search all on the following attributes listed on this page, plus CN. During the "bind", the user string will be in this format: bindattr=USER ID, DN. So, if the following settings are given:

User ID: LDAPUser

BindAttr: CN

DN: cn=Users,DC=chico,DC=bi-tech,DC=com

then the bind string will be: CN=LDAPUser,cn=Users,DC=chico,DC=bi-tech,DC=com

**Step 2:** Based on the Locate Attribute, search for a match with the user-entered ID. The Locate Attribute should be a unique value for each Person in the Directory; e.g., sAMAccountName for Active Directory or uid for Sun's Directory Server.

**Step 3:** Authenticate the user against the server by building the full Distinguished Name of the user based on Bind Attribute value and distinguished name from the record found in Step 2.

**Step 4:** If the User's ID in IFAS is different from the value identified by the 'Locate Attribute', use the IFAS Attribute to identify the proper value to use. For example, if the ID entered is 'Dwight.David.Eisenhower' but his ID in IFAS is simply 'DDE', then a new element can be added to the schema to track this value. The name of the attribute is identified in the 'IFAS Attribute' field. For Active Directory schema changes, our registered Company OID is 1.2.840.113556.1.8000.1757, and in particular, the SBIUser attribute is 1.2.840.113556.1.8000.1757.1.

Note: For LDAP-based users, Password expiration and changing the password are not available from IFAS.

**Troubleshooting:** If, when using the Test button/dialog, you receive a message about "Invalid Credentials [ADMIN]", this means that the user/password/DN combination provided by earlier is not valid. Usually, it is the DN portion that is incorrect. For those familiar with 7i Tracing, the LDAP logic is in BT30.

## 2.2.11 Archive Tab

This tab is used for Sungard's Imaging product. If Document's Online is loaded, this tab is not used. Properties about additional database information specific to the classic imaging database product are entered here. You may use this tab to establish an additional DB for imaging if necessary, or use the current DB as shown in the example above. For new installs, this is usually left disabled.



**Edit Connection**

WWW Servers | Login Domains | LDAP | Workflow  
Special | Driver Options | App User | Jobs  
General | Database | Archive | Report Server | Telnet Host | User

Enable

Database

Host Name:  Same As DB

Database Name:  Same As DB

Type/Instance:  Same As DB

User:  Same As DB

Password:  Same As DB

OK Cancel

**Host Name:** Name or IP Address of server that hosts the Archiving database.

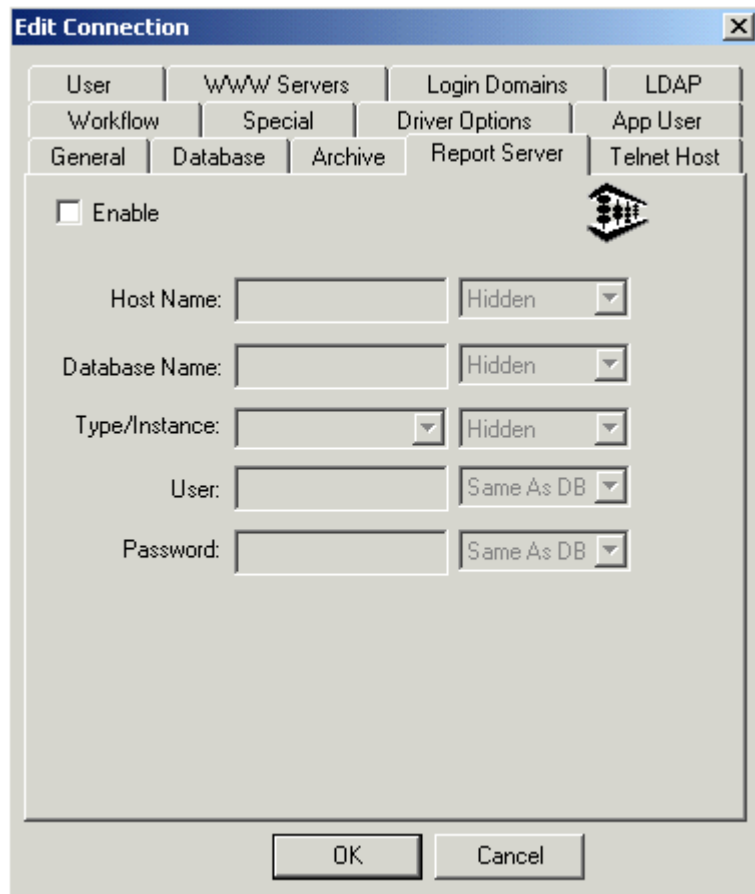
**Database Name:** Name of the archiving database. If set to Same as DB, will use the name of the IFAS db.

**Type/Instance:** Must match the name of your Domain Alias from the oplrqb.ini file if using Openlink. This field is case sensitive

**User/Password:** User defined at the database level.

## 2.2.12 Report Server Tab

This tab is not typically used.



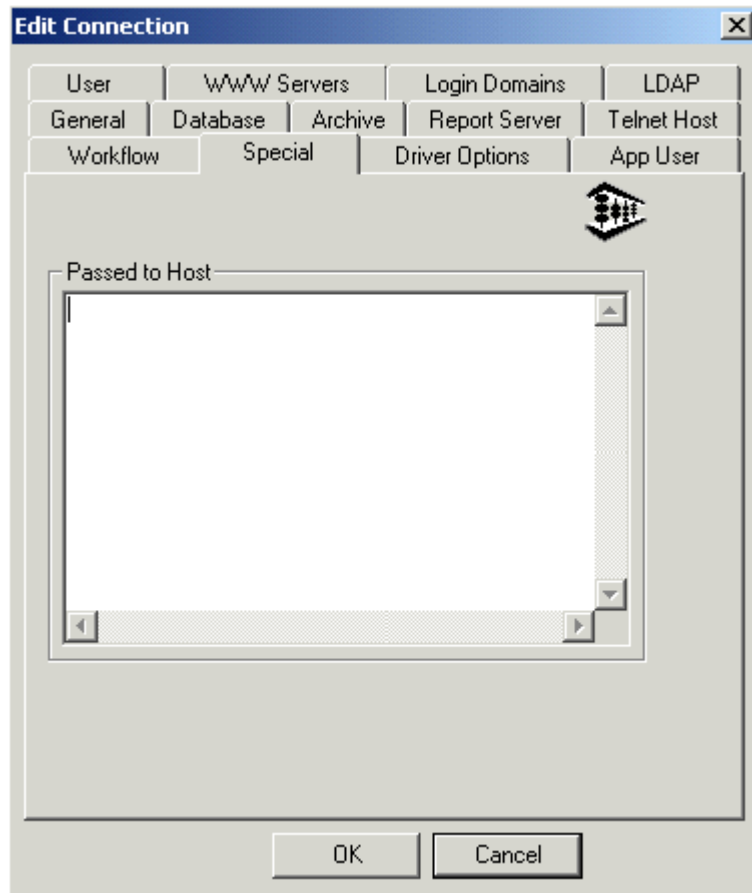
The screenshot shows the 'Edit Connection' dialog box with the 'Report Server' tab selected. The dialog has a title bar with 'Edit Connection' and a close button. Below the title bar are several tabs: 'User', 'WWW Servers', 'Login Domains', 'LDAP', 'Workflow', 'Special', 'Driver Options', 'App User', 'General', 'Database', 'Archive', 'Report Server', and 'Telnet Host'. The 'Report Server' tab is active. It contains an 'Enable' checkbox (unchecked) and a small icon of a server rack. Below this are five rows of input fields, each with a text box and a dropdown menu:

- Host Name: [ ] Hidden
- Database Name: [ ] Hidden
- Type/Instance: [ ] Hidden
- User: [ ] Same As DB
- Password: [ ] Same As DB

At the bottom of the dialog are 'OK' and 'Cancel' buttons.

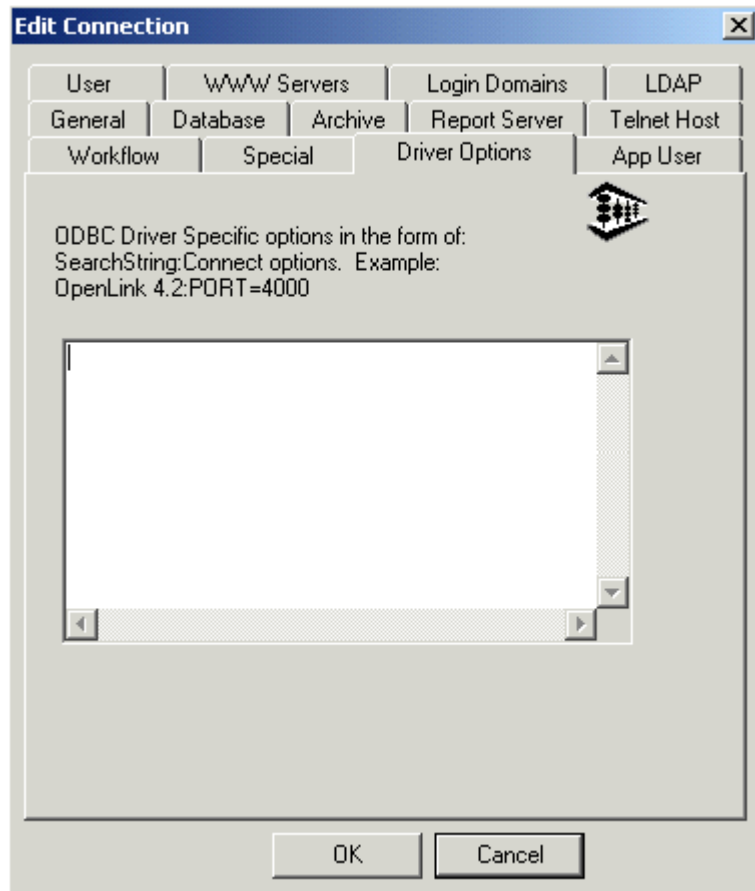
## 2.2.13 Special Tab

This tab allows for special UNIX commands to be sent to the UNIX host, and is not typically used.



## 2.2.14 Driver Options Tab

This tab allows for special commands to be sent to the request broker, and is not typically used.



## 3 Administrative Console Screens

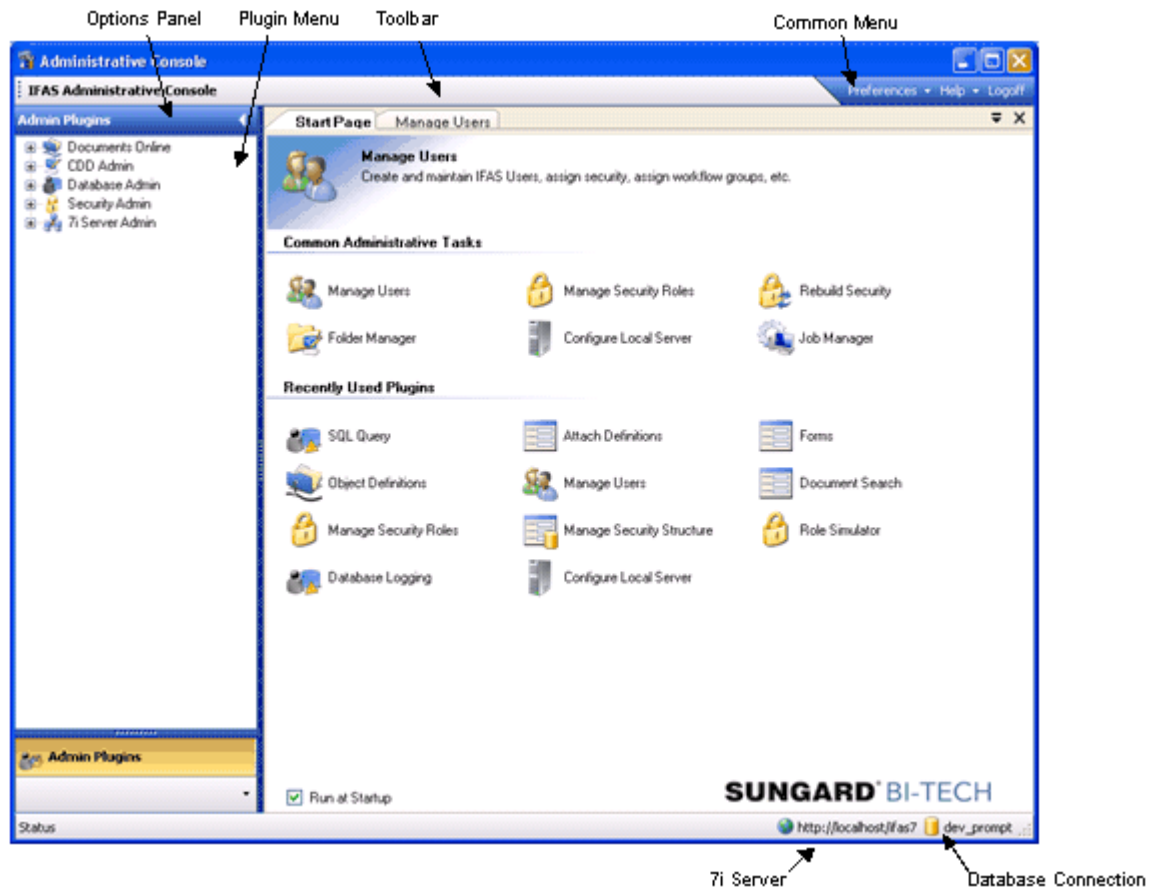
Much of the 7i Server Administration is done using the various screens of the Administrative Console. The Admin Console is also used to administer Application Security, Documents Online, and a host of other Administrative tasks. Screens associated with these other areas are documented separately in the Security, Documents Online, and other guides.

This chapter begins with an overview of the Admin Console, then discusses the screens relevant to administering the 7i servers: Configure Local Server, Job Manager, and Monitor Servers. Discussion of the Database Logging Worksheet is also included here.

## 3.1 Getting Started

### 3.1.1 Introduction

The purpose of the Administrative Console is to centralize the common setup and administrative functions into one application. It is assumed that this tool will only be made to a small subset of users at any given client site as it provides access to some very sensitive configuration options and system access. While the software uses a local Connection Manager connection by default for its data access some of the Plugins it loads also require a connection to a functional 7i server.



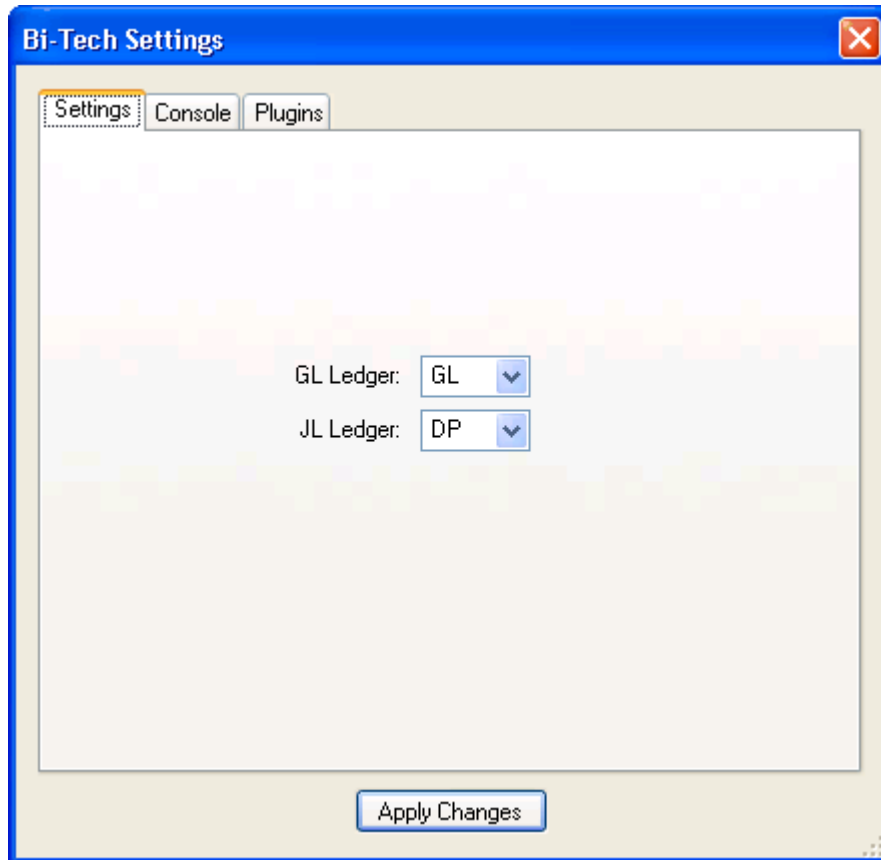
The different administrative functions are broken into different sub-applications named Plugins. These Plugins are treated as separate units of functionality by the application. The menu on the left side of the application provides a basic menu system for organizing the Plugins. By opening the menu options and double-clicking on the desired Plugin it will be initialized and load in the panel on the right of the application in a separate tab. You can open many different Plugins at the same time but be aware that each one opened will increase the amount of memory being actively used by the application. The amount may differ dramatically between Plugins and for some could be quite substantial.

Once you are finished using a Plugin, the “X” button on the tab strip at the far right will allow you to close that individual Plugin. To shutdown the entire application simply click the “Logoff” menu option at the top right of the menu bar.

The status bar at the bottom of the app tracks the current connections. By default the “Database Connection” will show the IFAS connection selected by the user when first connecting. If a plugin is loaded that uses a connection to 7i then a “7i Server” connection will also be initiated and will show in the status bar to the left of the database connection.

### 3.1.2 Settings

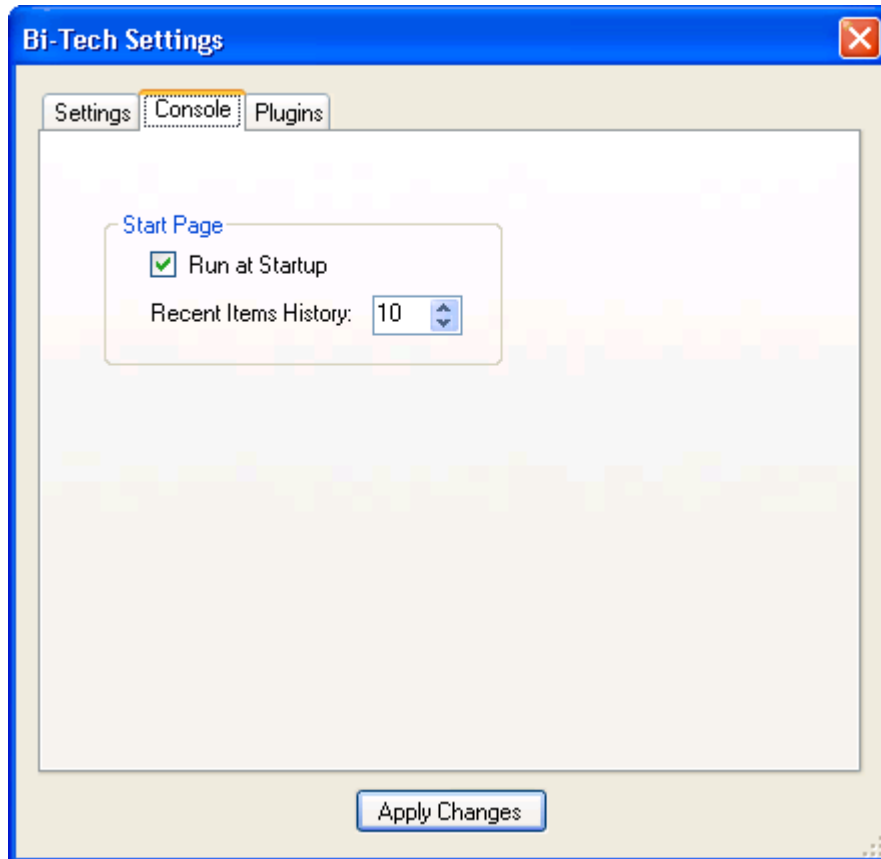
The settings dialog provides access to both the settings specific to the currently active Plugin as well as the general settings for the Administrative Console.





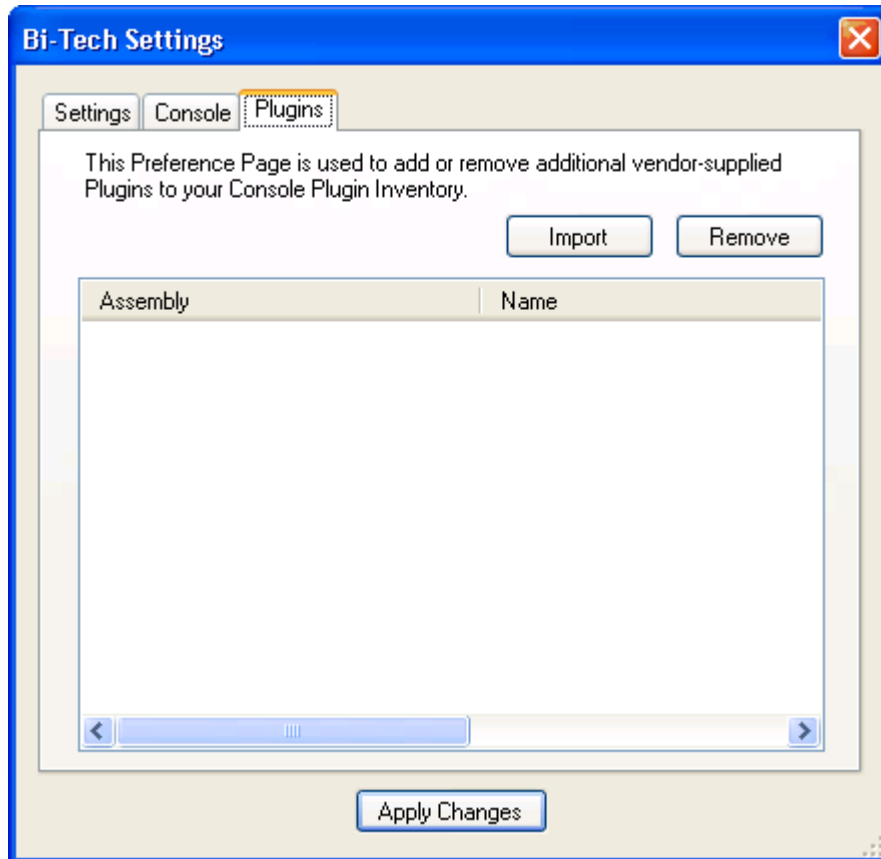
## Settings Tab

This tab provides access to the applications “current” GL and JL ledger. Changing this default will change the default used by any Plugin or report that references the “Default” ledger.



## Console Tab

This tab controls if the Startup Page will be shown and how much history to maintain. The Startup page is the initial Plugin loaded into the Administrative Console and just provides a shortcut to commonly used Plugins.



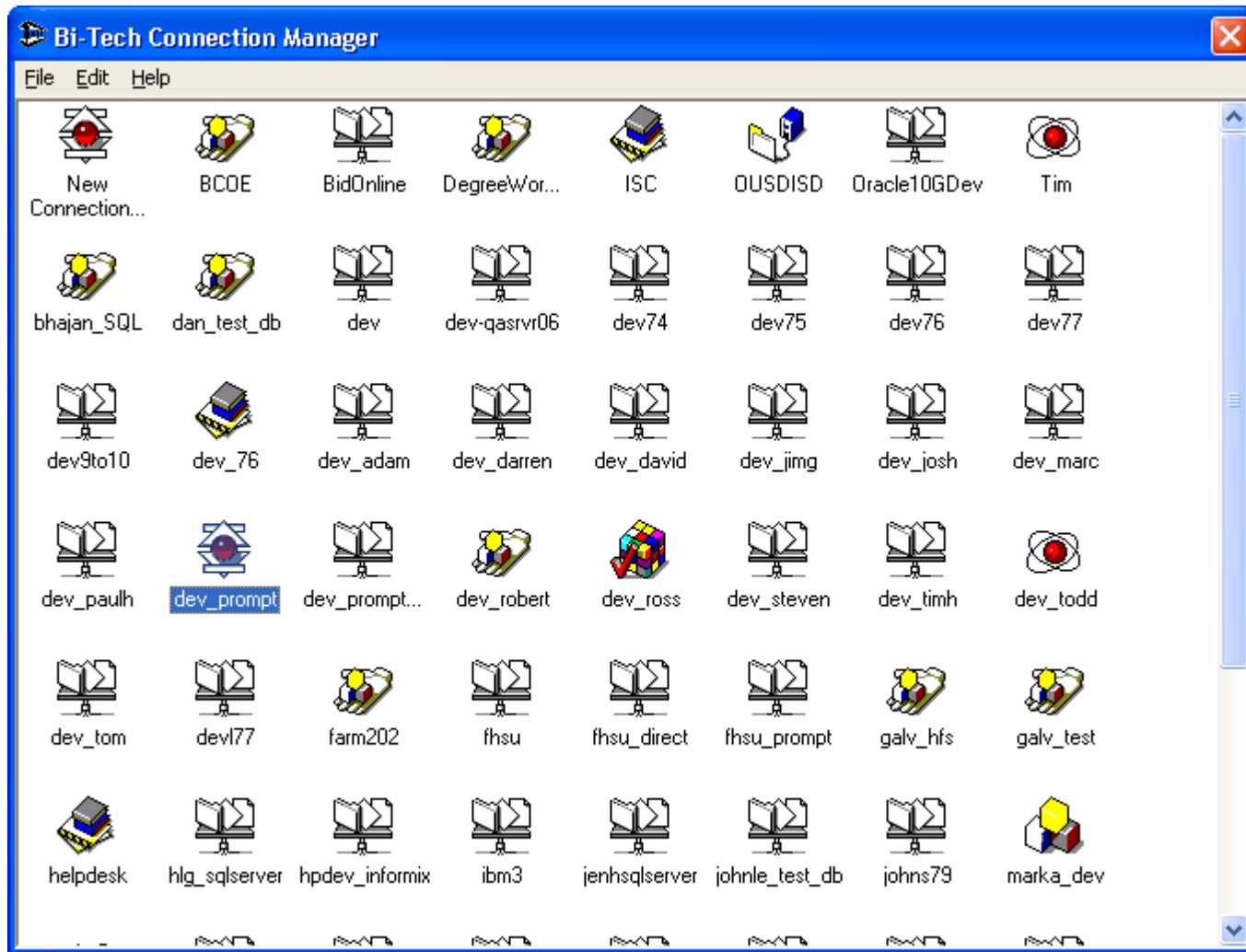
## Plugins Tab

The Administrative Console also allows SunGard Public Sector Bi-Tech to write custom Plugins and deliver them outside of the standard software release. In that situation those custom Plugins can be added to the User's current Plugin menu by using this tab. If you have a custom Plugin delivered the developer working on it should deliver instructions for adding the Plugin along with the deliver of the Plugin assembly.

Note: this feature is only for use by SunGard Public Sector Bi-Tech and should not be used to load any other types of Plugins.

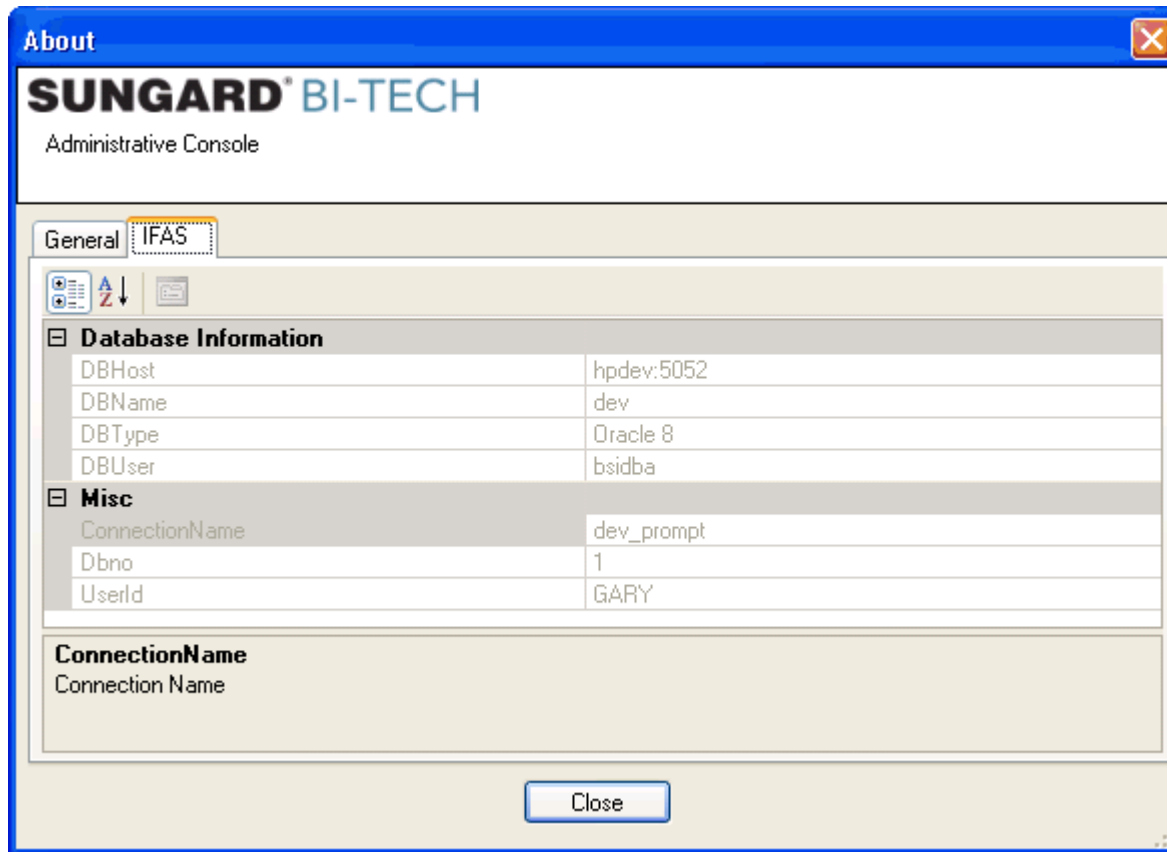
### 3.1.3 Connection Manager

From the “Preferences” menu on the Common Menu you can activate the Connection Manager in an editable form that will allow you to create a new connection or edit the existing connections. Please keep in mind that some of the changes to existing connections can have a significant impact on the system and should be done carefully.



## About

The “Help” menu on the Common Menu provides access to the information about the Administrative Console. The “IFAS” tab allows the user to view all of the information relating to the current IFAS connection and can be beneficial when answering support questions.



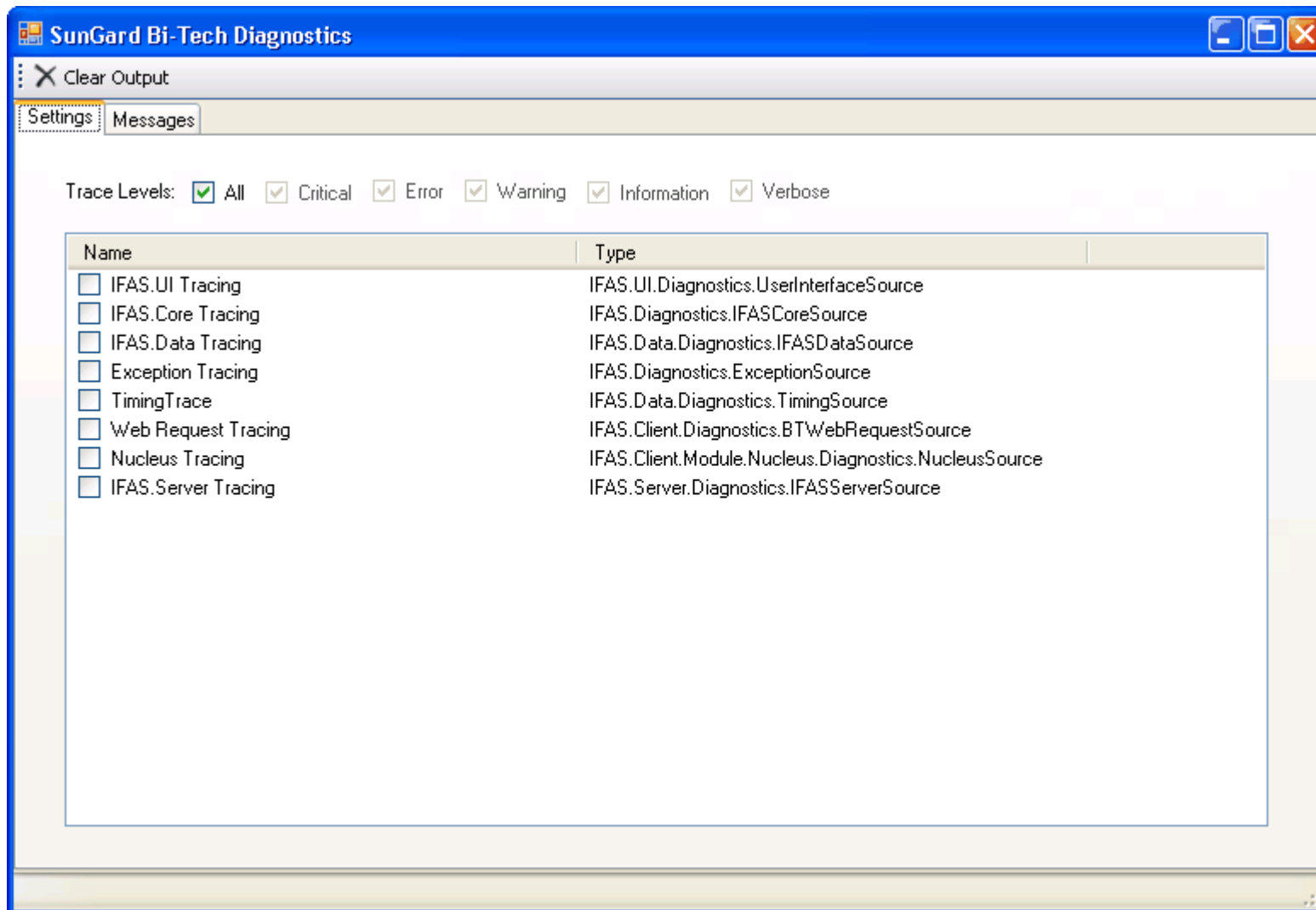
## Tracing

In order to provide tracing in the Web Client screens we needed to provide a new way of tracing in the software for those modules that did not require the user to access the registry. This was done by leveraging the .Net built in tracing. So for those Plugins that are built on that technology the tracing mechanism in the Admin Console is the same as it is in the Web Client screens. However, since the Admin Console is

also capable of making a direct connection to the database it also uses the standard tracing that the 7i software uses which is enabled in the registry.

To minimize the confusion both trace outputs were placed on the same Tracing dialog in the application. Although, the two types are enabled separately they can be viewed in one place. Obviously we would prefer to merge all tracing into one mechanism and hopefully that can be made available sometime in the future.

To enable the tracing go to the “Help” menu on the Common Menu and then select “Support” and “Enable Tracing”. At this point the Diagnostics dialog box should appear.



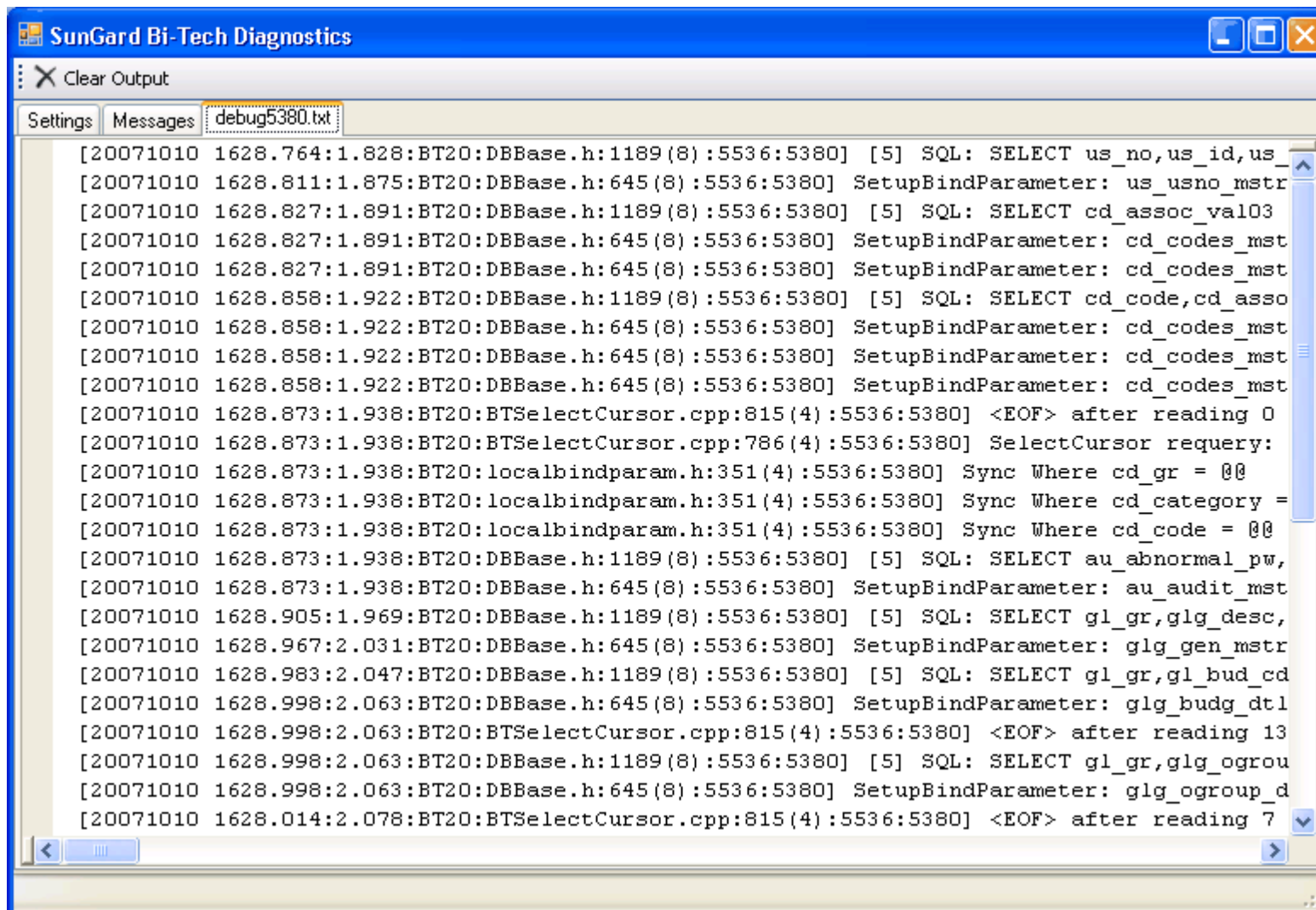
By checking the boxes next to the different tracing names the tracing for that section of code will be enabled. The type of tracing to use will most likely be suggested by a resource at SunGard Public Sector Bi-Tech as part of the troubleshooting process. Some options such as “Exception Tracing” will enable a different tab specific to their type of tracing. This is also true for “Web Request Tracing”.

At any time the “Clear Output” button can be used to clear out the current trace information. This is useful in situations where the amount of trace information has built up to the point where it is becoming difficult to review.

In addition to specifying the type of tracing to enable you can also select how much tracing information to show.

- |             |   |
|-------------|---|
| Critical    | These messages only appear for critical errors that will most likely leave the application in a questionable running state. |
| Error       | These messages are errors the system generates such as business rule errors that are not necessary terminal.                |
| Warnings    | These are warnings typically generated by business rules.   |
| Information | These are general informational trace messages.   |
| Verbose     | This option can result in large amounts of trace information depending on the Plugin.                                       |
| All         | This checkbox enables all the various trace types at once.  |

If the standard tracing for the software has been enabled in the local registry (HKLM\Software\Bi-Tech\Common\Trace) an additional tab will appear monitoring the trace file for the current process. Note: currently that level of tracing must be enabled in the registry prior to launching the Admin Console.



```
[20071010 1628.764:1.828:BT20:DBBase.h:1189(8):5536:5380] [5] SQL: SELECT us_no,us_id,us_
[20071010 1628.811:1.875:BT20:DBBase.h:645(8):5536:5380] SetupBindParameter: us_usno_mstr
[20071010 1628.827:1.891:BT20:DBBase.h:1189(8):5536:5380] [5] SQL: SELECT cd_assoc_val03
[20071010 1628.827:1.891:BT20:DBBase.h:645(8):5536:5380] SetupBindParameter: cd_codes_mst
[20071010 1628.827:1.891:BT20:DBBase.h:645(8):5536:5380] SetupBindParameter: cd_codes_mst
[20071010 1628.858:1.922:BT20:DBBase.h:1189(8):5536:5380] [5] SQL: SELECT cd_code,cd_asso
[20071010 1628.858:1.922:BT20:DBBase.h:645(8):5536:5380] SetupBindParameter: cd_codes_mst
[20071010 1628.858:1.922:BT20:DBBase.h:645(8):5536:5380] SetupBindParameter: cd_codes_mst
[20071010 1628.858:1.922:BT20:DBBase.h:645(8):5536:5380] SetupBindParameter: cd_codes_mst
[20071010 1628.873:1.938:BT20:BTSelectCursor.cpp:815(4):5536:5380] <EOF> after reading 0
[20071010 1628.873:1.938:BT20:BTSelectCursor.cpp:786(4):5536:5380] SelectCursor requery:
[20071010 1628.873:1.938:BT20:localbindparam.h:351(4):5536:5380] Sync Where cd_gr = @@
[20071010 1628.873:1.938:BT20:localbindparam.h:351(4):5536:5380] Sync Where cd_category =
[20071010 1628.873:1.938:BT20:localbindparam.h:351(4):5536:5380] Sync Where cd_code = @@
[20071010 1628.873:1.938:BT20:DBBase.h:1189(8):5536:5380] [5] SQL: SELECT au_abnormal_pw,
[20071010 1628.873:1.938:BT20:DBBase.h:645(8):5536:5380] SetupBindParameter: au_audit_mst
[20071010 1628.905:1.969:BT20:DBBase.h:1189(8):5536:5380] [5] SQL: SELECT gl_gr,glg_desc,
[20071010 1628.967:2.031:BT20:DBBase.h:645(8):5536:5380] SetupBindParameter: glg_gen_mstr
[20071010 1628.983:2.047:BT20:DBBase.h:1189(8):5536:5380] [5] SQL: SELECT gl_gr,gl_bud_cd
[20071010 1628.998:2.063:BT20:DBBase.h:645(8):5536:5380] SetupBindParameter: glg_budg_dtl
[20071010 1628.998:2.063:BT20:BTSelectCursor.cpp:815(4):5536:5380] <EOF> after reading 13
[20071010 1628.998:2.063:BT20:DBBase.h:1189(8):5536:5380] [5] SQL: SELECT gl_gr,glg_ogrou
[20071010 1628.998:2.063:BT20:DBBase.h:645(8):5536:5380] SetupBindParameter: glg_ogroup_d
[20071010 1628.014:2.078:BT20:BTSelectCursor.cpp:815(4):5536:5380] <EOF> after reading 7
```



### 3.1.4 Start Page

The Start Page Plugin to the Admin Console provides a method of easily accessing commonly used Plugins. The top list is the default list of common Plugins shipped with the software. The bottom list is built based on the most recent usage of the Admin Console on a particular person's machine. This list is dynamic and will be based on the usage pattern before the application was last closed. The number of Plugins to track can be controlled by the "Console" settings in the settings dialog.

**Administrative Console** Preferences ▾ Help ▾ Logoff

**IFAS Administrative Console** Start Page ▾ X

**Admin Plugins**







- Documents Online
- CDD Admin
- Database Admin
- Security Admin
- 7i Server Admin

---










**Administrative Console**

This is the SunGard Bi-Tech Administrative Console. Select a plugin from the icons below or browse to the plugin by application on the left.


**Common Administrative Tasks**

 Manage Users	 Manage Security Roles	 Rebuild Security
 Folder Manager	 Configure Local Server	 Job Manager

**Recently Used Plugins**

 SQL Query	 Document Search	 Manage Security Roles
 Manage Security Structure	 Manage Users	 Object Definitions
 Role Simulator	 Database Logging	 Configure Local Server

**Admin Plugins**  Run at Startup

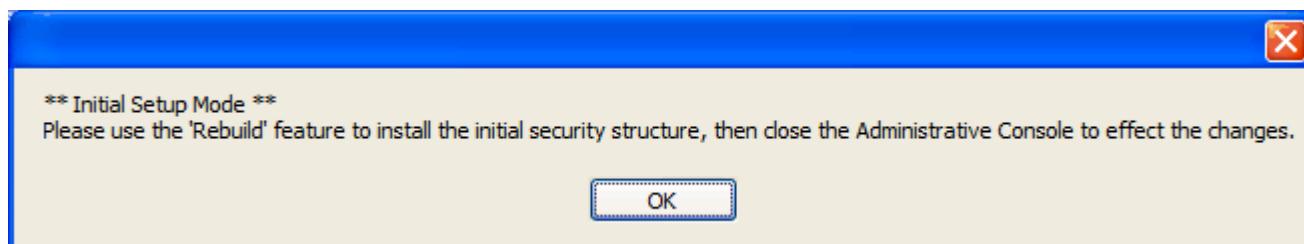
Status SUNGARD<sup>®</sup> BI-TECH  dev\_prompt

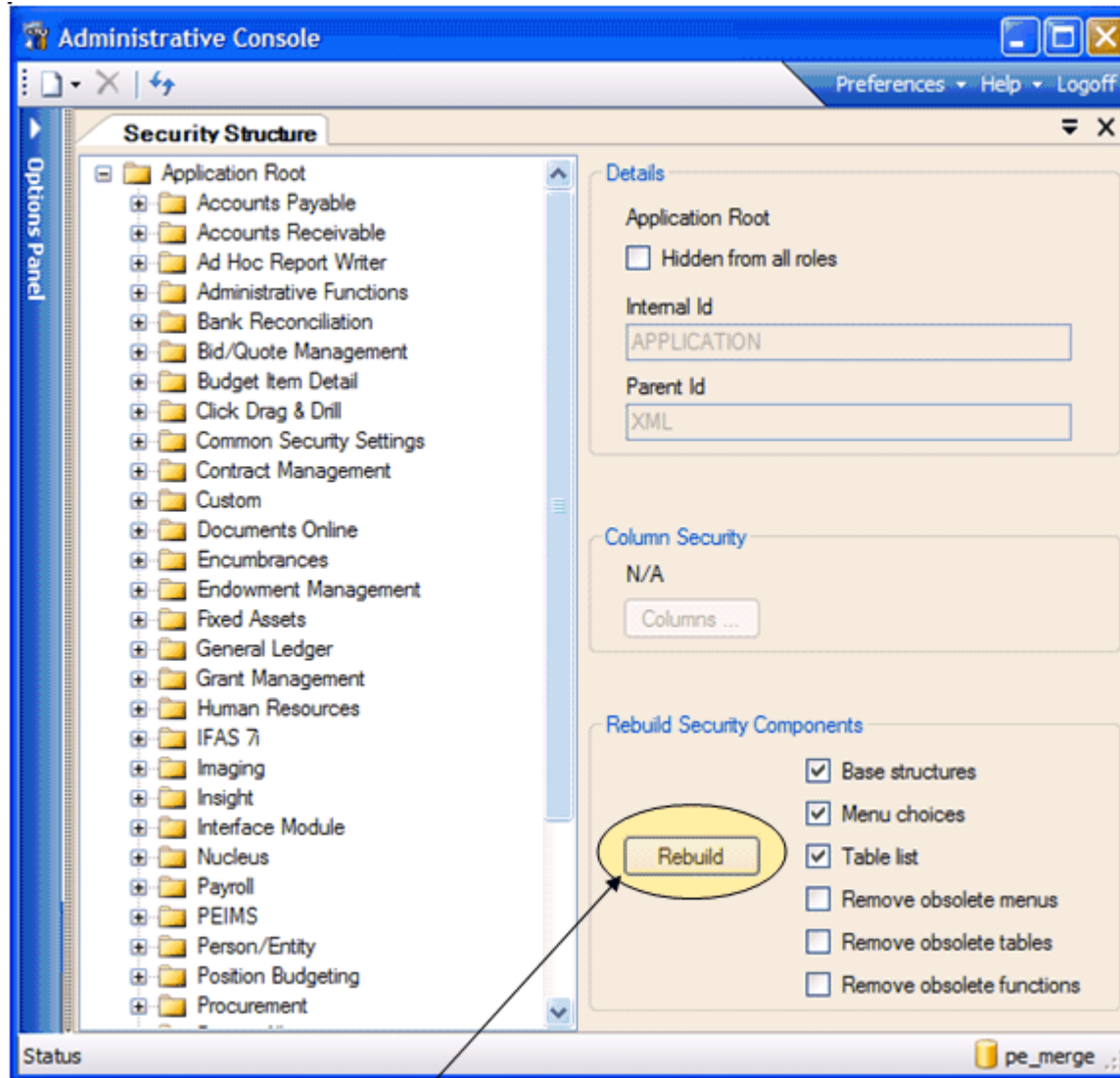
## 3.2 First Time Administrative Console Use

### 3.2.1 Initial setup

When the 7.9 security structures are accessed for the first time, it is possible that there are no users with sufficient security to perform the initial security configuration steps. This “chicken and egg” problem is resolved by the Administrative Console, by automatically granting access upon first use.

Normally the existing “System Administrator” security will be in place from earlier IFAS releases, so one or more users with full security permissions should exist. One of these “full security” users should be used to run the Administrative Console in order to update the security structures. At this point it is likely that the base structure is missing key 7.9 entries, so the Console will display the following message and automatically bring up the “Security Structure” screen.





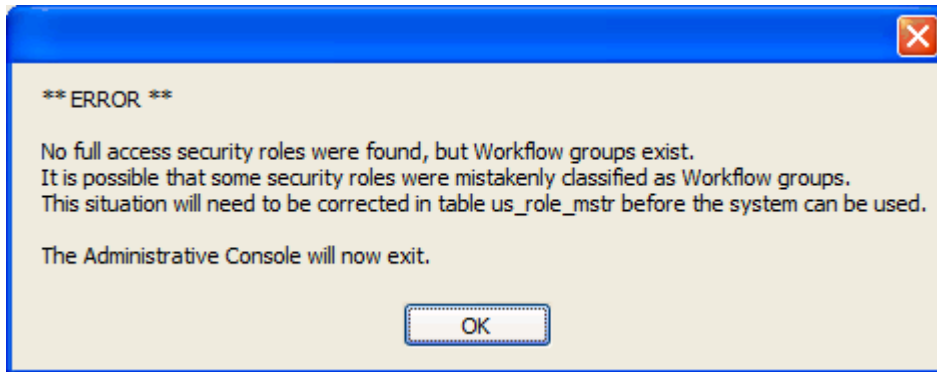
After using the “Rebuild” button on the Security Structure screen to update the security structures and the 7i server, exit from the console and run it again to view the changes. At that point the full list of screens should be available on the Admin Plugins list.

## Problem scenarios

### Workflow groups

In the pre-7.9 System Administrator, the Role Type for each role could be set to “Enable workflow”. This checkbox was used to define Workflow Groups, which are a separate concept from security roles. Any roles defined in this way will not function as security roles.

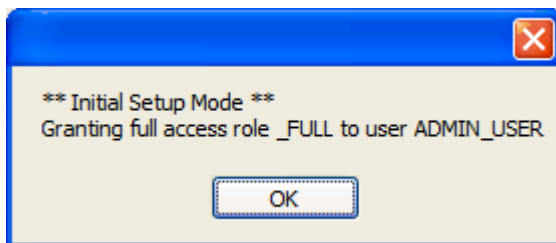
It is possible that the existing security roles are defined as Workflow groups. If this is the case the following message will appear when running the Administrative Console:



Correcting this problem would require manually adjusting the us\_wf\_enable flag within the database table us\_role\_mstr, for all roles which were intended as security roles rather than workflow groups.

### No full access roles

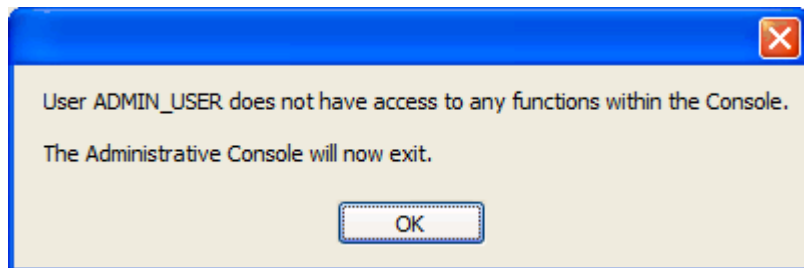
If there are no roles that grant full access and are assigned to one or more users, the Administrative Console will create a new “\_FULL” role upon first use, and grant that role to the current user. If this occurs the following message will be shown:



As indicated, this will automatically create a new `_FULL` role which grants access to all IFAS functions, and will assign that new role to the current user (`ADMIN_USER`, in this example).

### No permissions

If the current user does not have access to any of the administrative functions managed by the Administrative Console, the console will issue the following message and will immediately exit.



To resolve this problem, run the Administrative Console as a user with full permissions. This message should only be issued if one or more users with full capabilities exist on the system.

## 3.3 Configure Local Server

This Administrative Plugin is used to setup the registry entries on the local 7i Server that are required for the application to run properly. The “Save Changes” button is used to save all of the changes to the registry. Multiple changes can be made prior to choosing to save. However, in order to maintain software stability the Data Processing Service must be shutdown prior to the save. The status information on the toolbar maintains the current status of the Data Processing Service. The button to the right of the “Save Changes” button allows you to stop and start the services depending on the current status.

### 3.3.1 Server Config

How the software functions while shutting down the service is configurable. From the “Preferences” menu the “Settings” option will allow you to configure how long to wait on the BTMQHOST processes when shutting down and whether or not to terminate them if they are still not shutdown after the timeout. Please use these settings sparingly as terminating those processes may compromise the stability of the server.

These settings are required by the Data Processing Service during initial startup. They are required for the server to run properly at all.

Upgrade Note: Prior to 7.9 this functionality was handled by the “Configure IFAS Server” option in the System Administrator.

**System Catalog Connection:** This is the ODBC connection string used by the Data Processing Service to locate the system catalog. The “refresh” button on the far right of the field allows the connection string to be refreshed using the current connection.

**IFAS7 Root Directory Path:** This is the path on the local machine where the web components of 7i are installed. Example: “c:\inetpub\wwwroot\ifas7”.

**Server Group:** This is the name of the group that this server will belong to. A group of servers are assumed to share the same connections and for functions such as cache flushing this allows them to be grouped. This name does not necessarily correspond to a domain or load balancing name.

**Active IFAS Connections:** This list allows one or more connections to be activated for this server. All connections must reside in the current System Catalog and at least one connection must be defined per server. Enabling multiple connections can have an adverse impact on server resources.

### 3.3.2 Data Processing Service

These settings control how the Data Processing Service will behave once its in a running state. Prior to 7.9 these settings were accessible from a web page. However, since many of them necessitated a service restart to take effect they were moved to this location where they could be setup while the services were not running.

**Initial System Threads:** Internally the Data Processing Service maintains a thread pool to process incoming service requests. The primary request type is for a list of active connections that are being serviced on the local machine. This thread pool starts with an initial number of threads that are governed by this value. The default value is 4.

**Maximum System Threads:** The internal system request thread pool begins with the number of threads specified by the “Initial System Threads” value. As requests come in a thread from the pool processes them. If there are more requests than threads and the maximum thread value has not been reached, a new thread will be created and added to the pool. The default value is 4.

**Maximum Restarts:** This value controls how many times a particular queue processor will be restarted by the service before ceasing to perform the restart. This value allows you to control having a situation where the service continually restarts the queue processor when the UNIX host is not available. The default value is 0.

**Maximum Restart Failures:** This value controls the maximum number of failures that can occur during the attempt to restart a queue processor. If a queue processor loses its connection to the database and has to be restarted, there may be a time during which the UNIX box will not accept new requests. The default value is 10.

**Maximum Connection Loss:** This value controls how many times a processor may lose its connection to the database before indicating that it needs to be restarted by the service. When a database connection is lost, the queue processor will stop processing messages on the thread that lost the connection and spawn a new thread. The default value is 5.

**Ping Interval:** This value governs the time (in seconds) between inter-group ping messages. Each instance of the Data Processing Service maintains contact with other instances in the same processor group by sending a ping message at an interval specified by this value. This is used to maintain a consistent configuration amongst the servers. The default value is 10.

**Restart Delay:** This value specifies the number of seconds that the Data Processing Service should wait before restarting a processor that has requested restart. In some cases, the back-end database connectivity layer may wait for a particular interval to elapse before accepting new requests; this allows the software to wait for that interval before attempting to establish a new connection. The default value is 10 seconds and the minimum value is 5 seconds.

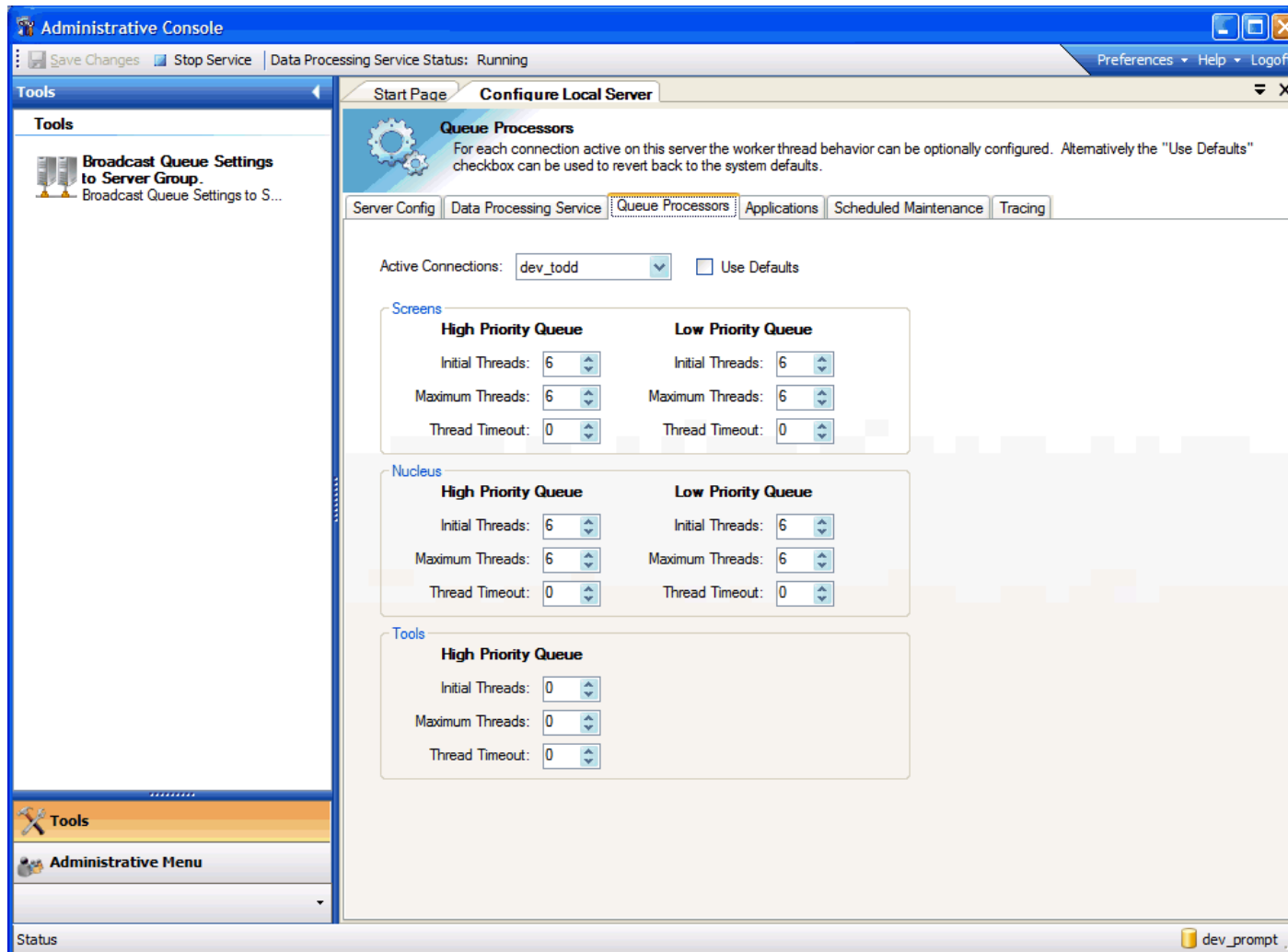
For example, the OpenLink request broker will wait for a specified interval before accepting new connections. This value is controlled by the “ReceiveTimeout” configuration parameter in the OpenLink rulesbook (oplrqb.ini). If this value is set to 30, the restart delay is set to 5 and the maximum restart failure is set to 3, it is possible that the service could attempt to restart the processor 3 times, with each failing, before the OpenLink request broker accepts the request. At this point, the Data Processing Service would not attempt to restart the queue processor and would log an event to the Event Log.

**Restart Processors:** This value controls whether or not the Data Processing Service will attempt to restart a processor that has lost its connection to the database or faulted due to network-related issue, etc. This is enabled by default.



### 3.3.3 Queue Processors

Each connection on the server can have its own thread settings for the Screens and Nucleus queues. Alternatively, the connection can simply use the system defaults.



**Use Defaults:** If checked the system will use the default values as opposed to reading them from the registry.

**Initial Threads:** Internally the Data Processing Service has both a high priority request processor and a low priority request processor to service requests coming in from data entry screens. These processors maintain a thread pool to process incoming service requests. This thread pool starts with an initial number of threads that are governed by this value. The default value is 1. The processes automatically adjust the number of executing threads based on the demand for the service. The number of active thread will be adjusted up until the Maximum Threads setting is reached.

**Maximum Threads:** A screen processors thread pool begins with the number of threads specified by the “Initial Threads” value. As requests come in a thread from the pool processes them. If there are more requests than threads and the maximum thread value has not been reached, a new thread will be created and added to the pool. The default value is  $2 * \text{the number of CPU's} + 2$ . We have successfully used a minimum of 6 with a maximum of 12.

**Thread Timeout:** When a screen’s processor thread has processed a request for a specified period of time without completion, the thread will be timed out, and a timeout status returned to the data entry user. This field defaults to 0, indicating no timeout.

**Propagating the Settings:** Prior to 7.9 the Config web page attempted to broadcast these settings to the other servers in this same server group. This had two issues. First, it was not always clear which settings would and would not get broadcast. Second, not all clients were aware this broadcast was happening. Now the broadcast is no automatic but is instead handled with a tool in the Tools panel of this plugin.

### 3.3.4 Applications

Some applications have settings that specify how that particular application will function while running on a 7i Server.

**Report Expiration:** After a report is run, there are intermediate files that are kept in the directory specified by the File Storage Path parameter. The report expiration value indicates how many seconds a file should be kept on disk before removing the file from the file system. This field defaults to 7.

**File Cleanup Interval:** The file cleanup interval specifies the interval in seconds between runs of the process that removes the temporary intermediate files. This field defaults to 720.

**Initial Instances:** CDD report requests that are made under IFAS 7i are queued to a request queue that is serviced by a number of instances of CDD. The initial instances specifies how many instances of CDD are launched when the first report request is received and serves as a low water mark for how many instances will remain alive. This field defaults to 2. The Initial Instances must be greater than or equal to 2.

**Maximum Instances:** The maximum instance count represents the largest number of CDD instances that will be active at any one time. If the initial instances value is set to 3, the maximum instances is set to 5 and 6 requests arrive, 1 of the requests will be queued until one of the other 5 requests is fully processed by CDD. This field defaults to 5. The Initial Instances must be less than or equal to 25.

**Instance Timeout:** The instance timeout represents the time, in seconds, that an instance of CDD will remain running without running a report. The number of instances will not fall below the low water mark specified by the initial instances value. This field defaults to 15.

**File Storage Path:** The file storage path represents the physical directory under which the temporary CDD files are created. Place this directory on a partition that has at least 1 GB of free disk space. It is advised that you configure the directory for compression. This field defaults to <NetsightRootPath>\cddxml. It is the installer that sets up this default, so it's the value of <NetsightRootPath> at installation time that is used in this default.

**Default Printer:** The default printer is the local printer that is defined during install and needed by IFAS 7i and CDD.net. This printer **MUST** be the same on each 7i web server in the farm. This defaults to Local Printer, which is a 'fake' printer that is setup by the installer.

**Attachment Directory:** The attachment directory represents the physical directory on the 7i web server under which the temporary images are stored. This directory should be placed on a partition that has at least 1 GB of free disk space. It is advised that you configure the directory for compression. This field defaults to <NetsightRootPath>\attachtemp. It is the installer that sets up this default, so it's the value of <NetsightRootPath> at installation time that is used in this default.

**Attachment Cleanup Interval:** The attachment cleanup interval is displayed in minutes and displays how often the attachment directory is cleaned out. This defaults to 1.

### 3.3.5 Scheduled Maintenance

Some of the data in the IFAS database requires regular maintenance and clean up. Because these settings are used against the IFAS database only one server needs to be configured to perform them against that connection. Note: this may require one server in a test farm and one server in a production farm separately configured to perform the maintenance.

The screenshot displays the 'Administrative Console' window with the 'Configure Local Server' tab selected. The 'Scheduled Maintenance' section is active, showing settings for Dashboard, Job Manager, and Logging. The 'Dashboard' section includes 'Dashboard Session Cleanup Interval' (60 minutes) and 'Dashboard Session Cleanup Expiration' (60 hours). The 'Job Manager' section includes 'Job History Cleanup Interval' (60 minutes), 'Job History Expiration' (60 days), and two checked options: 'Remove tailsheets when removing job history' and 'Remove associated documents when removing job history'. The 'Logging' section includes a checked option 'Enable 7i request logging', 'Log File Cleanup Interval' (480 minutes), 'Log File Expiration' (30 days), and 'Log File Location' (c:\dev\Log).

**Administrative Console**  
Save Changes | Stop Service | Data Processing Service Status: Running | Preferences | Help | Logoff

**Tools**  
Broadcast Queue Settings to Server Group.  
Broadcast Queue Settings to S...

**Configure Local Server**  
Start Page | **Scheduled Maintenance** | Server Config | Data Processing Service | Queue Processors | Applications | Tracing

**Scheduled Maintenance**  
Scheduled maintenance and database cleanup functions can be setup to run on a 7i server. Note: Since these functions are run against the IFAS Database only one server is required to perform them currently.

**Dashboard**  
Dashboard Session Cleanup Interval: 60 minutes  
Dashboard Session Cleanup Expiration: 60 hours

**Job Manager**  
Job History Cleanup Interval: 60 minutes  
Job History Expiration: 60 days (0 to disable)  
 Remove tailsheets when removing job history  
 Remove associated documents when removing job history

**Logging**  
 Enable 7i request logging  
Log File Cleanup Interval: 480 minutes  
Log File Expiration: 30 days (0 to disable)  
Log File Location: c:\dev\Log

Status | dev\_prompt

## Dashboard

**Dashboard Session Cleanup Interval:** Defines how often (in minutes) the clean up process should wake up and check for expired Dashboard Session blobs.

**Dashboard Session Cleanup Expiration:** Defines how long (in hours) to leave a Dashboard Session blob before it is considered expired and removed.

## Job Manager

**Job History Cleanup Interval:** Defines how often (in minutes) the clean up process should wake up and check for expired job history.

**Job History Expiration:** Defines how old (in days) a job must be before it is considered expired and its history removed. Set this value to 0 if job history is not to be removed.

**Remove tailsheets when removing job history:** If this option is enabled, when expired job history is removed the “tailsheet” output for each job will be removed as well.

**Remove associated documents when removing job history:** If this option is enabled, when expired job history is removed all related documents in the Documents Online system (referencing the same job number) will be removed as well.

## Logging

**Enable 7i request logging:** Requests made to the 7i server are logged temporarily in thread-specific files which exist only for the duration of the request. The log files include a timestamp, the xml request, any business rules called, and any insert/update/delete database operations performed. If a request does not complete for some reason the log file will remain on the system, providing clues as to what may have occurred. Log files which remain may eventually be removed, depending on the log file expiration settings. Request logging is enabled by default but it may be disabled by unchecking this option.

**Log File Cleanup Interval:** Defines how often (in minutes) the clean up process should wake up and check for expired log files.

**Log File Expiration:** Defines how old (in days) a log file must be before it is considered expired and deleted from the log directory. Set this value to 0 if the request log files are not to be removed.

**Log File Location:** Specifies the directory location for the request log files and the “Extended run time” log files (which are created when a request takes more than a predefined time interval which is typically 30 seconds). By default, this location is a “Log” directory contained within the 7i install directory. For example, “C:\Program Files\SunGard Bi-Tech\Log”.

### 3.3.6 Tracing

Prior to 7.9 any server-side tracing was done by manually editing the registry. This tab simplifies the process of setting up the tracing on a server. Tracing changes do not take effect until the Data Processing Service is restarted.

**File Path:** The path to place any tracing files on the server. The path must exist prior to enabling tracing.

**Enabled:** This checkbox enables or disables tracing on the server.

**Add Module:** This edit box in combination with the “Add” button adds a module to the module list.

**Import:** This option allows you to import the module list from the reglist.xml file in the install directory and then select from that imported list which modules to add to the local tracing setup.

**All Modules:** This option enables tracing for all modules. Warning: this could have a very bad impact on performance on the 7i server.

**Modules:** This is the collection of modules identified on the local server. Checking a module will enable tracing for that particular module.

## 3.4 Job Manager

With the 7.9 release of IFAS a new method of tracking jobs was added to the software. This new tracking is referred to as the Job Manager. Essentially anything classified as “Job” in the software is tracked in the NU\_JOB\_MSTR table in the IFAS Database. The plugin used to monitor the Jobs is the “Job Manager” which is accessible either from the Admin Console or from the SYUTJM menu mask in IFAS.

The screenshot displays the 'Job Manager' window within the 'Administrative Console'. The interface includes a search panel on the left and a main table of job records. The table is organized into three sections: 'Completed', 'Failed', and 'Cancelled'.

Job Number	Mask	Description	User	Status	Type	Start	End
<b>Completed</b>							
360374	HREMPR	SetAndSyncPinwayContact	TOM	OK	Tool Execution		
360364	GL_GET_IF_...	Completed, 12 Page(s)	MKESM	OK	CDD 7i Report		
360363	GLTRANSR...	Completed, 1065 Page(s)	MKESM	OK	CDD 7i Report		
360352	POUPPR	REBUILD_SECURITY	GARY	OK	Workflow		
360351	POUPPR	REBUILD_SECURITY	GARY	OK	Workflow		
360350	POUPPR	REBUILD_SECURITY	GARY	OK	Workflow		
360343	POUPPR	REBUILD_SECURITY	GARY	OK	Workflow		
360338	POUPPR	REBUILD_SECURITY	GARY	OK	Workflow		
360325	POUPPR	REBUILD_SECURITY	GARY	OK	Workflow		
360315	PE0004_37...	Completed, 10 Page(s)	GARY	OK	CDD 7i Report		
360314	PE0004_37...	Completed, 1 Page(s)	GARY	OK	CDD 7i Report		
360309	HREMPR	CascadeChanges	TOM	OK	Tool Execution		
360308	HREMPR	CascadeChanges	TOM	OK	Tool Execution		
360307	HREMPR	CascadeChanges	TOM	OK	Tool Execution		
360305	HREMPR	SetAndSyncPinwayContact	TOM	OK	Tool Execution		
360304	HREMPR	CascadeChanges	TOM	OK	Tool Execution		
360296	HREMPR	CascadeChanges	TOM	OK	Tool Execution		
360295	HREMPR	CascadeChanges	TOM	OK	Tool Execution		
360294	HREMPR	SetAndSyncPinwayContact	TOM	OK	Tool Execution		
<b>Failed</b>							
360362	DwCPARIM...	DwCPARIM_507_581	MKESM	FL	CDD 7i Report		
360360	Ticket335168	Ticket335168	MKESM	FL	CDD 7i Report		
360324		REBUILD_SECURITY	GARY	FL	Workflow		
360322		REBUILD_SECURITY	GARY	FL	Workflow		
360320	PE0004_37...	PE0004_374278	GARY	FL	CDD 7i Report		
360313		REBUILD_SECURITY	GARY	FL	Workflow		
<b>Cancelled</b>							
360319	PE0004_37...	Report Cancelled	GARY	CA	CDD 7i Report		
360318	PE0004_37...	Report Cancelled	GARY	CA	CDD 7i Report		

The interface also features a search panel on the left with fields for Job Number, Mask, Description, and User ID, along with a date range selector (From: 8/21/2007, To: 8/21/2007) and a Search button. The status bar at the bottom indicates 'Complete (Last Updated: 2:15:49 PM)' and the URL 'http://localhost/ifa7? dev\_prompt'.

### 3.4.1 What is a Job?

Traditionally in IFAS a job was a menu mask run in job mode on the Application Server. This definition of a job still applies, but the concept also been extended to include CDD Reports, Tools launched from screens, utilities, etc. Essentially anything in the software that generates an IFAS Job Number can be tracked using the Job Manager.

For additional details on the types of jobs IFAS produces and the internal details of job management, please refer to the section entitled “IFAS Job Management”.

**Refresh:** This toolbar button refreshes the Job Manager information. However, the Job Manager will refresh itself automatically based on the user settings.

**Filter By Type:** Since there can be a significant amount of data in the Job Manager this option allows the user to remove job types that may not be important currently. Example: Classic Interact jobs are interactive jobs on the Application Server and may not be essential depending on the needs of the user monitoring job status on the system.

**Filter By Status:** This option allows jobs to be filtered based on their current status. This allows the user to potentially remove large numbers of completed jobs and help make the list more manageable.

**Top-Level Group:** This option changes to the top level grouping in the Job Manager to be based on Status, Type or User. Currently only one top-level grouping can be defined at any given time.

**Cancel Job:** If the IFAS Job currently selected can be cancelled this toolbar button will be enabled as well as a “Cancel Job” context menu item. This allows the user to set the status of the Job to be “CR” (Cancel Requested). Then once the software reaches a point where it can safely cancel the job the job will be terminated and the status automatically changed to “CA” (Cancelled). Whether or not a job can be cancelled is based on both the type of job and the current status of the job.

**Search:** By default the Job Manager lists all of the jobs created with today’s date. However, the Job Search panel allows the user to search for other jobs in the system based on the criteria listed in the panel. To return the Job Manager to its default simply clear the search criteria and choose “Search” again.



**Properties:** To view the full property list for a given job simply right-click on the desired job and select the “Properties” menu option. This is the full list of information about a given job.




The screenshot shows a window titled "Job Properties" with a blue header and a close button. Below the header, there is a section with a group of three people icon, displaying "Job Number: 360350" and "Type: Workflow". Below this is a toolbar with icons for list view, sort (A-Z), and print. The main area is a table with the following data:

EndDt	
JobNo	360350
StartDt	
Status	OK
Type	Workflow
Url	https://todd.chico.bi-tech.com/ifas7/
UsId	GARY
UsNo	5
<b>[-] Nucleus</b>	
BatchFile	
CheckDt	
Mask	POUPPR
MPE	0
PostDt	
<b>[-] Progress</b>	
MiscKey	
Page	0

At the bottom of the window, there is a section titled "BatchFile" with the text "The batch file used by this job."

**View Tailsheet:** If the Job being viewed has tailsheet information available it can be accessed either by double clicking the job in the list or by right-clicking and selecting “View Tailsheet”. Tailsheets can include output on the Application Server or Tailsheets archived in Documents Online.

When one or more documents are found in Documents Online, a “+” sign will be shown next to the job. Clicking the “+” will expand the list of available documents, which may be opened by double-clicking.

	 379039	NUUPUS	User Security Audit	GARY	OK	Tool Execution	11/20/2007 11:38
		User Audit for 00000 (Doc ID: 10679)					
	 379040	NUUPUS	User Security Audit	GARY	OK	Tool Execution	11/20/2007 11:39
	 379067	TDHRUTAT	ATA Emppay Rollover	KURT2	OK	Workflow	11/20/2007 1:12:

In the case of classic jobs, the “View Tailsheet” option will attempt to contact the application server to obtain the first 100 lines of the tailsheet. Note: This option requires some configuration on the server (x\_print\_cmds) to ensure that copies of all tailsheets are maintained in a known location.

In the case of Workflow Jobs which are in progress, the “View Tailsheet” option will read the relevant ifas\_output\_dtl records from the database and present the incomplete tailsheet information for the job in progress.

**User Settings:** To control how often the job information is refreshed select the “Settings” option from the toolbar. The “Job Manager” tab allows the user to select the frequency of the Job Status refresh queries.



### 3.4.2 IFAS Job Management

IFAS Jobs represent work that is being done on a user’s behalf by the 7i server or the application server. There are several types of jobs, but these are all tracked through the same mechanism and consolidated on the same “Jobs” list. The possible job types are listed below.

**Workflow** – Jobs processed by the 7i Workflow engine which may be submitted from multiple areas of 7i

**Classic Job** – Jobs processed by the application server, submitted either through the Home page, the Dashboard, or a classic session

**Classic Interactive Job** – Jobs run interactively through a classic session (Insight, telnet, or Reflection)

**CDD Report** – Reports run from CDD installed on a client PC

**CDD 7i Report** – Reports run through 7i

**Tool Execution** – Tools requested from the options bar on a 7i screen

The following sections describe some internal details of IFAS job management.

## Job Tracking

When any of the possible job types are introduced into the system, a record is created in the table nu\_job\_mstr. This record will track the progress of a job from introduction to completion, including any success or failure information and other relevant details. The information shown on the 7i “Jobs” page and the Job Manager screen is collected from these nu\_job\_mstr records.

Job records will remain in nu\_job\_mstr until they expire and are removed, based on the settings defined in the “Configure Local Server” console plugin. When the record of a job is removed, the corresponding job output may optionally be removed as well.

Within nu\_job\_mstr, the status of a job is encoded in the column nuj\_status, which will be one of the following values:

CA	Cancelled	(by user request or due to job limit)
CR	Cancel Requested	(pending cancellation by the Workflow Engine)
FL	Failed	(completed unsuccessfully or terminated)
IN	Initialized	(newly created job)
IP	In progress	(started by the Workflow Engine, processing)
OK	Successful completion	
QU	Queued, pending	(job not yet executing)
WF	Queued by Workflow	(instance not yet processed by Workflow Engine)

A few key nu\_job\_mstr fields which may be useful:

nuj\_type Job type as text ( “CDD Report”, “Classic Job”, “Workflow”, “Utility Job”, etc.)

nuj\_progress Text indicating progress or most recent activity within the job

nuj\_start Start date/time

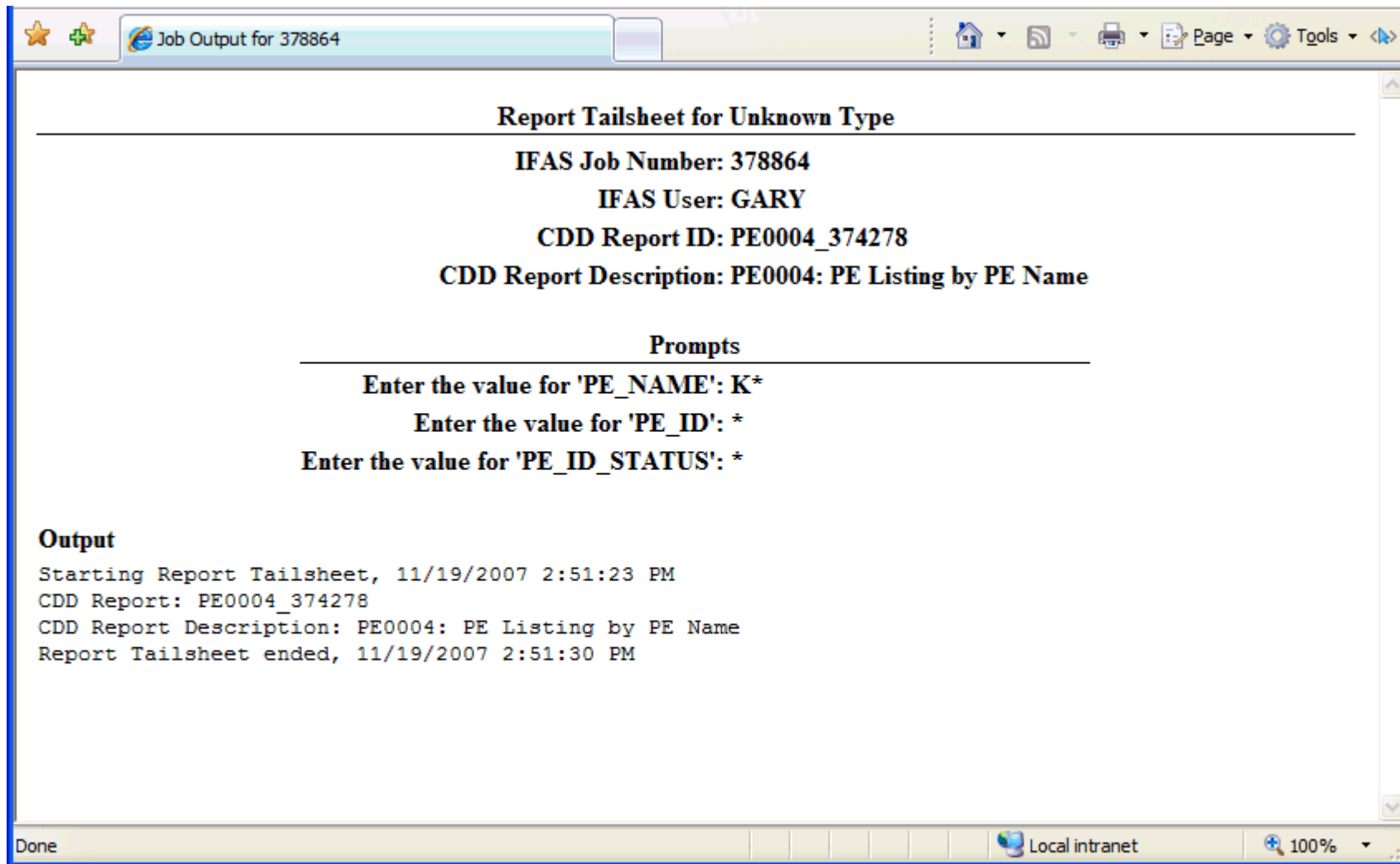
nuj\_end Ending date/time

create\_when Date/time job was introduced to the system

us\_id IFAS user submitting the job

## Job Output

There are two main types of output produced by IFAS: Reports and Tailsheets. Reports are created by all jobs that are specifically designed to produce output. Tailsheets are produced by the system as a record of the job itself, including details about the job request, the actual processing, and the results where relevant. Here is an example of a simple tailsheet for a job of type “CDD Report”:



**Report Tailsheet for Unknown Type**

---

**IFAS Job Number: 378864**  
**IFAS User: GARY**  
**CDD Report ID: PE0004\_374278**  
**CDD Report Description: PE0004: PE Listing by PE Name**

**Prompts**

---

**Enter the value for 'PE\_NAME': K\***  
**Enter the value for 'PE\_ID': \***  
**Enter the value for 'PE\_ID\_STATUS': \***

**Output**

```
Starting Report Tailsheet, 11/19/2007 2:51:23 PM
CDD Report: PE0004_374278
CDD Report Description: PE0004: PE Listing by PE Name
Report Tailsheet ended, 11/19/2007 2:51:30 PM
```

Done Local intranet 100%

Job output is stored within Documents Online. In addition to being viewable through links on the Job Manager screen, the job output and tailsheets may also be seen using the Documents Online screens. Please refer to the Documents Online guide for details on viewing documents.

## Classic Job Output

Classic IFAS jobs run on an application server may also have their output redirected to Documents Online, but this requires some additional configuration. To use this feature it is necessary to specify a particular printer (typically called “workflow”), which must be defined in the `$XPORTDIR/scripts/x_print_cmds` file on the application server. Typically that printer definition looks like this:

```
workflow)
    wfqueue -v -u $3 -m $5 -j $8 $1 >>/tmp/wfqueue.log 2>&1
;;
```

In most cases it is also desirable to store classic tailsheets in a location which is viewable from 7i. This requires using a Documents Online printer (“workflow”), or copying all `STDLIST` output (the tailsheets) to a `STDLISTS` subdirectory. When the `STDLISTS` subdirectory is used, the tailsheet copies are retained on the application server and can be partially viewed (the first hundred lines) using the link on the Jobs page. In this configuration, the application server file `$XPORTDIR/scripts/x_print_cmds` should contain logic similar to the following:

```
case $5
in
STDLIST) export auditdir=$XPORTDIR/.spool/STDLISTS
    shortname=`basename $1`
    cp $1 $auditdir/$shortname
    mask=`grep ":JOB" $shortname|cut -c5-20|cut -f1 -d", "`
```

```
echo "$mask `date` $1 $3" >> $auditdir/STDLIST.log  
exit  
;;  
esac
```

This logic moves each classic tailsheet into the xport/.spool/STDLISTS directory which is searchable by the 7i Jobs page.

## Workflow Jobs

Many of the jobs available in the Home or Dashboard pages, as well as many of the tools within 7i screens, have been implemented as Workflow jobs. Workflow jobs use a particular type of Workflow model in order to process a particular function which must run in the background and which is to be tracked as a job. In most cases the generic “JOB” model is used for Workflow job processing, but specific models may also be used for jobs with higher complexity or specific needs.

When a Workflow job is requested, a record will be created in the wf\_instance table which stores Workflow instances. The Workflow engine is responsible for processing these instances. When the engine picks up an instance which is marked for job processing, it will automatically track the job in the nu\_job\_mstr table. As the job progresses through the various activities defined in the model, the job progress will be updated. A typical Workflow job model contains only a Process activity, which runs a particular 7i utility within a separate process (launcher.exe). When the activity is complete, the status of the Workflow instance is updated, indicating that the engine should again pick up the instance for any additional processing that is needed.

For further details on workflow jobs, please refer to the Workflow user guide.



## 3.5 Monitor Servers

This plugin provides a mechanism for administrators to monitor the servers on a 7i server farm. The “Servers” panel lists all the servers currently being monitored. Selecting a server from that list will populate the main plugin area with the information from that server.

**Administrative Console** Preferences Help Logoff

Refresh Add Servers Login to Server

---

**Servers** Start Page Monitor Servers

**UPSTAIRS**

- david.chico.bi-tech.com
- todd.chico.bi-tech.com

**KURT**

- kurt.chico.bi-tech.com

**FARM202**

- Dev.chico.bi-tech.com

Server Status Updated 7:52:36 AM

**Servers**

**Tools**

**Admin Plugins**

**Dev.chico.bi-tech.com**  
7.9.0.116  
dev\_prompt  
http://dev.chico.bi-tech.com/ifas7

Server Functioning Properly

General Status CDD Report Queue Request Broker Status Log

Authentication	
Connection	dev_prompt
UserID	GARY
Misc	
Connections	dev_prompt
HostName	Dev.chico.bi-tech.com
LastUpdate	12/6/2007 7:52 AM
ServerGroup	FARM202
ServerUrl	http://dev.chico.bi-tech.com/ifas7
Status	OK
StatusRequest	<sbixml><NetSightMessage><Header><UserID>_SUPPORT</UserID></Header><Body></Body></NetSightMessage></sbixml>
StatusResponse	<sbixml><NetSightMessage><Header><Connection>dev_prompt</Connection></Header><Body></Body></NetSightMessage></sbixml>
StatusText	Server Functioning Properly
Version	7.9.0.116

**Connection**  
Connection name used to authenticate to this server.

Complete

Status http://localhost/ifas7 dev\_prompt

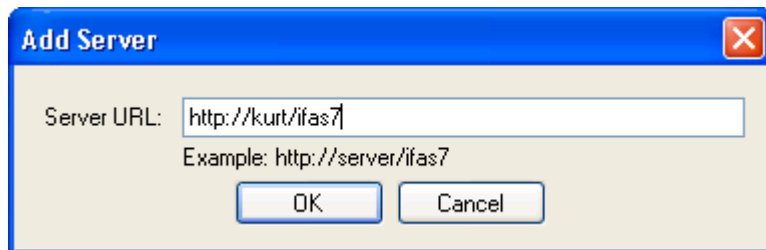
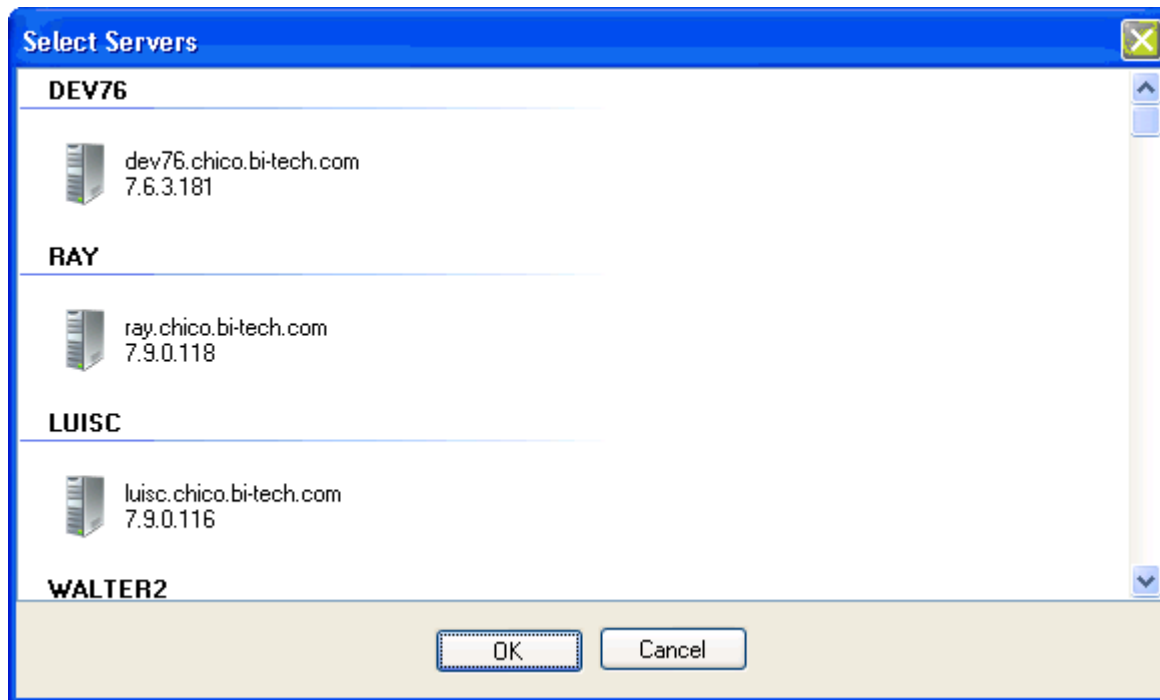
You will be automatically authenticated to any server using the active connection the same as the one you connected to the Admin Console with. However, those servers with a different list of active connections will require you to login to that server in order to perform status checks on it. To login simply select the server you wish to authenticate to and click the “Login to Server” button on the toolbar. From there the login dialog will appear allowing you to select a connection and enter your login and password.



The image shows a Windows-style dialog box titled "Connect to a 7i Server". The dialog has a blue header bar with a close button (X) in the top right corner. Below the header, the SunGard BI-TECH logo is displayed. A message reads: "Please enter the 7i Server Url, select the desired Connection and enter your IFAS User and IFAS Password." The main area of the dialog is titled "SunGard WebClient Login" and features a globe icon with a user profile icon overlaid. The form contains the following fields and controls:

- Server URL: A text box containing "http://todd.chico.bi-tech.com/ifas7" with a dropdown arrow on the right.
- Connection: A dropdown menu showing "dev\_todd" with a dropdown arrow on the right.
- IFAS User: A text box.
- IFAS Password: A text box.
- A checkbox labeled "Remember Login" next to the IFAS User field.
- At the bottom, there are two buttons: "Login" and "Cancel".

Additional servers can be added to the list of servers by selecting the “Add Servers” dropdown from the toolbar and adding them from the Server List Query and selecting one or more servers or by entering the server information directly.



After adding a new server it may take a few moments for the server’s information to be queried and its status made available in the plugin. Simply adding a server may not allow it to be fully monitored. If the new server does not have the connection active that was selected when launching the Admin Console the user will need to authenticate directly to that server.

### 3.5.1 General Status

The “General Status” tab allows access to the basic information about that server. This information is updated each time a status request is made to that server. The information is read-only and is here for informational purposes only.

Setting	Description
Connection	The name of the IFAS connection used to authenticate to the server.
UserID	The userid used to authenticate to the server.
Connections	The active connection list for that server
HostName	The hostname for that server.
LastUpdate	The last time the server status was updated by the plugin for that server.
ServerGroup	The server group this server belongs to in the farm.
ServerURL	The URL used to access the 7i software on the server.
Status	The last status retrieved from the server.
StatusRequest	The XML status request for that server. This is an XML file stored on the server itself in the IFAS7/Admin directory named “ServerStatusRequestsDefault.xml”.
StatusResponse	The last XML response from the status request.
StatusText	The text representation of the server’s current status.
Version	The version of the 7i software running on the server.

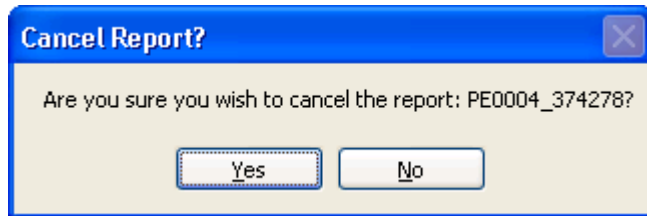
## 3.5.2 CDD Report Queue

The “CDD Report Queue” tab allows the administrator to view and cancel CDD reports from the CDD Queue on that server.

The screenshot displays the Administrative Console interface for monitoring servers. The main window is titled "Monitor Servers" and shows the "CDD Report Queue" tab selected for the "localhost" server. The interface includes a sidebar with server groups (UPSTAIRS, KURT, FARM202) and a main panel with tabs for General Status, CDD Report Queue, Request Broker, and Status Log. The CDD Report Queue tab is active, displaying a table of reports with columns for User, Report, Status, Page, Time Queued, Time Started, and Time Completed. The table shows two reports: one running and one queued.

User	Report	Status	Page	Time Queued	Time Started	Time Completed
GARY	PE0004_374278	Running	3	12/6/2007 2:18:00 PM		
GARY	PE0004_374278	Queued:1	0	12/6/2007 2:18:00 PM		

To cancel a report, select the report from the list and choose the “Cancel Report” button from the tab’s toolbar. The software will prompt you to confirm the cancellation and then issue a cancel request to the server for that report.



Once the cancel request has been issued the panel will report that in a status message at the top of the plugin.

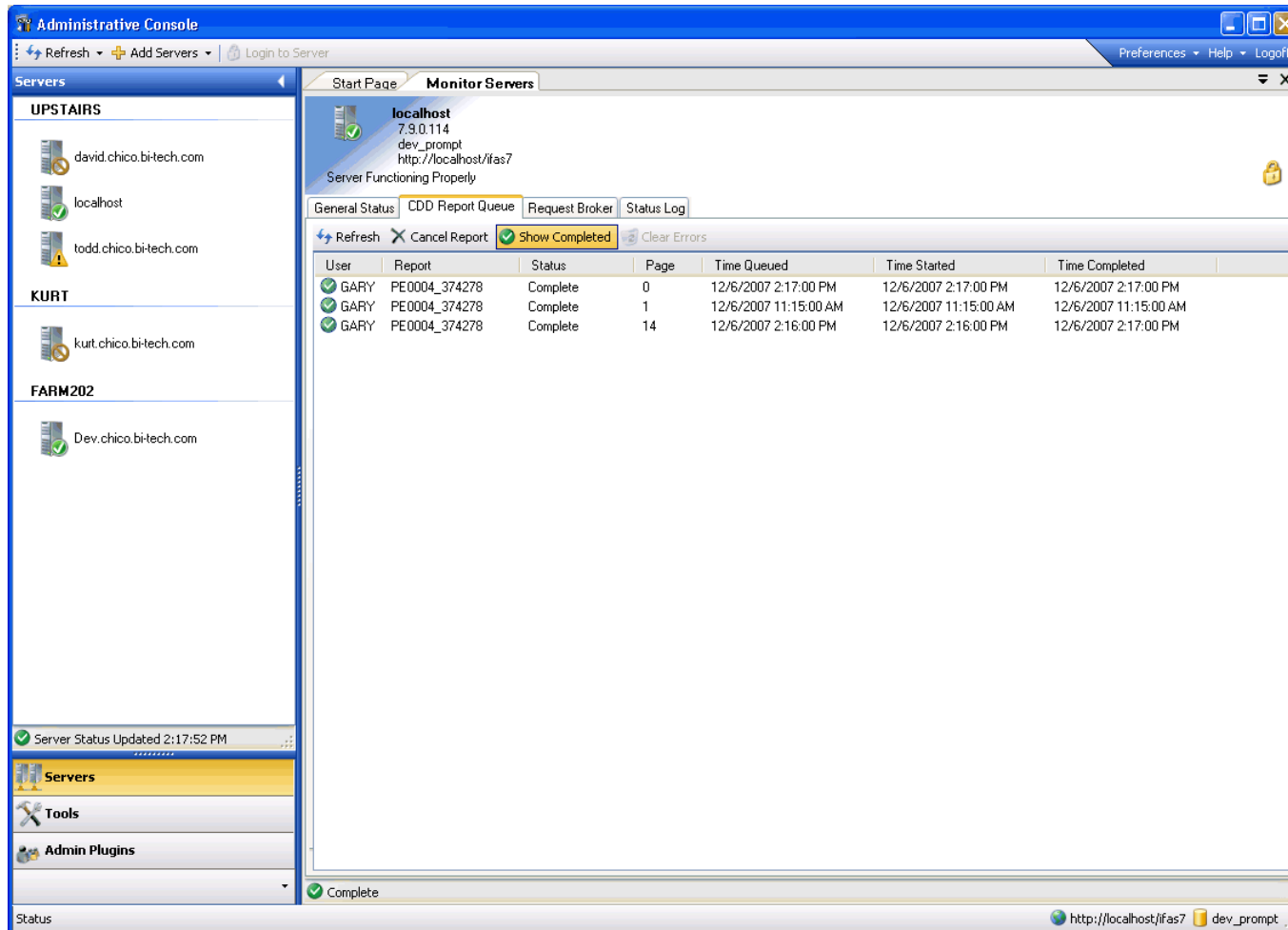
The screenshot displays the Administrative Console interface. The main window is titled 'Monitor Servers' and shows a status message 'Report Cancelled.' at the top. Below this, the 'localhost' server is shown with a green checkmark and the text 'Server Functioning Property'. The interface includes a sidebar with server groups: UPSTAIRS (david.chico.bi-tech.com, localhost, todd.chico.bi-tech.com), KURT (kurt.chico.bi-tech.com), and FARM202 (Dev.chico.bi-tech.com). The main content area has tabs for 'General Status', 'CDD Report Queue', 'Request Broker', and 'Status Log'. The 'CDD Report Queue' tab is active, showing a table with the following data:

User	Report	Status	Page	Time Queued	Time Started	Time Completed
GARY	PE0004_374278	Starting	0	12/6/2007 2:17:00 PM		

At the bottom of the console, there is a status bar with a green checkmark and the word 'Complete'. The address bar shows 'http://localhost/ifa7' and 'dev\_prompt'.



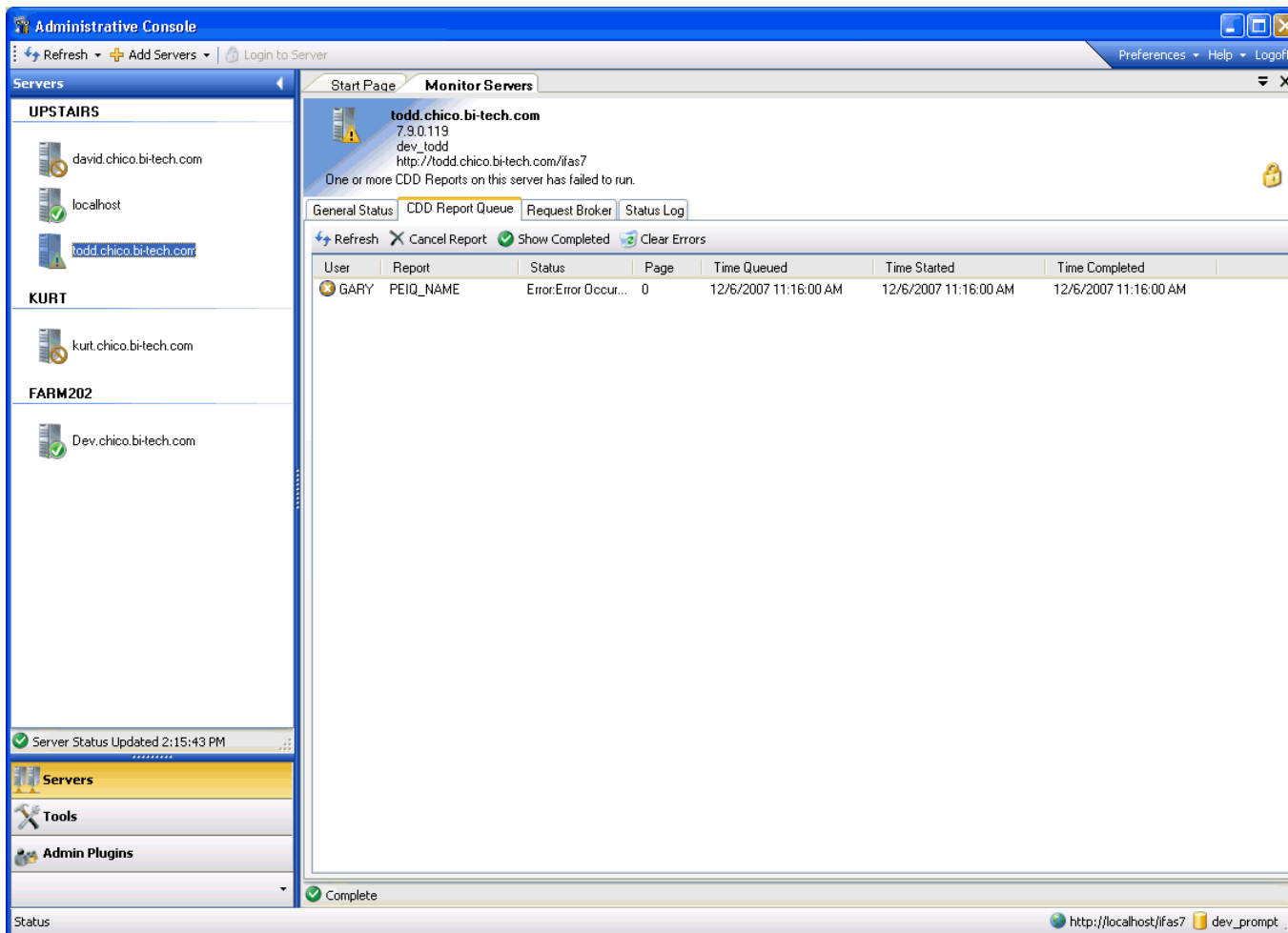
By default the CDD Report Queue tab will not show completed reports. However, the “Show Completed” option on the tab’s toolbar will allow completed reports to appear in the list.



The screenshot shows the Administrative Console interface. The left sidebar lists servers under three categories: UPSTAIRS (david.chico.bi-tech.com, localhost, todd.chico.bi-tech.com), KURT (kurt.chico.bi-tech.com), and FARM202 (Dev.chico.bi-tech.com). The main window is titled 'Monitor Servers' and shows details for 'localhost' (7.9.0.114). The 'CDD Report Queue' tab is active, and the 'Show Completed' checkbox is checked. The report queue table shows three completed reports for user GARY.

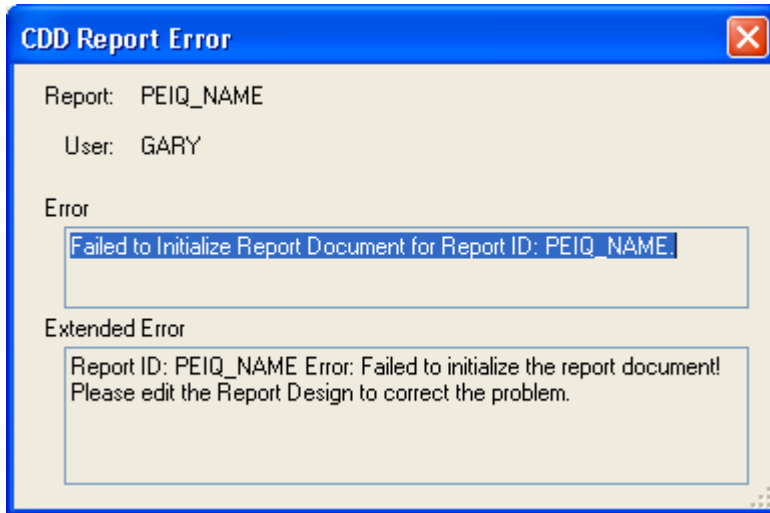
User	Report	Status	Page	Time Queued	Time Started	Time Completed
✓ GARY	PE0004_374278	Complete	0	12/6/2007 2:17:00 PM	12/6/2007 2:17:00 PM	12/6/2007 2:17:00 PM
✓ GARY	PE0004_374278	Complete	1	12/6/2007 11:15:00 AM	12/6/2007 11:15:00 AM	12/6/2007 11:15:00 AM
✓ GARY	PE0004_374278	Complete	14	12/6/2007 2:16:00 PM	12/6/2007 2:16:00 PM	12/6/2007 2:17:00 PM

In the event that an error occurred while running the report the icon and status of that report in the list will change. Additional details about the error can be viewed by double clicking the report in the list. In addition to showing the error in the tab the icon for that server will also change



The screenshot shows the Administrative Console interface. The main window is titled "Monitor Servers" and displays details for the server "todd.chico.bi-tech.com". A warning icon is visible, and a message states "One or more CDD Reports on this server has failed to run." Below this, a table shows a report for user "GARY" with status "Error:Error Occur...". The table columns are User, Report, Status, Page, Time Queued, Time Started, and Time Completed.

User	Report	Status	Page	Time Queued	Time Started	Time Completed
GARY	PEIQ_NAME	Error:Error Occur...	0	12/6/2007 11:16:00 AM	12/6/2007 11:16:00 AM	12/6/2007 11:16:00 AM



### 3.5.3 Request Broker

The Request Broker information page is available on the “Request Broker” tab. This page provides basic statistical information about the performance of request processing on the server.

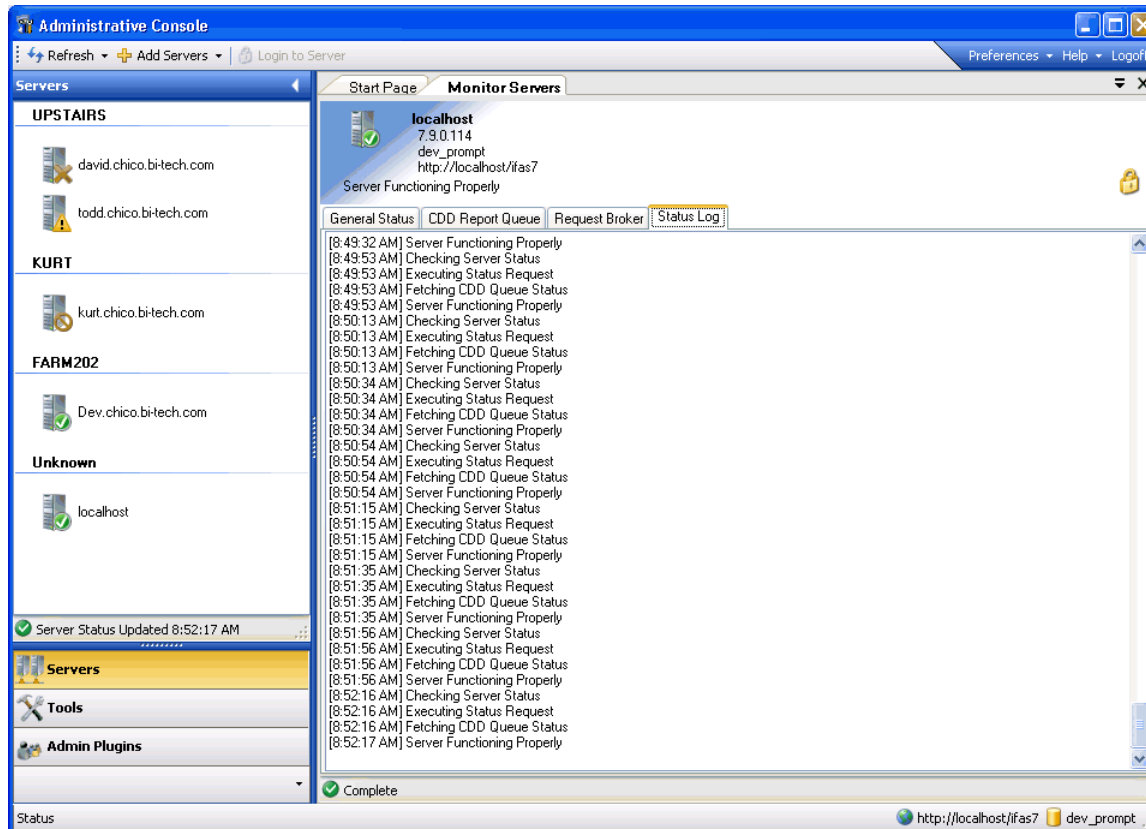
The screenshot shows the Administrative Console interface. The main content area displays the 'Request Broker' status for the 'localhost' server. The status is 'Server Functioning Properly'. The 'Request Broker' tab is selected, showing the following statistics:

Request Broker Stats	
<b>Process Information</b>	
Process ID	2860
<b>Request Information</b>	
Total Processed	7
Response Pending	0
Requests Queued	0
Orphaned	0
Timeouts Possibly Still Running	0
Max Waiting	0
Timeout	30000
<b>Response Times</b>	
Overall	828
Last 500	828
Last 1000	828
Last 1500	828
Last 2000	828
<b>Thread Info</b>	
Request Threads	10

The interface also shows a sidebar with server groups: UPSTAIRS (david.chico.bi-tech.com, todd.chico.bi-tech.com), KURT (kurt.chico.bi-tech.com), FARM202 (Dev.chico.bi-tech.com), and Unknown (localhost). The bottom status bar indicates 'Complete' and the URL 'http://localhost/ifa7'.

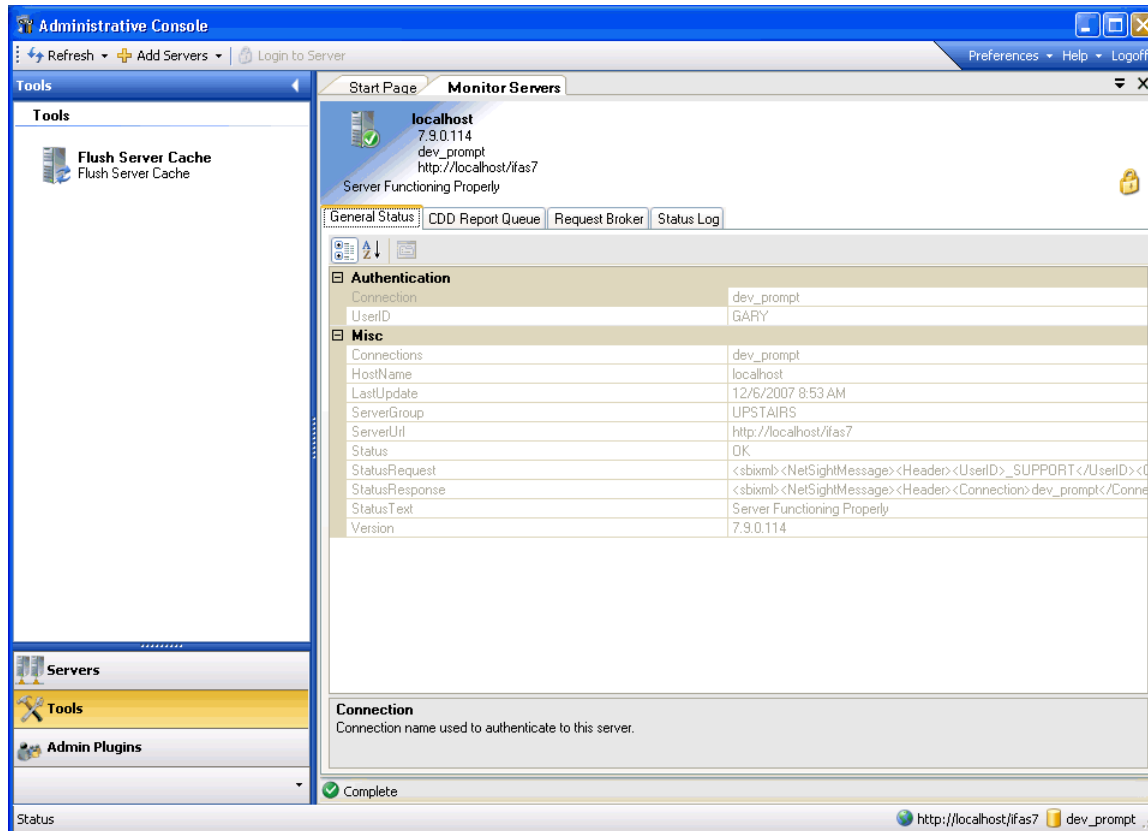
## 3.5.4 Status Log

The plugin maintains a history of the server's monitoring processes in the "Status Log" tab. This allows the administrator to look back at the history of this server's status checks.



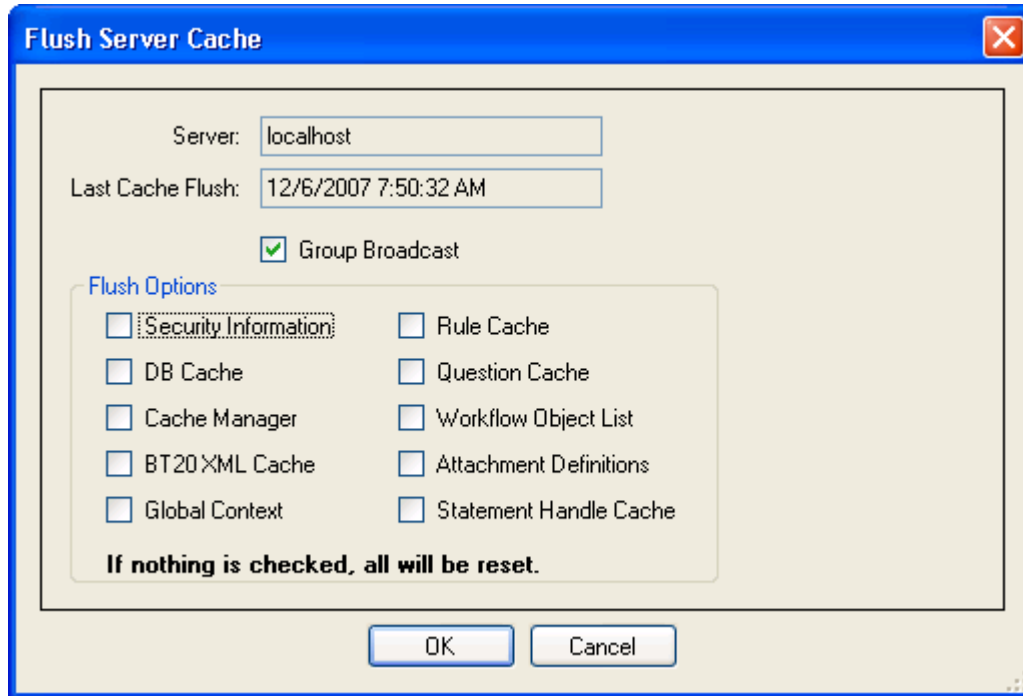
## Flush Server Cache

For performance reasons the 7i Servers cache a lot of information. However, updates may take place during the normal business hours that require the servers to release the cached information and fetch the updated versions. A tool on the tools panel of this plugin allows the user to execute a Flush Cache request to the server.



Option	Description
Group Broadcast	If checked the server will also attempt to request a flush from any other servers accessible within the server group.
Security Information	Clears any security information from the 7i Server cache.
Rule Cache	Clears cached business rule information from the 7i Server.

DB Cache	Clears cached database information from the 7i server.
Question Cache	Clears the cached Nucleus question information from the 7i server.
Cache Manager	A common caching mechanism throughout the software. This option will inform it that all of its data needs to be cleared.
Workflow Object List	Clears the list of workflow objects used by the 7i Server.
BT20 XML Cache	Clears out cached BT20 xml from the server.
Attachment Definitions	Clears the Documents Online attachment definitions from the server.
Global Context	A common caching mechanism across threads. This option will inform it that all of its cached data must be cleared.
Statement Handle Cache	Used to speed query time this cache stores prepared statements. This option will clear all cached statement handles.



Once the tool is executed a status message will appear at the top of the plugin.



The screenshot shows the Administrative Console interface. The main window is titled "Monitor Servers" and displays the status of the "localhost" server. The server is identified as "localhost" with version "7.9.0.114" and is using the "dev\_prompt" connection. The status is "Server Functioning Properly".

The "Misc" section contains the following information:

Connections	dev_prompt
HostName	localhost
LastUpdate	12/6/2007 8:54 AM
ServerGroup	UPSTAIRS
ServerUrl	http://localhost/ifs7
Status	OK
StatusRequest	<sbixml><NetSightMessage><Header><UserID>_SUPPORT</UserID></Header></sbixml>
StatusResponse	<sbixml><NetSightMessage><Header><Connection>dev_prompt</Connection></Header></sbixml>
StatusText	Server Functioning Properly
Version	7.9.0.114

The "Connection" section indicates the connection name used to authenticate to this server. The status bar at the bottom shows "Complete" and the URL "http://localhost/ifs7" with the user "dev\_prompt".

## 3.6 Database Logging Worksheet

### 3.6.1 Database Logging

#### Overview

The Database Logging Worksheet allows the user to setup logging of columns in any table in the database. Entries and changes made in the Worksheet are not active until database triggers are recreated at a later time – usually during system downtime. So, changes can be made in the worksheet without affecting the logging status. When you want to activate the changes, preferably during downtime, the xschema process is

run on the application server for the subsystem to be changed. This process will create the commands needed to establish database triggers that will log any changes to the table columns selected for logging.

Launch this plugin by selecting Database Logging from the Administrative Console, as shown in Figure 1.



**Figure 1. Launching the Database Logging plugin**

This will launch the screen show below in Figure 2:

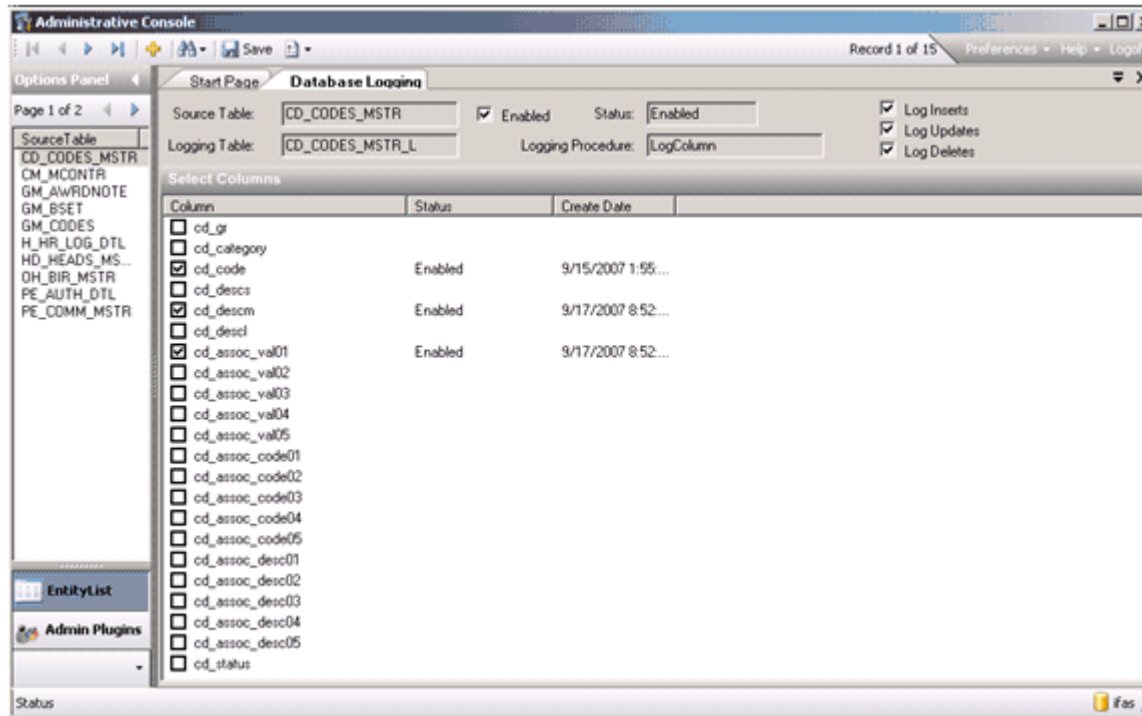
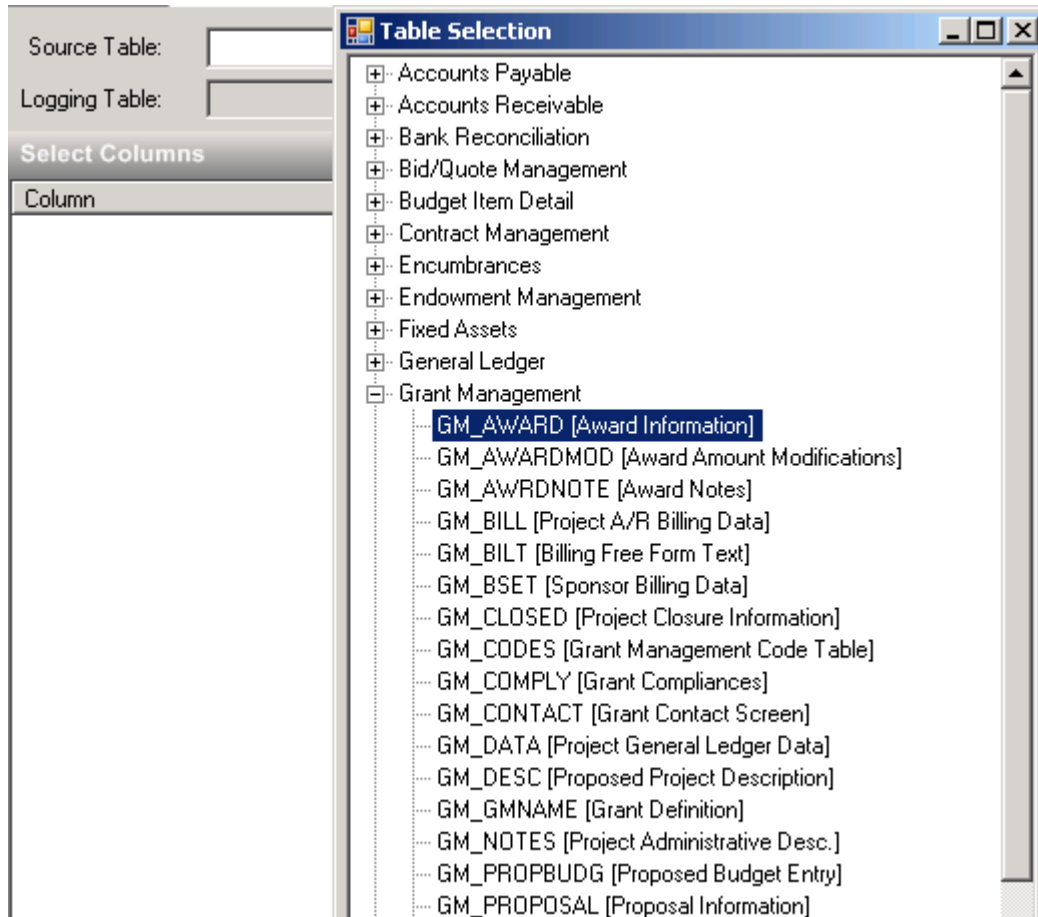


Figure 2. The Database Logging Worksheet.

## Navigation

The Entity List panel on the Left of the screen allows you to select from any tables already configured for logging. You can select one of those tables to change logging settings, or add a new table by going into Add Mode, and selecting a table from the Source Table lookup, as shown in figure 3:



**Figure 3: Selecting a new table to be logged**

The selection lookup is organized with tables inside of subsystems.

The Database Logging worksheet has 2 main panels. The Header panel contains the Table Level Fields, and the column selection panel contains the column level list.

### Header Panel Field Descriptions

**Figure 4: Header section**

**Source Table:** Source Table is the table to be logged.

**Enabled:** The Enabled checkbox allows the user to enable or disable logging for the Source Table.

**Status:** The Status field shows the current logging status of this table. The possible values are Enabled, Disabled, Changes Pending, and Disable Pending. The Changes and Disable Pending indicate that the user has made a change, but that the triggers have not been rebuilt to make those changes effective yet.

**Log Inserts/Updates/Deletes:** These checkboxes indicate which database actions you wish to log. By default, Updates and Deletes are logged (since Inserted data can usually be ascertained by looking at the delete/update information in conjunction with the current record itself).

**Logging Table:** The Logging Table field displays the name of the table that holds the logged information. By default, it is the original table name with a “\_X” appended to the end of it. The logging table has the following structure:

```
CREATE TABLE sourcetable_x
tx_id CHAR(36),
at_id CHAR(18),
    event_id CHAR(1),
    old_value (150),
    new_value (150),
    rec_unique_key CHAR(36),
```

```

log_who CHAR(16),

log_when,

unique_key CHAR(36)

)

```

The tx\_id, or transaction ID is unique to the insert/update/delete instance, so will be the same for all columns logged for a given action. The at\_id is the column name. The event\_id is I (insert), U (update), or D (delete). The old and new values are the column values before and after the change. The rec\_unique\_key ties this log record back to the original record. The log\_who and log\_when are the user id and time of the change.

**Logging Procedure:** The Logging Procedure field shows the name of the Stored Procedure used to log changes, if any.

**Column Selection Panel Field Descriptions:** Once a Source Table is selected in the Header panel, the lower part of the screen shows the columns in the Source Table, allowing the user to select which columns should be logged:

Column	Status	Create Date
<input type="checkbox"/> cd_gr		
<input type="checkbox"/> cd_category		
<input type="checkbox"/> cd_code	Disable Pending	9/15/2007 1:55:15 PM
<input type="checkbox"/> cd_descs		
<input checked="" type="checkbox"/> cd_descm	Enabled	9/17/2007 8:52:27 AM
<input type="checkbox"/> cd_desci		
<input checked="" type="checkbox"/> cd_assoc_val01	Enabled	9/17/2007 8:52:27 AM
<input type="checkbox"/> cd_assoc_val02		
<input type="checkbox"/> cd_assoc_val03		
<input type="checkbox"/> cd_assoc_val04		
<input type="checkbox"/> cd_assoc_val05		
<input checked="" type="checkbox"/> cd_assoc_code01	Enable Pending	
<input type="checkbox"/> cd_assoc_code02		
<input type="checkbox"/> cd_assoc_code03		
<input type="checkbox"/> cd_assoc_code04		
<input type="checkbox"/> cd_assoc_code05		
<input type="checkbox"/> cd_assoc_desc01		
<input type="checkbox"/> cd_assoc_desc02		
<input type="checkbox"/> cd_assoc_desc03		
<input type="checkbox"/> cd_assoc_desc04		
<input type="checkbox"/> cd_assoc_desc05		
<input type="checkbox"/> cd_status		

**Figure 5: Column Selection panel**

**Column:** The column checkbox can be checked for any column to be logged.

**Status:** The Status shows the current logging status of the column. Possible values are blank, Enabled, Enable Pending, and Disable Pending. Enable pending means it has been selected for logging, but is not currently logged. Disable pending means it is currently being logged, but is marked for disabling. When the xschema/xsql process is subsequently implemented to actually make all changes active, the pending status will be changed.

**Create Date:** The date and time when the last change was made.

**Activating Logging:** IFAS Database Logging is accomplished using database triggers. A pair of application server programs, xschema and xsql, are used to read the information entered into the Database Logging Worksheet, and create the database triggers needed to activate the logging function.

Whenever anything needs to be done to the structure of the IFAS Database, including changes to database triggers, it is normally done for a given subsystem using the xschema/xsql processes. This usually occurs during the Update Services process. In fact, the Update Services process will activate any “pending” changes for logging.

There are two options for activating logging (besides waiting for an Update Services): (1) by subsystem, and (2) globally for all subsystems. To illustrate, suppose you have just added a Purchasing table in the logging worksheet, and have some Enabled Pending columns. To activate, you would log onto your application server using a terminal connection, and an administrative user id, like ‘bsi’ or ‘sbi’. The following should generally be performed when the system is not in use.

One option would be to rebuild all of the triggers for the Purchasing subsystem, as follows:

```
cd schema
xschema podb
xsql podb.trig
```

Briefly, the above steps make your current directory the app/schema directory, where the schema files exist. Then, the “xschema podb” takes the podb schema file as input, and outputs a number of SQL files, one of which is the podb.trig file. This file contains the SQL commands to drop and recreate all triggers for the Purchasing tables. The “xsql podb.trig” connects to the database and actually runs the SQL against the database.

The other option is to rebuild only the triggers for tables that have logging defined and active for them, regardless of subsystem. The steps for that option are:

```
cd schema
```

```
xschema -logging [table name]
xsql logging.sql
```

In this case, the “-logging” option of xschema generates only the logging.sql file, which contains the SQL needed to activate any pending logging triggers. If the table name is included, it only rebuilds triggers for that table. There is also a –logging2 option that will “force” rebuilding of all logging triggers, even those which are already active.

## Viewing Logged Information from 7i

To see the logging history for a given record from 7i, browse to the record of interest, and select the Record Information tool.

Associated Numeric Values	Associated Codes	Associated Descriptions
		WARN
	P	
		BLOCK

**Figure 6: Record Info for a common code in the NUUPCD 7i screen**



Since we are logging changes on the common code table (cd\_codes\_mstr), and within that we are logging changes to the medium description (cd\_descm), we will see any changes to the medium description in the Log Data tab of the Record Information dialog:



The screenshot shows a web browser window titled "Bi-Tech Record Data -- Web Page Dialog". Inside, there are two tabs: "Record Data" and "Log Data". The "Log Data" tab is active and displays a table for "BT20.CDCodesMaster". The table has four columns: "Date", "Type", "Who", and "Transaction". The "Transaction" column is further divided into "Attribute", "Old Value", and "New Value". The table shows a single entry for an update performed by SCOTT on 10/9/2007 at 10:41:00 AM. The attribute is "cd\_descm", and the new value is "this is logged". A "Close" button is located at the bottom of the dialog.

Date	Type	Who	Transaction
	Attribute	Old Value	New Value
10/9/2007 10:41:00 AM	UPDATE	SCOTT	E5B7E5C7-071F-4B3A-916C-4B08E76D0C30
	cd_descm		this is logged

**Figure 7: Log Data dialog**

This dialog shows the logging history in a 2 line format, with the most recent changes first.

## 4 Other 7i Administration Tasks

### 4.1 Screen Compile

The screen compile process takes a screen definition xml file and a number of other files and settings and generates an asp file containing the web page usable by a browser. The following is a list of the inputs that the screen compile process uses:

Screen definition xml file found in screens/subsystem/screenname.xml.

The business rules xml documents stored as a resource of a dll used to handle the tables presented on the page.

Any custom business rules defined for the tables presented on the page.

Any custom business rules defined for the specific page.

The VB Script used to determine the field labels (field tags) to be used when the page is rendered.

Default field formatting information and custom field formatting information.

After field rule definitions.

When any event occurs that causes this information to change, a screen compile of affected screens is in order. The screen compile itself is run on the Windows server with the following command, from the ifas7 directory:

```
cscript screencompile.js connection-name [-p ###] [-ssl] [-o output Directory][-v virtual Root] [-f filename] [-s system Name]
```

Examples:

```
cscript screencompile.js prod -v ifas7
```

This command will compile all of the screens for the prod connection, and the name of the virtual root on this machine is ifas7.

```
cscript screencompile.js prod -v ifas7 -s AccountsPayable
```

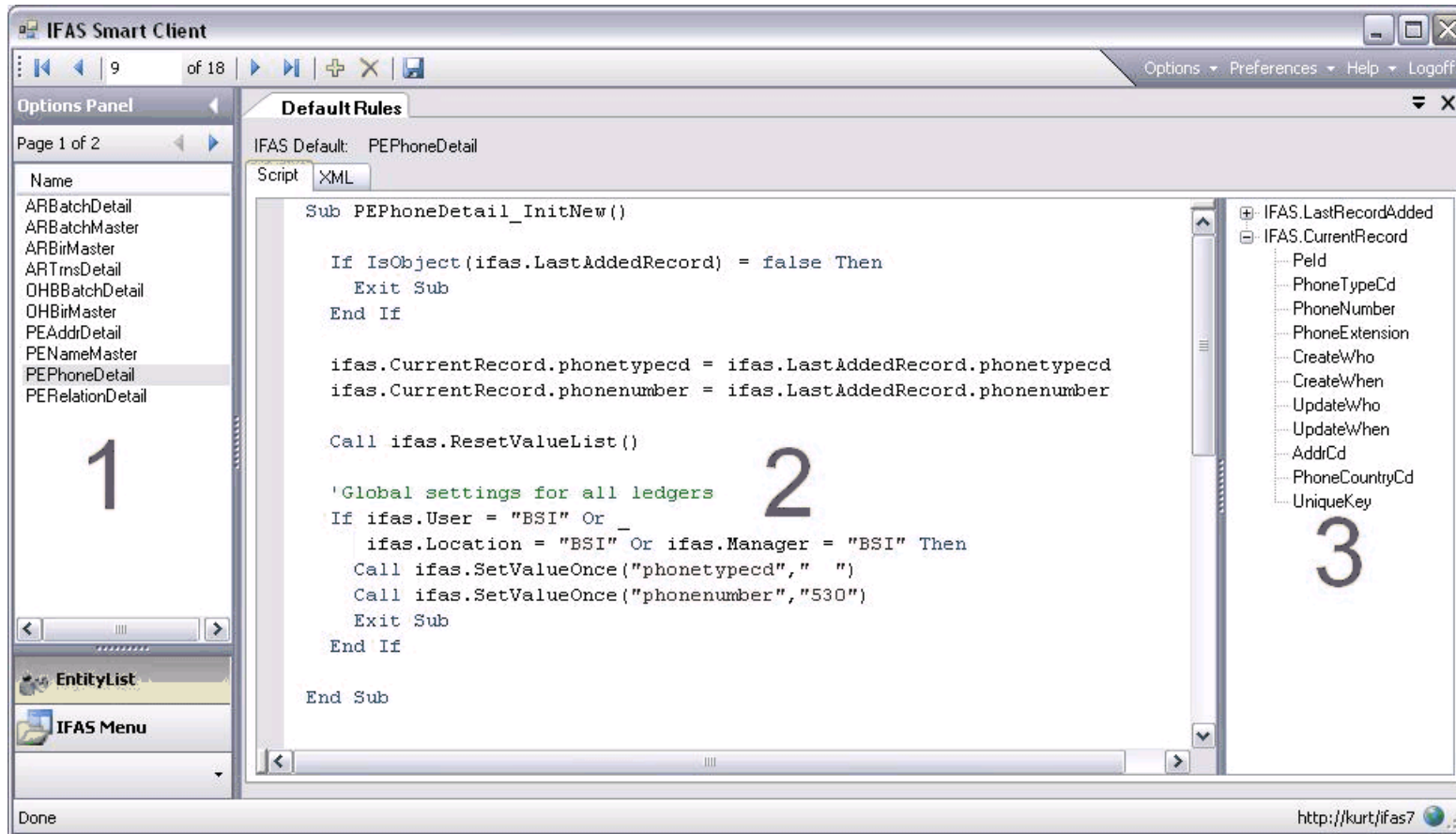
This command will compile all of the Accounts Payable screens for the prod connection, and the name of the virtual root on this machine is ifas7.

```
cscript screencompile.js prod -f PersonEntity\PEUPPE.xml
```

## 4.2 Defaults Rules (NUUPDF)

The System Default Definition screen is used to define certain data elements that could default for users during data entry. Logic can be used to default in values for new records, or updated records based on what has been already entered. Each Default consists of 2 parts: VBScript and XML. The VBScript is used to create subroutines that will be called by 7i when business rules are fired on a particular BT20 (table). The XML is used to define which routine will be executed for each kind of event that can happen on a BT20.

Below is a sample screenshot of the NUUPDF screen, which runs in the IFAS Smart Client only and not as a thin 7i web screen.



The screen is made up of 3 parts:

**Entity List** for navigating existing defaults. This is a list of each BT20's default that has been created. New ones can be added by clicking the '+' button in the navigation bar.

**VBScript/XML Editor.** This is where you can edit the default rules that will be executed for the selected BT20.

**BT20 Tree View.** This view will allow you to see the attributes for the selected BT20 that can be used in the VBScript/XML editors.

## 4.2.1 Entity List

The Entity List allows navigation of existing records and the ability to delete and create new ones. The navigation bar at the top of the screen can be used to choose records or simply click the one you wish to view in the entity list with the mouse. When you are done editing a default (or creating a new default), click the save button to save your changes to the database. Both the XML and VBScript are stored in the database in a BLOB to be accessed by the 7i server. When the save is complete, you will receive a 'Record Accepted' message. At this time, the screen has also notified the 7i server to reload the defaults it has loaded so your new changes are ready to take effect. If you (or any other users) have any open 7i screens simply exit them and open new ones for the new defaults to take effect.

**CAUTION:** If you delete a record, it is permanently destroyed and the defaults will have to be recreated.

The screenshot displays the IFAS Smart Client interface. The main window is titled "IFAS Smart Client" and shows a list of entities on the left and a configuration panel on the right. The "Default Rules" tab is active, showing the script for the "PEPHONEDETAIL" entity. The script is written in VBA and includes logic for initializing new records and setting global settings for all ledgers based on the user and location.

**Options Panel**  
Page 1 of 2

**EntityList**

- ARBatchDetail
- ARBatchMaster
- ARBirMaster
- ARTmsDetail
- DHBBatchDetail
- DHBirMaster
- PEAddrDetail
- PENameMaster
- PEPhoneDetail**
- PERelationDetail

**Default Rules**  
IFAS Default: PEPHONEDETAIL

**Script** **XML**

```
Sub PEPHONEDETAIL_InitNew()  
  
    If IsObject(ifas.LastAddedRecord)  
        Exit Sub  
    End If  
  
    ifas.CurrentRecord.phonetypecd = i  
    ifas.CurrentRecord.phonenumber = i  
  
    Call ifas.ResetValueList()  
  
    'Global settings for all ledgers  
    If ifas.User = "BSI" Or _  
        ifas.Location = "BSI" Or ifas.L  
        Call ifas.SetValueOnce("phonety  
        Call ifas.SetValueOnce("phonenum  
        Exit Sub  
    End If  
  
End Sub
```

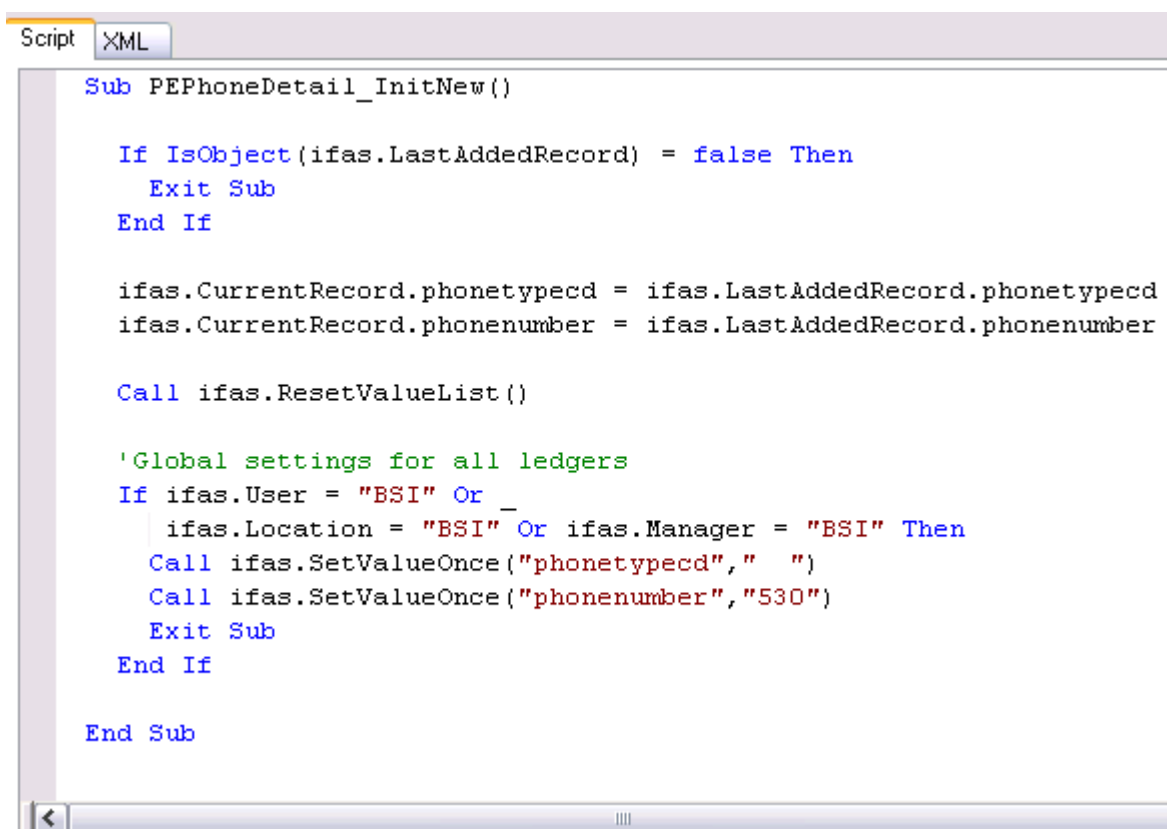
Done

## 4.2.2 VBScript Editor

There are two separate tabs, one for editing the VBScript that will be run when a business rule in 7i is fired, and the other to tell 7i which routine to execute on a business rule.

The IFAS defaults are written with the VBScript language. The editor will highlight key words for you to assist in writing your defaults. Once you make a change to the script and save the record, the changes will immediately take effect in 7i. We recommend that you first test your script changes in a test account to avoid encountering problems in your live environment when developing your defaults.

Below is a sample of an InitNew routine that will be setup to be called when a new record is created when you go into add mode in a 7i screen.

The image shows a screenshot of a web-based VBScript editor. At the top, there are two tabs: 'Script' (which is active) and 'XML'. The main area contains a VBScript routine named 'PEPHONEDETAIL\_InitNew'. The script logic is as follows: it first checks if 'ifas.LastAddedRecord' is an object. If not, it exits the sub. If it is, it copies the 'phonetypecd' and 'phonenumbr' values from the last added record to the current record. It then calls 'ifas.ResetValueList()'. A comment indicates 'Global settings for all ledgers'. Another conditional check follows: if the user is 'BSI' or the location is 'BSI' or the manager is 'BSI', it sets 'phonetypecd' to a space and 'phonenumbr' to '530' using 'Set Value Once' calls. Finally, it exits the sub.

```
Script XML
Sub PEPHONEDETAIL_InitNew()

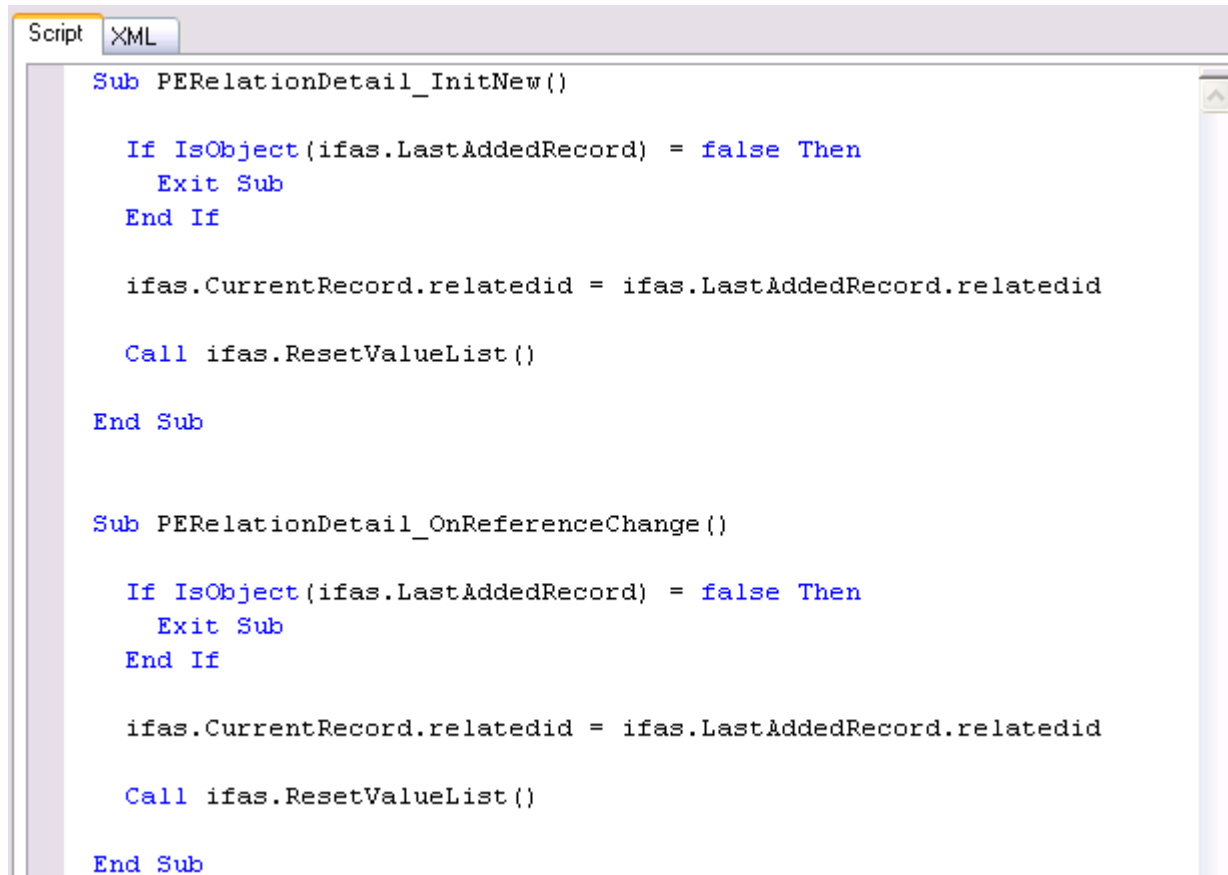
    If IsObject(ifas.LastAddedRecord) = false Then
        Exit Sub
    End If

    ifas.CurrentRecord.phonetypecd = ifas.LastAddedRecord.phonetypecd
    ifas.CurrentRecord.phonenumbr = ifas.LastAddedRecord.phonenumbr

    Call ifas.ResetValueList()

    'Global settings for all ledgers
    If ifas.User = "BSI" Or _
        ifas.Location = "BSI" Or ifas.Manager = "BSI" Then
        Call ifas.SetValueOnce("phonetypecd", " ")
        Call ifas.SetValueOnce("phonenumbr", "530")
    End If
End Sub
```

More than one routine can be specified in the script editor. Simply separate them by a blank line:

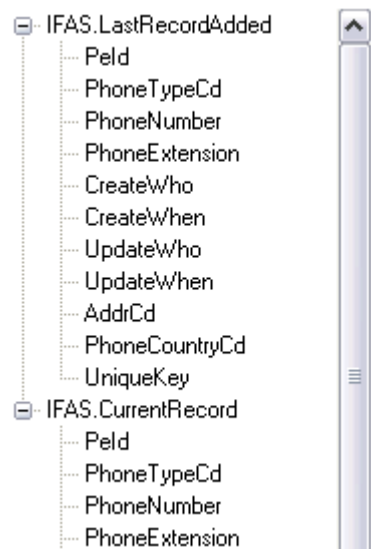
A screenshot of a script editor window. The window has a title bar with 'Script' and 'XML' tabs. The main area contains two subroutines, each starting with 'Sub' and ending with 'End Sub', separated by a blank line. The code is as follows:

```
Sub PERelationDetail_InitNew()  
  
    If IsObject(ivas.LastAddedRecord) = false Then  
        Exit Sub  
    End If  
  
    ivas.CurrentRecord.relatedid = ivas.LastAddedRecord.relatedid  
  
    Call ivas.ResetValueList()  
  
End Sub  
  
Sub PERelationDetail_OnReferenceChange()  
  
    If IsObject(ivas.LastAddedRecord) = false Then  
        Exit Sub  
    End If  
  
    ivas.CurrentRecord.relatedid = ivas.LastAddedRecord.relatedid  
  
    Call ivas.ResetValueList()  
  
End Sub
```

There are some built in variables and functions that can be used within your scripts. Below is a list:



**ifas.CurrentRecord.** – The data structure that holds the current record on the screen. Each column of the BT20 is defined on the data structure. See the screen shot below for an example on the PEPHONEDETAIL.



**ifas.LastAddedRecord** – The same data structure as above, but it contains the last record that was added through the 7i screen.

**ifas.SetValueOnce(property, value)** – Sets the named 'property' on the BT20 to provided 'value'. If a second call to ifas.SetValueOnce is made within the same script, it will not update the value. Calling ifas.ResetValueList() will allow additional changes to be saved.

**ifas.ResetValueList()** – Clears the set property list so that ifas.SetValueOnce() can set new values.

**ifas.User** – Retrieves the user id of the current user in the 7i screen.

**ifas.UserName** - Retrieves the Users' Name off of the us\_usno\_mstr.

**ifas.UserDesc** - Retrieves the Users' Description off of the us\_usno\_mstr.

**ifas.Manager** - Retrieves the Users' Manager off of the us\_usno\_mstr.

**ifas.Location** – Retrieves the Users' Location code off of the us\_usno\_mstr.

**ifas.CurrentScreenMask** - Retrieves mask of the current 7i screen.

**ifas.ClientId** - Retrieves the Client Id from the SYID CLIENTID common code.

**ifas.GetBackgroundPart(part, ledger, key, obj)** – Retrieves the key or object background part for the requested ledger.

**ifas.AddMode** – Returns 1 if the screen is in add mode, set to 0 otherwise.

**ifas.IFASVersion** – Retrieves the current IFAS Version that the 7i server is running.

**ifas.CurrentLedger.GILedger** – Retrieves the current user's GILedger.

**ifas.CurrentLedger.JILedger** – Retrieves the current user's JILedger.

**ifas.TranslateText** – Retrieves translated strings for known tokens such as “\CD8” will return the date in YYYYMMDD format. Below is a list of acceptable values.

Ifas.TranslateText changes all \xxxx', where:

```

\Xxxx    = 'G' GL side or 'J' JL side
\xXxx    = 'K' key part, 'O' object part
          'L' Ledger, 'B' Budget
\xxXx    = 'F' Fund
          'X' Function
'O'      Budget officer
          'D' Description    (S,M,L valid)
          '1-8' Array part   (S,M,L valid)
'A'      Account Type
          'B' Budget Category
          '9-0' Array part   (only valid for xBXx)

```

'W' Working Budget (S,M,L valid)

\xxxX = 'S', 'M', 'L' short, medium, or long

\xTn = GL Account title from GEN master with  
n = 'B' meaning both desc's.

## Other options

\xSnn = Subsystem ID (2 char.); n=01 through 10 for  
the 10 subsystems defined in GLG-GEN-MSTR

\xPnn = Period name; n=01 through 14

\xDyxxx = Symbolic Date conversion  
y = Date Output Format (eg S-Z,6,8)  
xxx = Symbolic Date (eg FYB,FYE,...)

\xMnn = Misc. Code desc.; n=01 through 08

## Other non-ledger/ar specific

\USER = Upshifted USER ID

\CDx = Current date

Currently supported values of 'x' date format:

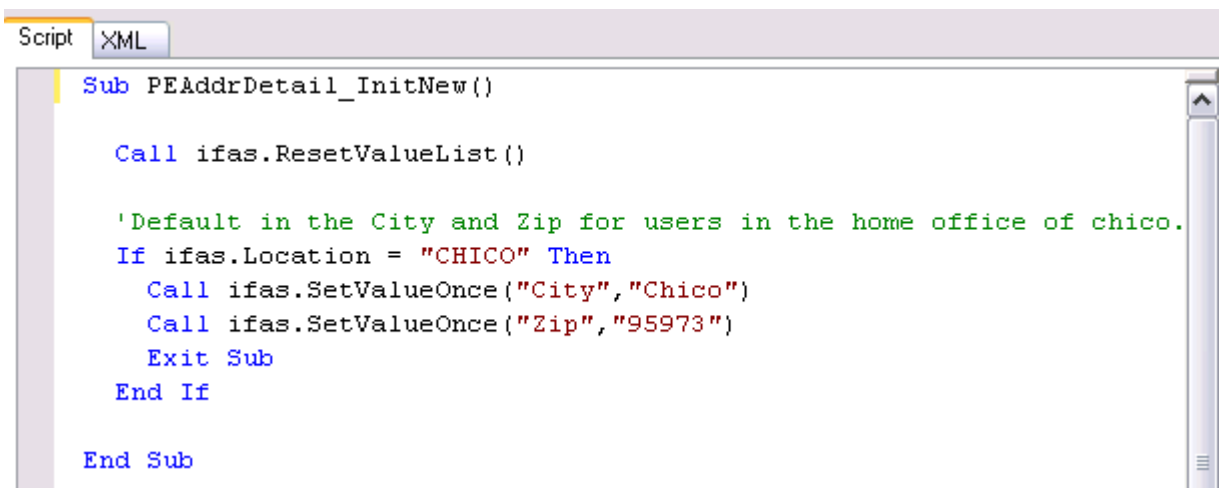
T = MM/DD/YYYY

8 = YYYYMMDD

M = HHMMSS (time)

## Script examples

**Setting a field value if a condition is true.** The below script is a sample for the PEAddrDetail which is the BT20 for the Address Detail in PE. In the InitNew routine, the script sets the zip code to “95973” and the city to ‘Chico’ if the User’s Location is set to ‘CHICO’. Every time a user with this Location adds a new address record in PEUPPE, the city and zip will default in. Notice we call ifas.ResetValueList() first to ensure that our ifas.SetValueOnce() calls will take effect and save their values.



```
Script XML
Sub PEAddrDetail_InitNew()

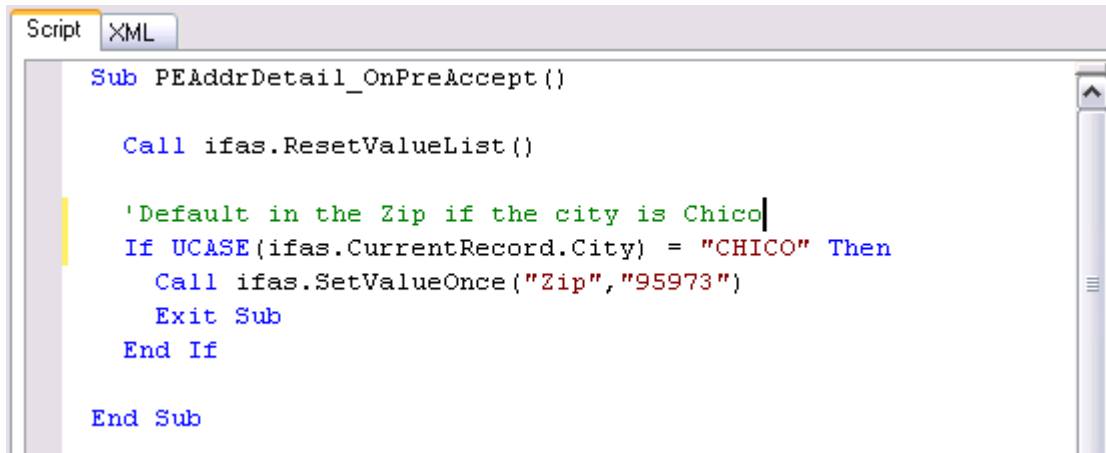
    Call ifas.ResetValueList()

    'Default in the City and Zip for users in the home office of chico.
    If ifas.Location = "CHICO" Then
        Call ifas.SetValueOnce("City", "Chico")
        Call ifas.SetValueOnce("Zip", "95973")
    Exit Sub
    End If

End Sub
```

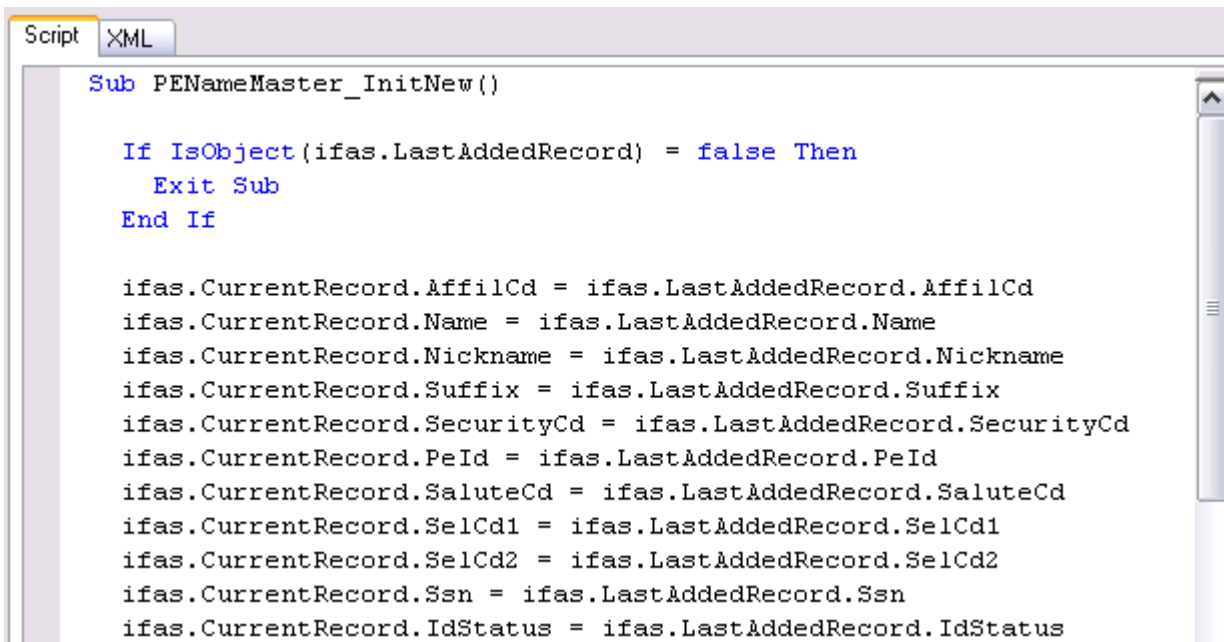
**Setting a field value if a condition based on another field is true.** The below is a sample for the PEAddrDetail’s PreAccept routine (see the XML tab section for more information on PreAccept). Notice we call ifas.ResetValueList() first to ensure that our ifas.SetValueOnce() calls will take effect and save their values.

We then up shift the City on the current record and check to see if it equal to “CHICO”. If it is, we set the zip code to 95973.

A screenshot of a script editor window with a tab labeled 'Script' and 'XML'. The code is as follows:

```
Sub PEAddrDetail_OnPreAccept()  
  
    Call ifas.ResetValueList()  
  
    'Default in the Zip if the city is Chico  
    If UCASE(ifas.CurrentRecord.City) = "CHICO" Then  
        Call ifas.SetValueOnce("Zip", "95973")  
        Exit Sub  
    End If  
  
End Sub
```

**Setting field values from the last added record.** The below script is a sample for the PENAMEMaster which is the BT20 for the Name Master in PE. If a user is going to add more than one record in succession, we can set fields on the new record to what was previously entered to help speed up data entry. In the InitNew routine, the script first checks to see the user has previously added a record in the screen. Then the various fields on the record are set on the CurrentRecord from the LastAddedRecord.



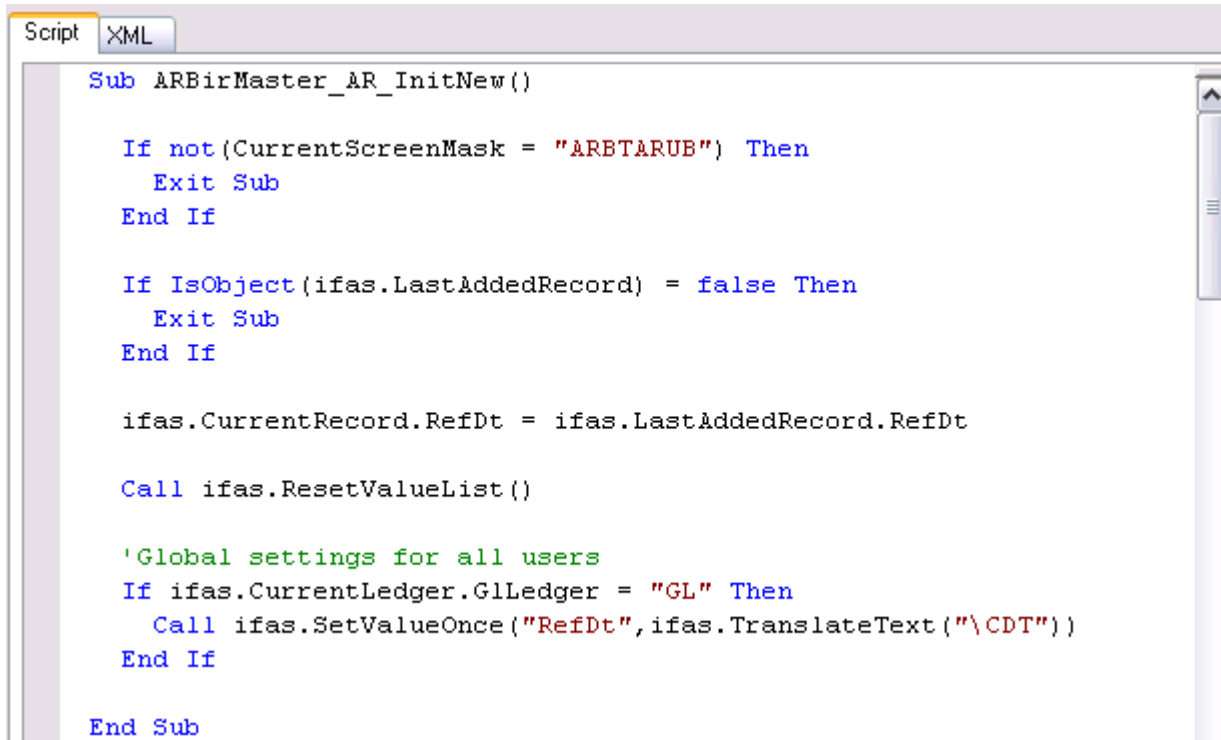
```
Script XML
Sub PENAMEMaster_InitNew()

    If IsObject(Ifas.LastAddedRecord) = false Then
        Exit Sub
    End If

    Ifas.CurrentRecord.AffilCd = Ifas.LastAddedRecord.AffilCd
    Ifas.CurrentRecord.Name = Ifas.LastAddedRecord.Name
    Ifas.CurrentRecord.Nickname = Ifas.LastAddedRecord.Nickname
    Ifas.CurrentRecord.Suffix = Ifas.LastAddedRecord.Suffix
    Ifas.CurrentRecord.SecurityCd = Ifas.LastAddedRecord.SecurityCd
    Ifas.CurrentRecord.PeId = Ifas.LastAddedRecord.PeId
    Ifas.CurrentRecord.SaluteCd = Ifas.LastAddedRecord.SaluteCd
    Ifas.CurrentRecord.SelCd1 = Ifas.LastAddedRecord.SelCd1
    Ifas.CurrentRecord.SelCd2 = Ifas.LastAddedRecord.SelCd2
    Ifas.CurrentRecord.Ssn = Ifas.LastAddedRecord.Ssn
    Ifas.CurrentRecord.IdStatus = Ifas.LastAddedRecord.IdStatus

End Sub
```

**ADVANCED: Setting values with TranslateText.** The below script is a sample for the ARBirMaster which is the BT20 for the AR Batches in ARBTARUB. This script first checks the current screen to be ARBTARUB. If it is not that screen, the script exists. At the end, it checks the GILedger code to be "GL". If it is, it sets the "RefDt" equal to the current date in "MM/DD/YYYY" format with ifas.TranslateText("\CDT").



```
Script XML
Sub ARBirMaster_AR_InitNew()

    If not (CurrentScreenMask = "ARBTARUB") Then
        Exit Sub
    End If

    If IsObject(ifas.LastAddedRecord) = false Then
        Exit Sub
    End If

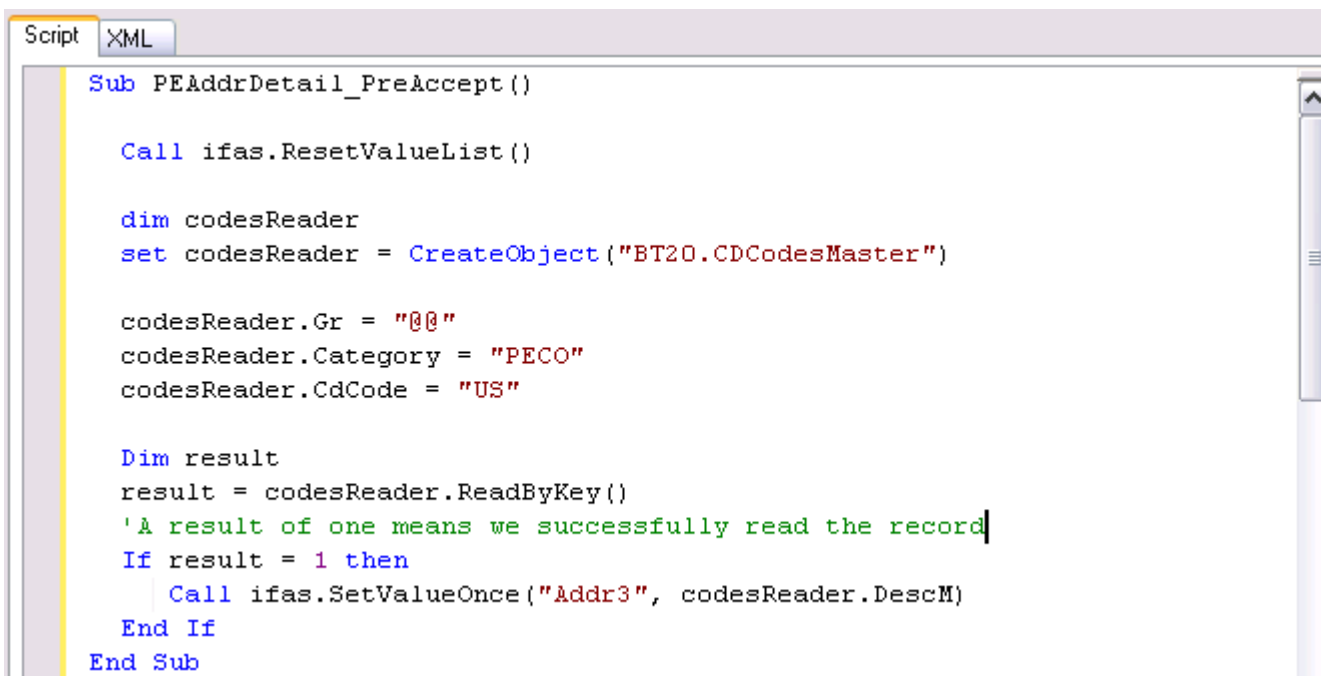
    ifas.CurrentRecord.RefDt = ifas.LastAddedRecord.RefDt

    Call ifas.ResetValueList()

    'Global settings for all users
    If ifas.CurrentLedger.GILedger = "GL" Then
        Call ifas.SetValueOnce("RefDt", ifas.TranslateText("\CDT"))
    End If

End Sub
```

**ADVANCED: Reading common codes.** This is an example of reading a common code from a default script and setting the “Addr3” (Address line 3) field equal to the Medium description of the common code. When reading common codes, you must specify all 3 columns, the Gr, Category, and CdCode or else the common code will not be fetched. A returned result of 1 from ReadByKey on a BT20 object means we successfully read the record we were searching for.

A screenshot of a script editor window with a tab labeled 'Script' and 'XML'. The window contains a VBA script for a subprocedure named 'PEAddrDetail\_PreAccept'. The script performs the following actions: it calls 'ifas.ResetValueList()', declares a 'codesReader' object, sets it to 'CreateObject("BT20.CDCodesMaster")', and assigns values to 'codesReader.Gr' (set to "@@"), 'codesReader.Category' (set to "PECO"), and 'codesReader.CdCode' (set to "US"). It then declares a 'result' variable, calls 'codesReader.ReadByKey()', and includes a comment: 'A result of one means we successfully read the record'. An 'If result = 1 then' block contains a call to 'ifas.SetValueOnce("Addr3", codesReader.DescM)'. The script ends with 'End If' and 'End Sub'.

```
Script XML
Sub PEAddrDetail_PreAccept()

    Call ifas.ResetValueList()

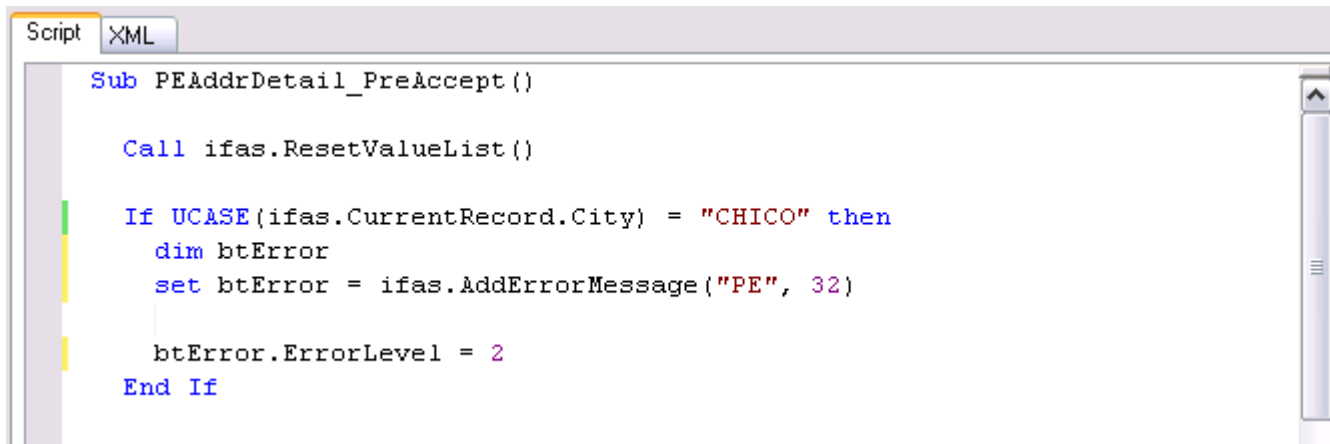
    dim codesReader
    set codesReader = CreateObject("BT20.CDCodesMaster")

    codesReader.Gr = "@@"
    codesReader.Category = "PECO"
    codesReader.CdCode = "US"

    Dim result
    result = codesReader.ReadByKey()
    'A result of one means we successfully read the record
    If result = 1 then
        Call ifas.SetValueOnce("Addr3", codesReader.DescM)
    End If
End Sub
```

**ADVANCED: Issuing error messages.** This is an example of how to issue an error message from a default. Please refer to the “7i System Documentation” on how to create your own custom errorcat with your error messages. The ErrorLevel can be set to a 1 for an error, 2 for a warning and 4 for informational. The below example is for a warning. The replaceable fields with the ‘~’ are not implemented from default scripts at this time. You may only issue static error messages.





```
Script XML
Sub PEAddrDetail_PreAccept()

    Call ifas.ResetValueList()

    If UCASE(ifas.CurrentRecord.City) = "CHICO" then
        dim btError
        set btError = ifas.AddErrorMessage("PE", 32)

        btError.ErrorLevel = 2
    End If
```

### 4.2.3 XML Editor

The XML editor is used to define which VBScript business rules will be fired when certain events happen on a BT20. Below is an example of the PE Address Detail with a few routines defined. The XML needs to be defined in this specific way with the <XML> and <BUSINESSRULES> nodes. Remember, XML is case sensitive.

CAUTION: Creating an error with the XML definitions will prevent 7i from saving records for that BT20.

```

<XML>
  <BUSINESSRULES>
    <BT20.PEAddrDetail.1>
      <INITNEU>
        <RULEOBJECT SCRIPTLOCATION="Defaults" SCRIPTNAME="PEAddrDetail">
          <METHOD ID="PEAddrDetail_InitNew" />
        </RULEOBJECT>
      </INITNEU>
      <PREACCEPT>
        <RULEOBJECT SCRIPTLOCATION="Defaults" SCRIPTNAME="PEAddrDetail">
          <METHOD ID="PEAddrDetail_PreAccept" />
        </RULEOBJECT>
      </PREACCEPT>
      <AFTERFIELD>
        <RULEOBJECT SCRIPTLOCATION="Defaults" SCRIPTNAME="PEAddrDetail">
          <METHOD ID="PEAddrDetail_OnChangePEID_GLOBAL_001">
            <BT20OBJ>
              <RETPROP>PeId</RETPROP>
            </BT20OBJ>
            <TRIGPROP>PeId</TRIGPROP>
          </METHOD>
        </RULEOBJECT>
      </AFTERFIELD>
      <CHANGEREf>
        <RULEOBJECT SCRIPTLOCATION="Defaults" SCRIPTNAME="PEAddrDetail">
          <METHOD ID="PEAddrDetail_OnReferenceChange" />
        </RULEOBJECT>
      </CHANGEREf>
    </BT20.PEAddrDetail.1>
  </BUSINESSRULES>
</XML>

```

## XML Syntax

Below is sample XML with all of the available nodes defined that can be set to perform different actions on the BT20.

```

<XML>

  <SORTS></SORTS>

  <FILTERS></FILTERS>

  <COLDATA></COLDATA>

```

```
<BUSINESSRULES>

  <BT20. PEAddrDetail.1>

    <INITNEW></INITNEW>

    <TAGNAME></TAGNAME>

    <AFTERFIELD></AFTERFIELD>

    <PREACCEPT></PREACCEPT>

    <PREINSERT></PREINSERT>

    <POSTINSERT></POSTINSERT>

    <PREUPDATE></PREUPDATE>

    <POSTUPDATE></POSTUPDATE>

    <PREDELETE></PREDELETE>

    <SetControlProperties></SetControlProperties>

    <TOOLS></TOOLS>

  </BT20. PEAddrDetail.1>

</BUSINESSRULES>

</XML>
```

## Node definitions

<XML>

Required starting XML tag.

<SORTS>

Used to make client defined sorts. Beware that adding a sort on columns that are not indexed will cause a big hit in performance.

Here is a sample to sort the addresses by City:

```
<SORTS>
  <INDEX desc="By City">
    <PROP>City</PROP>
  </INDEX>
</SORTS>
```

<FILTERS>

Not for use at this time.

<COLDATA>

Used to specify which columns are required. Below is a sample to make City required on the PEAddrDetail:

```
<COLDATA>
  <City><REQUIRED/></City>
</COLDATA>
<BUSINESSRULES>
```

Used for executing business rules on a BT20 object when certain events occur on that object. The node directly under <BUSINESSRULES> must be the exact BT20 name that the default has been created for with a .1 on the end of it.

Example for PEAddrDetail:

```
<BUSINESSRULES>
  <BT20. PEAddrDetail.1>
```

...

```
</BT20.PEAddrDetail.1>
```

```
</BUSINESSRULES>
```

Each triggering event node will have a similar format with a RULEOBJECT node and a METHOD NODE. In all cases, the SCRIPTLOCATION attribute will need to be set to "Defaults". The SCRIPTNAME attribute will need to be the same name as the Default. The METHOD node's ID attribute will be the name of routine you will create in the VBScript for this default.

Example for PEAddrDetail:

```
<RULEOBJECT SCRIPTLOCATION="Defaults" SCRIPTNAME="PEAddrDetail">
  <METHOD ID="PEAddrDetail_InitNew" />
</RULEOBJECT>
```

Each node under the BT20 node is a triggering event on the BT20, which can fire a business rule from the VBScript that is created in the other VBScript tab. Below is each type of triggering event and when it gets fired.

```
<INITNEW>
```

This will execute a VBScript routine when in add mode and a new record is initialized. The default 7i INITNEW will fire first, then, your custom one defined here can set any changes after.

```
<INITNEW>
  <RULEOBJECT SCRIPTLOCATION="Defaults" SCRIPTNAME="PEAddrDetail">
    <METHOD ID="PEAddrDetail_InitNew" />
  </RULEOBJECT>
</INITNEW>
```

```
<TAGNAME>
```

This will execute a VBScript routine when the tags on the screen are created. You can set the tag names from your script. A screencompile will need to be performed for the tags on the screen to take effect. They are created by the screencompile routine. Note: If 7i is already setting the tagname, setting it in your own custom script will not override it. Setting it should only be done on additional fields you have added.

```
<TAGNAME>
  <RULEOBJECT SCRIPTLOCATION="Defaults" SCRIPTNAME="PEAddrDetail">
    <METHOD ID="PEAddrDetail_TagName" />
  </RULEOBJECT>
</TAGNAME>
```

Script sample:

```
Sub PEAddrDetail_TagName ()

    Call ifas.CurrentRecord.SetTag("ZIP", "Zip Code")

End Sub
```

<AFTERFIELD>

This will execute a VBScript routine when a field on the screen is left and it's value has changed. For example, changing a field from '1' to '2' and tabbing out of that field will trigger this event.

There is additional triggering information on this event. The <BT20OBJ> node is to signify that we are going to pass the BT20 object (CurrentRecord) into the routine. The <RETPROP> node below it specifies that we are going to allow changes to the PeId field and return its changes. The <TRIGPROP> node is used to specify which field will trigger this Afterfield method.

```
<AFTERFIELD>
  <RULEOBJECT SCRIPTLOCATION="Defaults" SCRIPTNAME="PEAddrDetail">
    <METHOD ID="PEAddrDetail_OnChangePEID_GLOBAL_001">
      <BT20OBJ>
        <RETPROP>PeId</RETPROP>
      </BT20OBJ>
      <TRIGPROP>PeId</TRIGPROP>
    </METHOD>
  </RULEOBJECT>
</AFTERFIELD>
```

Script sample:

```
Sub PEAddrDetail_OnChangePEID_GLOBAL_001(Obj)

    Call ifas.ResetValueList()

    'Global settings for all ledgers
    Call ifas.SetValueOnce("Addr1","123 Main St.")

End Sub
```

## <PREACCEPT>

This will execute a VBScript routine when a record has been submitted by the screen to be inserted or updated into the database. It will trigger this business rule prior to the update/insert taking place. If the business rule has an error, the update/insert will not happen.

```
<PREACCEPT>  
  <RULEOBJECT SCRIPTLOCATION="Defaults" SCRIPTNAME="PEAddrDetail">  
    <METHOD ID="PEAddrDetail_PreAccept" />  
  </RULEOBJECT>  
</PREACCEPT>
```

## <PREINSERT>

The same as PREACCEPT, except it is for inserted records only and not updated records.

## <POSTINSERT>

The same as PREINSERT, except it is triggered after a record has been inserted instead of before.

## <PREUPDATE>

The same as PREINSERT, except it is for updated records only and not inserted records.

## <POSTUPDATE>

The same as POSTINSERT, except it is triggered after a record has been updated instead of before.

## <PREDELETE>

The same as PREACCEPT, except it is triggered after a record has been deleted instead of before.

## <SetControlProperties>

Fields on the screen can be given default behavior by setting them in the SetControlProperties node. A screencompile and reopening the screen must be performed for the VBScript changes to take effect.

```
<SetControlProperties>
  <Url ScreenMask="PEUPPE" Qbe="1" Add="1" Update="1" Init="1"
    PreserveCase="false" Length="2" Enabled="false"/>
</SetControlProperties>
```

Below is a list of what can be set on each field. The PENAMEMaster's Url field was used as an example.

ScreenMask – Set this equal to the mask that you want this control property to apply to. Leaving it off applies to all masks.

Qbe – Set this equal to “1” to disable the control in Find mode.

Add – Set this equal to “1” to disable the control in Add mode.

Update – Set this equal to “1” to disable the control in Update mode.

Init – Set this equal to “1” to disable the control in when a record is first initialized.

Length – Set this to limit the length of accepted data in the control.

Enabled – Set this equal to “false” to allow setting the update/add/init/qbe attributes to “1” to disable.

PreserveCase – Set to false to upshift the control's values.

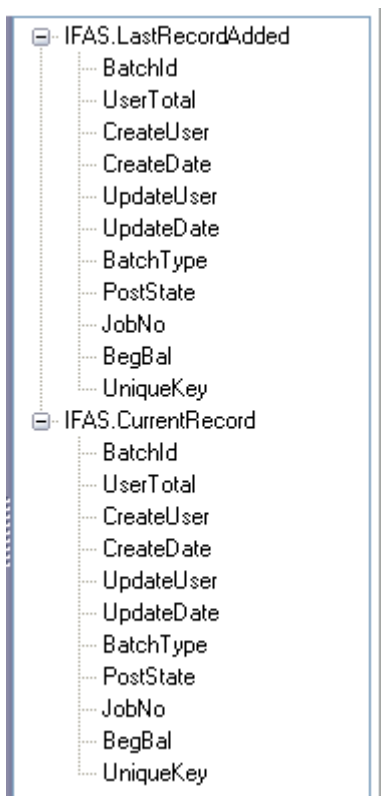
<TOOLS>

Not for use at this time.



## 4.2.4 BT20 Tree View

The tree view displays BT20 information that can be used to help you write your VBScripts and XML definitions for defaults. Any of the BT20 columns can be dragged and dropped into the editor with the mouse. Double clicking an item will insert it where the cursor is positioned.

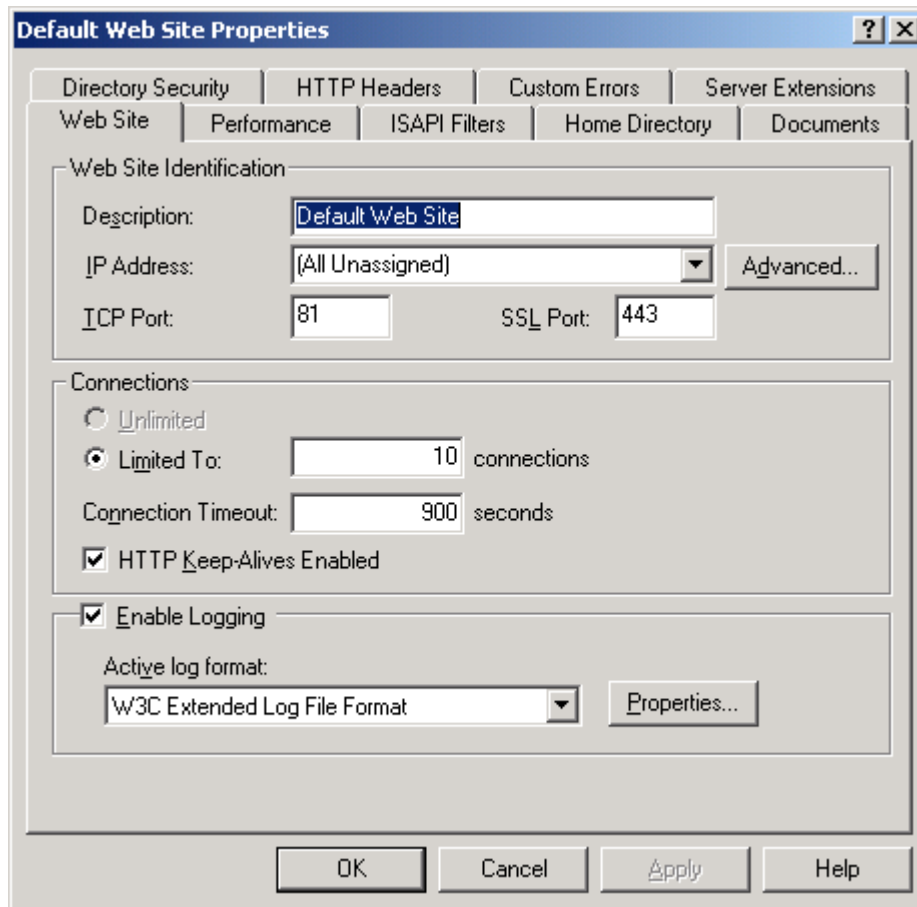


## 5 Network Level Security

This section discusses optional settings for security at the network, IIS, and systems levels. IFAS Application security is not discussed here. For securing parts of the application and database, please see the security guides. The topics in this section include configuration for IIS Port Selection, Secure Sockets Layer (SSL), our BrokerSecurityPolicy.xml file, Firewall considerations, and Windows Integrated Authentication.

## 5.1 IIS Port Selection

The Web servers can be configured to use a port other than the default. The default is port 80. The TCP Port setting below is where the port is configured. Use the Internet Information Services (IIS) Manager to access the Default Web Site Properties.

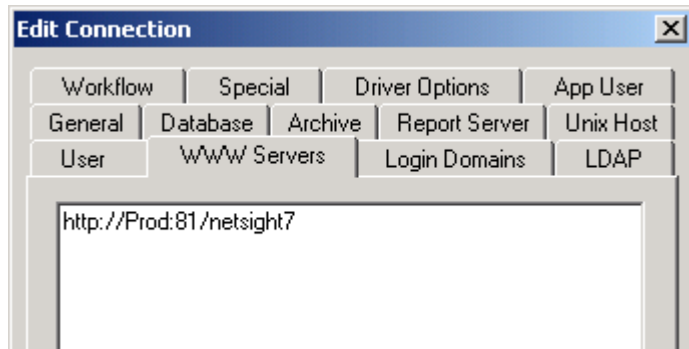


The example above will require that:

port 81 be used to access the web server. ie. `http://<servername>:81/ifas7`.

screen compiles are run with the `-p` switch: `cscript screencompile.js -p 81 ...`

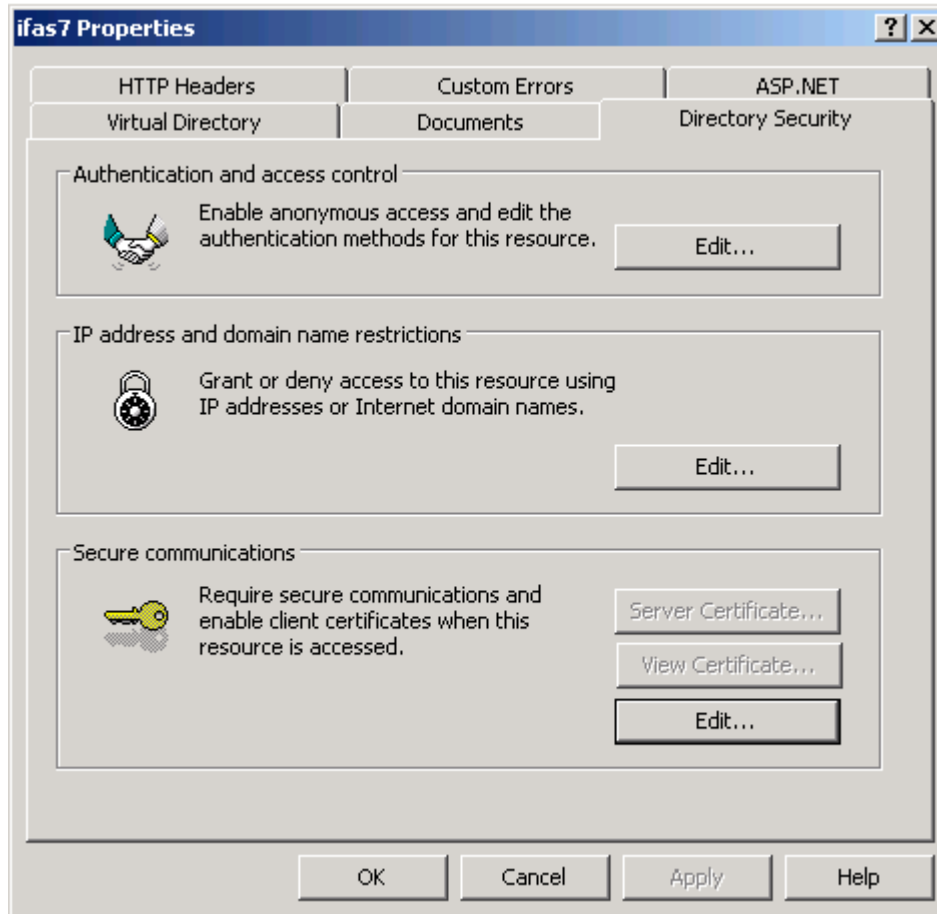
The WWW Servers tab of the Connection Manager includes the port # as shown below.



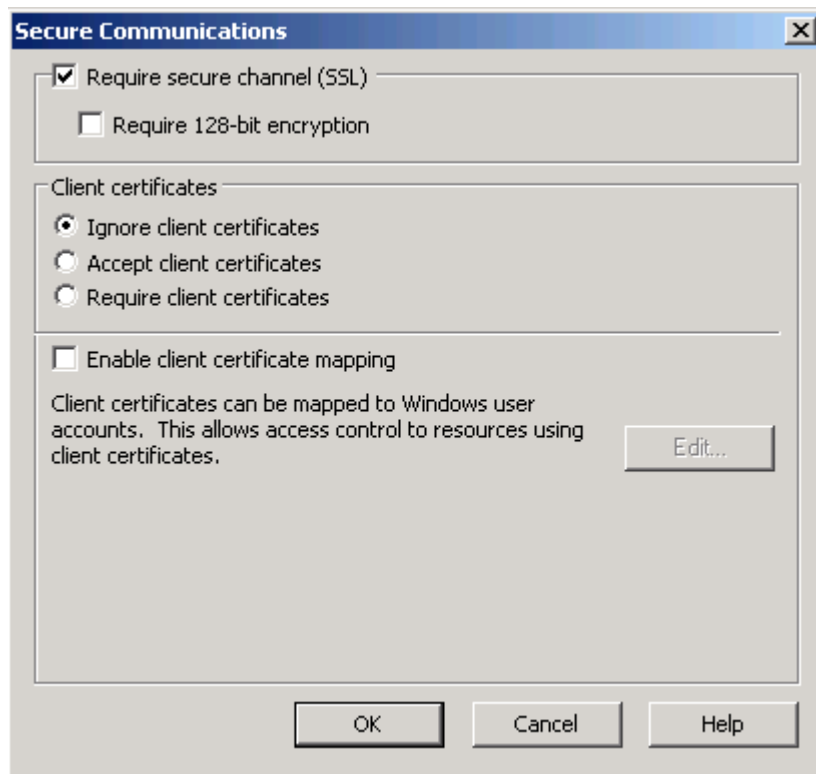
## 5.2 Secure Sockets Layer (SSL)

The Web server can be configured to use Secure Sockets Layer (SSL). SSL is disabled by default. Use the Internet Information Services (IIS) Manager to access the ifas7 Virtual Root under the Default Web Site Properties. The example here will require that SSL be used to access the web server. i.e. `https:// <server name>/ifas7`. It is recommended that SSL be configured for just the ifas7 virtual root instead of the Default Web Site level. Using that approach, other virtual roots are not required to use SSL.

The Directory Security tab of the Web Site Properties looks like the following:

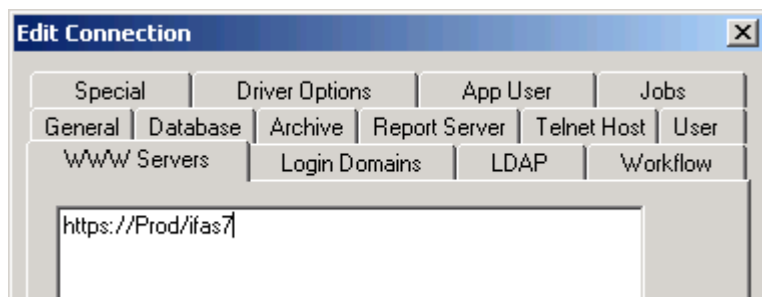


Using Server Certificate, you can setup your certificate. Use the Edit button in the Secure communications section of the Directory Security tab on the Default Web Site Properties dialog to see the following:



The check mark beside Require secure channel (SSL) is what determines if SSL is required for accessing the site.

The WWW Servers tab of the Connection Manager needs to include https:



## 5.2.1 Supporting Nucleus Defaults when SSL is used

When SSL is configured for the virtual root that users will be using to access the 7i services, a separate virtual root will need to be setup without SSL configured, providing access to the 7i services, but allowing access only from the Application server. Edit the IP Address and Domain Name Restrictions found on the Directory Security tab of the virtual root properties in the Internet Information Services Administrator, allowing access only from your internal UNIX or Windows Application Server(s). The Application Servers run a process named map2vb, which uses 7i services to transfer files from the Application Server to the 7i server farm.

## 5.3 Broker Security Policy for 7i

All 7i XML HTTP requests are routed through a security authorization layer. Based on a configuration file called BrokerSecurityPolicy.xml, this layer will examine the source IP, request type, and connection name. Based on that file, requests will either be allowed, denied (and logged to the event log), or it will require login (meaning that if your session has already logged into 7i the request will be accepted, but otherwise it will be rejected).

The default policy will typically allow requests from the 7i server itself and the application server. Most other requests will require a login, with a few necessary exceptions as defined in BrokerSecurityPolicy.xml.

These Security Policy definitions are used by btwebrqb to determine which XML/HTTP requests will be allowed. Any requests not matched here will automatically require a valid login (as determined by the presence of the authentication cookie). Policies will be processed in order, with the last matching policy used as the final result. If no matching rules are found and the request comes from either the 7i server or the application server, the request is allowed.

The <SourceAddresses>, <Connections>, and <Requests> nodes are optional. When not present, all requests are assumed to match the omitted section. A wildcard value of "\*" can be used to match any value. The wildcard character is supported for the "Address", "Name", and "Type" attributes.

<action> values:

Allow: Allow the request without any further requirements

Deny: Deny the request

RequireLogin: Allow the request only if the user has been authenticated

<address> values:

x.x.x.x Any valid IP address

\$LOCALHOST: The local host's IP address (or localhost)

\$APPSERVER: The application server's IP address obtained from the connection definition (telnet host or database host)

<key> Represents an optional "secret key" that must be included in the XML request's header in order to be accepted; if the key is missing or incorrect the policy is treated as "Deny"

<connection name> Must be a valid 7i connection name

<request type>The "Type=" attribute on the inbound XML request

<search path> An XML search path that may optionally be specified to further qualify the matching against the XML request



Following is an example:

```

XML Editor
<Policy Description="Query Strings requiring login" Action="RequireLogin" >
  <Requests>
    <Request Type="attachupload"/>
    <Request Type="attachfetch"/>
  </Requests>
</Policy>

<Policy Description="System Requests" Action="Allow" >
  <Requests>
    <Request Type="Login"/>
    <Request Type="ScreenXML"/>
    <Request Type="ConnectionList"/>
    <Request Type="ToolStatus"/>
  </Requests>
</Policy>

<Policy Description="Local Requests" Action="Allow" >
  <SourceAddresses>
    <SourceAddress Address="localhost"/>
    <SourceAddress Address="appserver"/>
  </SourceAddresses>
  <Connections>
    <Connection Name="*/>
  </Connections>
</Policy>

<!-- Example of search path usage: -->
<!--Policy Description="Disable cache flush" Action="Deny" >
  <Requests>
    <Request Type="ServiceControl" Search="//ServiceRequest[@Type='FlushCaches']"/>
  </Requests>
</Policy-->

```

Registry

## 5.4 Firewall Considerations

Following are the settings necessary to configure SunGard Public Sector Bi-Tech's PC products to connect to a UNIX host system through a firewall. Certain TCP/IP ports must be permitted through the firewall, and Network Address Translation (NAT) may also be required. Since

every brand of firewall software is different, this document will simply state the general requirements rather than attempting to address specific firewall configurations.

Many of the IFAS PC products use ODBC to handle database communication; normally OpenLink provides this layer for Informix and Oracle databases. SQL Server uses the SQL Server ODBC driver. The port numbers used by OpenLink are configurable, but by default the Request Broker will listen on port 5000. In extreme cases it may be necessary to install a proxy on the firewall system, which can act as a bridge between the user the OpenLink Request Broker. This proxy is available for download from OpenLink software, but in most configurations it is not needed. In addition to the ODBC requirement, the following ranges of ports are used (Incoming = PC to UNIX, Outgoing = UNIX to PC):

<b>Product</b>	<b>Ports - Incoming</b>	<b>Ports-Outgoing</b>
Insight	23 (telnet)	
OpenLink	60001 (UDP)	
OpenLink	5000-5999*	
7i (job launching)	8600	
Insight		8400, 8410, 9000-9100, 9500-9600
Cold Storage		8450
Imaging		8301

For Insight, the 9000-9100 and 9500-9600 ranges are estimates; the actual requirement is one port per concurrent connection, starting at 9000 and also at 9500. In the case of a Windows Terminal Server (or Citrix server), this range would allow for a maximum of 100 concurrent users running Insight on that server. Insight also requires NAT support. NAT provides this by issuing a temporary internal IP address to each firewall user, and translating between the two addresses as requests are made. To restrict the range of ports used by OpenLink, set the following in the oplrqb.ini configuration file:

```
[Protocol TCP]
```

```
PortHigh=5999
```

If VPN software is in use, the security is handled internally and the port restrictions should not become an issue. Some sites may wish to use VPN to avoid the configuration issues above.

## 5.5 Integrated Windows Authentication

Integrated Windows Authentication (IWA) is an authentication protocol that can automatically send hashed credentials from a browser to the Web Server. It works primarily with Internet Explorer, but Firefox 1.5 also supports IWA when running the Dashboard application. IWA essentially hashes the user's Windows credentials and maps them to an IFAS user through LDAP. To protect user credentials, if you are using IWA, you will want to also enable SSL, as noted above.

IWA can be particularly useful for avoiding having to repeatedly enter IFAS credentials when accessing IFAS web pages from other applications.

### 5.5.1 How “Integrated Windows Authentication” works with IFAS 7i/Dashboard

Once the following setup has been completed, the 7i login page will detect a new Web.config setting and attempt to auto-login the user based on their domain credentials. Internet Explorer can be configured to automatically send the user's credentials, or to prompt the user to login with their user, password and domain. The domain credentials are then passed off to IIS running on the 7i server, which then validates them against Active Directory. If they are valid, IIS will then execute the 7i login page, which will in turn use the credentials to authenticate to NUUPUS on the user's behalf. If the domain credentials don't match NUUPUS then the regular login screen will be presented and prompt the user to authenticate to 7i. The domain password is not used by IFAS at anytime; only IIS uses it for authentication.

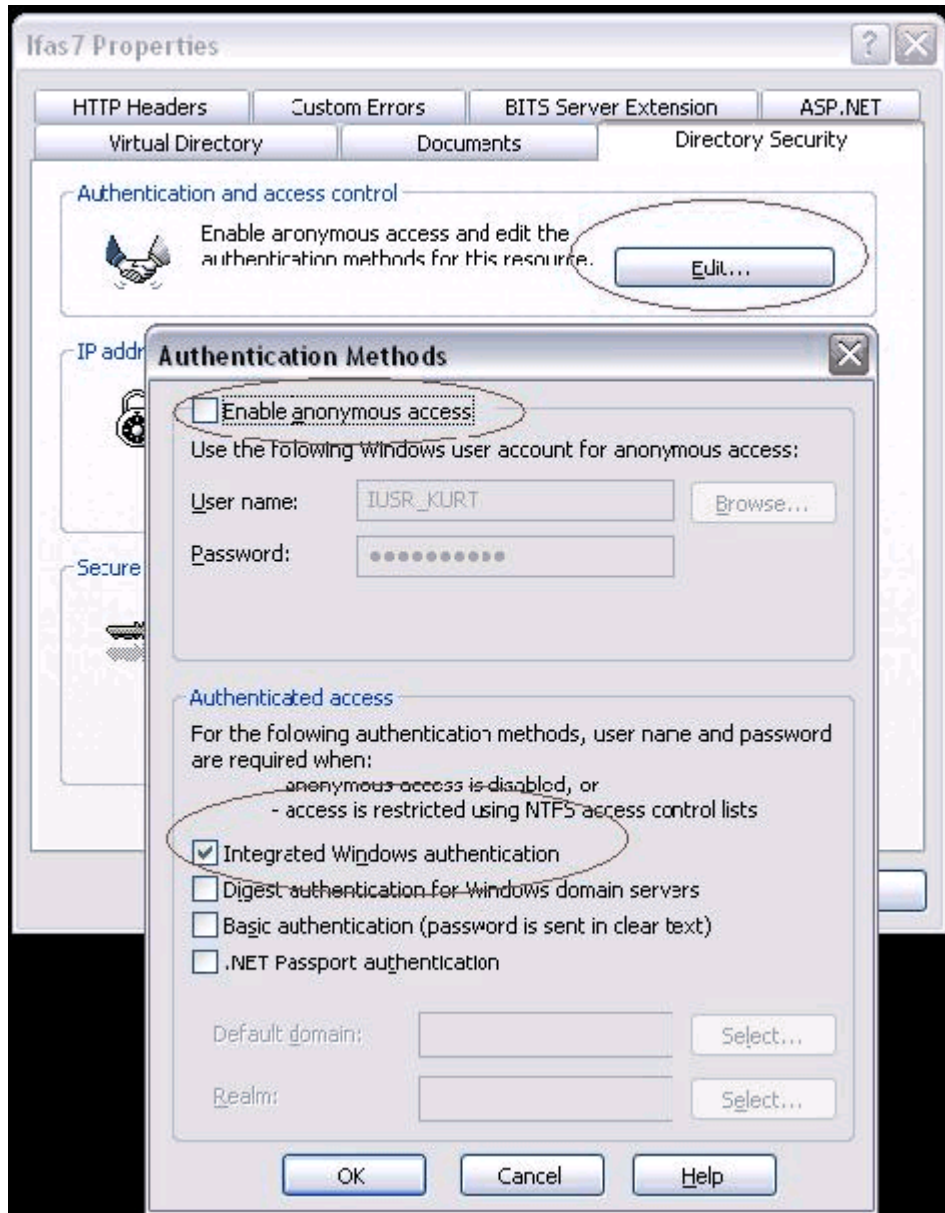
If 7i is running more than one connection and the user successfully authenticates with IWA, they will be presented with the list of connections to choose from. The auto-login feature will only execute the first time for each web browser session. If a user logs off, they will be redirected to the login screen. This will provide the ability to change the IFAS user id when logging back in. A different connection can be chosen at this time as well if more than one is available. You can also close the browser and open a new session to auto-login again.

### 5.5.2 Getting “Integrated Windows Authentication” enabled in 7i

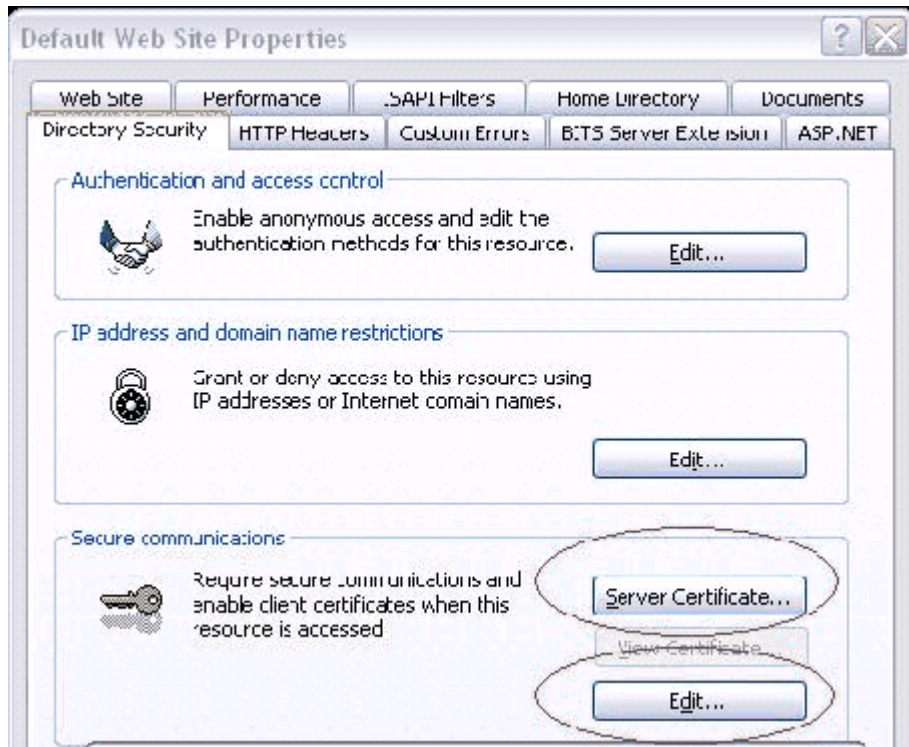
1) In Internet Information Services (IIS) Manager, open the Properties on the ifas7 virtual root. Change settings on the Directory Security tab as follows:

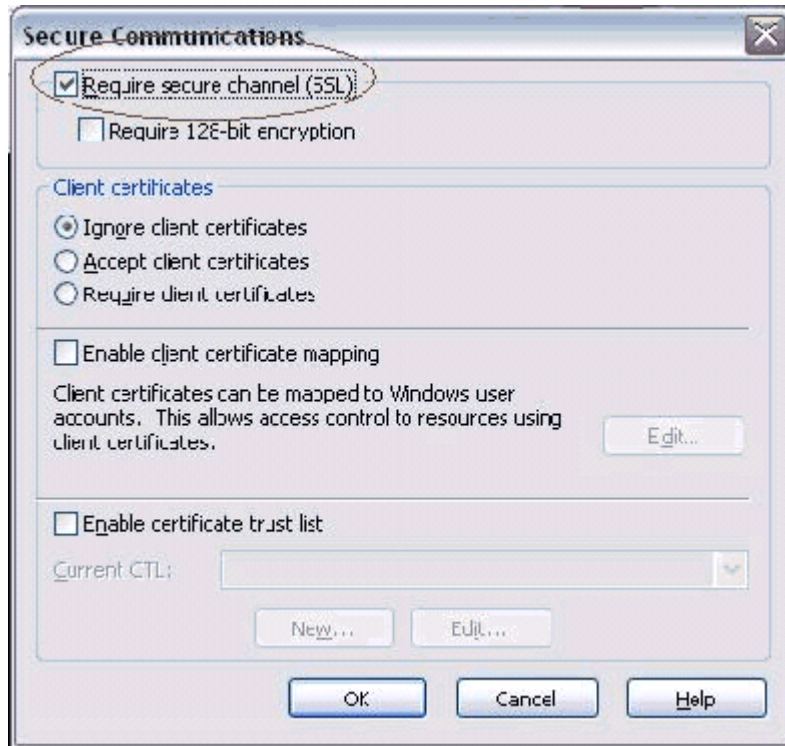
Uncheck 'Enable anonymous access'

## Check 'Integrated Windows Authentication'



2) Implement a SSL certificate and require it on the Ifas7 virtual root. This is optional, but highly recommended.

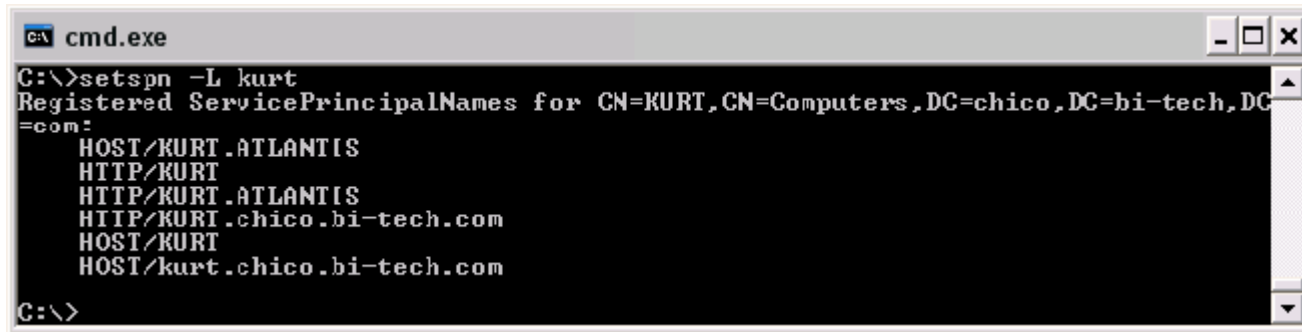




3) Ensure the 7i server is on the domain and registered in Active Directory. A Service Principal Name (SPN) will need to be added for http so IIS can authenticate to AD:

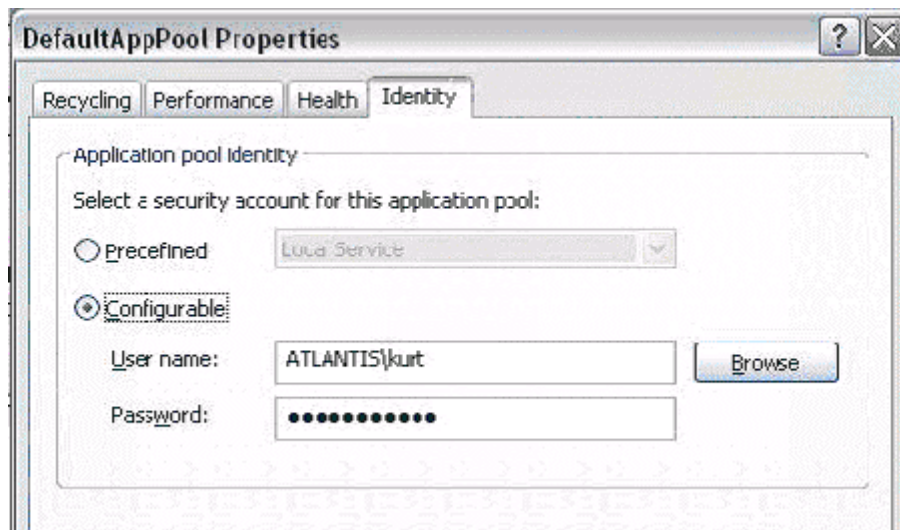
- a) `setspn -l servername`      Can be used to show the SPNs
- b) `setspn -a SPN servername`      Can be used to add the http SPN

Example for a server named 'kurt':



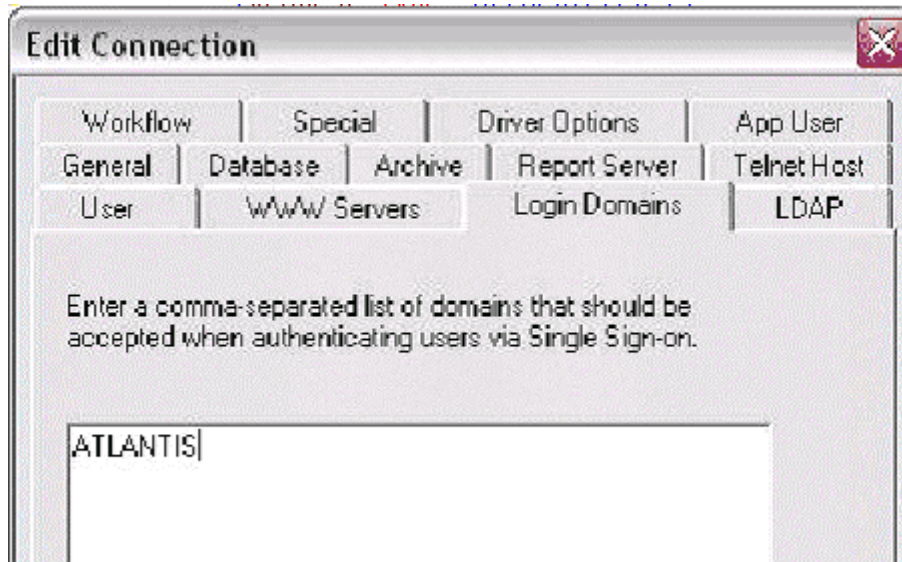
```
cmd.exe
C:\>setspn -L kurt
Registered ServicePrincipalNames for CN=KURT,CN=Computers,DC=chico,DC=bi-tech,DC=com:
HOST/KURT.ATLANTIS
HTTP/KURT
HTTP/KURT.ATLANTIS
HTTP/KURT.chico.bi-tech.com
HOST/KURT
HOST/kurt.chico.bi-tech.com
C:\>
```

4) The Application Pool that the 7i Virtual Root is running as needs to have an Identity of a domain user that can be authenticated to AD.  
Example: ATLANTIS\kurt.



5) In the IFAS Connection Manager, the domain name needs to set on the connection's 'Login Domains' tab.





6) Change WinHTTP proxy settings to always send credentials so that ServerXMLHTTP (7i Data Request) will authenticate as the identity on the Application Pool. If the client is using a proxy server, IE client settings will need to be configured to bypass it.

```
cmd.exe
G:\>proxycfg d p " " "x"
Microsoft (R) WinHTTP Default Proxy Configuration Tool
Copyright (c) Microsoft Corporation. All rights reserved.

Updated proxy settings
Current WinHTTP proxy settings under:
HKRY_LOCAL_MACHINE\
SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\Connections\
WinHttpSettings :

Proxy Server(s) :
Bypass List : *
```



7) Add a setting to the web.config on each 7i server to enable IFAS 7i to use IWA. The top red circle indicates the actual line you will be adding. The bottom circle shows the documentation that is in your web.config file on how to add the setting. You can simply copy the line from below to up above.

```
<appSettings>
  <add key="ProductName" value="SunGard Bi-Tech MyIFAS"/>
  <add key="ThemeName" value="Classic"/>
  <add key="AllowTrace" value="True"/>
  <add key="IntegratedWindowsAuth" value="True"/>
  <add key="ApplicationRoot" value="/ifas7i"/> <!--Note: if you change this be sure to change the MyIFASAuthentication authe
<!-- Optional Application Settings
  BrokerURL
  If your server is in a location other than the local machine you may need to hard-code the request broker.
  This feature is mostly used by developers and the line looks like this:
    <add key="BrokerURL" value="http://localhost/ifas7i/isapi/btwebrqb.dll"/>

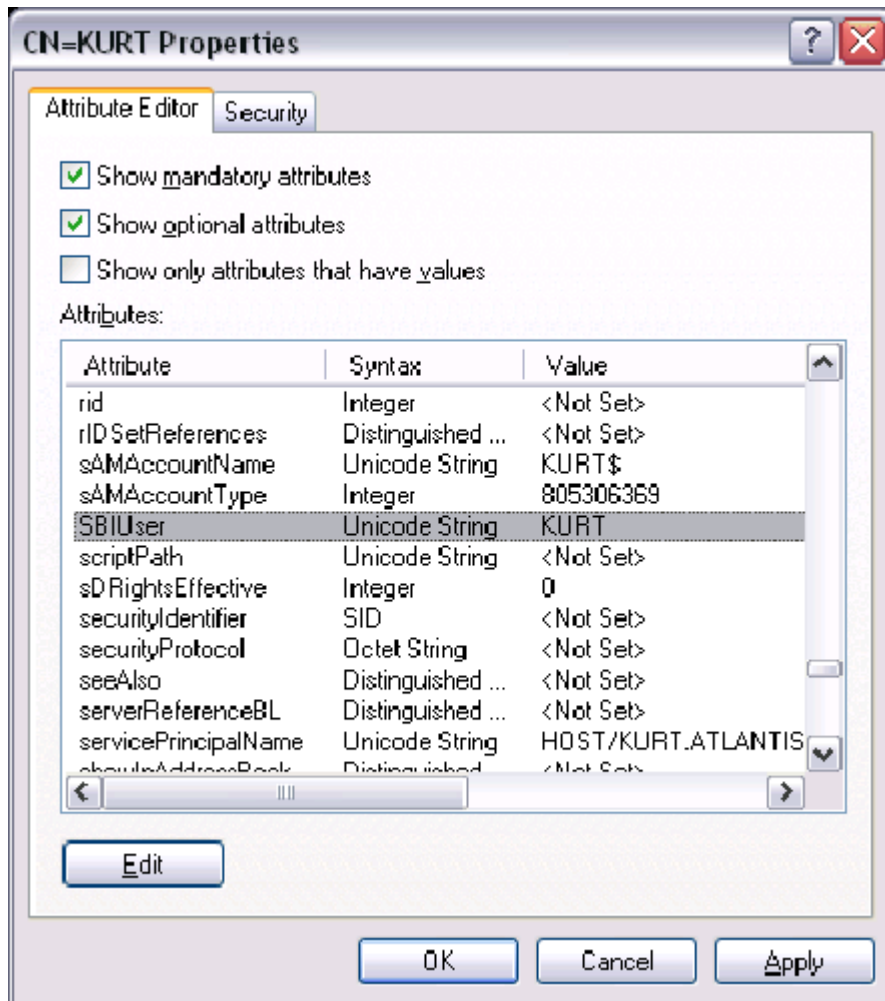
  WFModelId
  If you wish to specify a model other than the one in the AO Settings page, you can do so here:
    <add key="WFModelId" value="AO_APPLICATION"/>

  Connection
  If you would like all Home user so use the same connection by default you can do that by adding a connection setting
  similar to this one:
    <add key="Connection" value="dev_prompt"/>

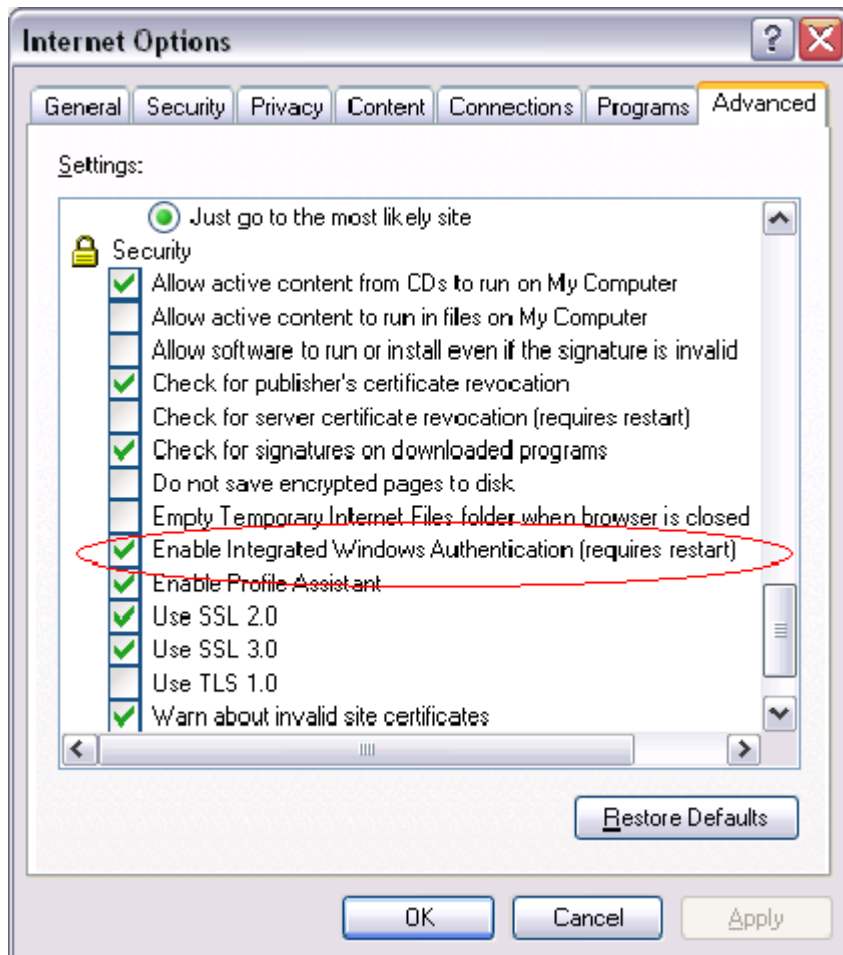
  CDDRequest
  Most of the time a change to the CDD request would also involve changing the scripts that control the viewer as well.
  If you would like to override the CDD request used in drilling to favorites from MyIFAS you can use the following:
  old viewer:
    <add key="CDDRequest" value="/Screens/Nucleus/CddRequest.asp"/>
  or
  new viewer:
    <add key="CDDRequest" value="/CDD/RunReport.aspx"/>
  or
  custom viewer?

  Integrated Windows Authentication
  If you would like IWA to work with 7i and the Dashboard you must correctly configure IIS, Active Directory,
  and turn on a setting here to enable this feature.
  |
  <add key="IntegratedWindowsAuth" value="True"/>
-->
```

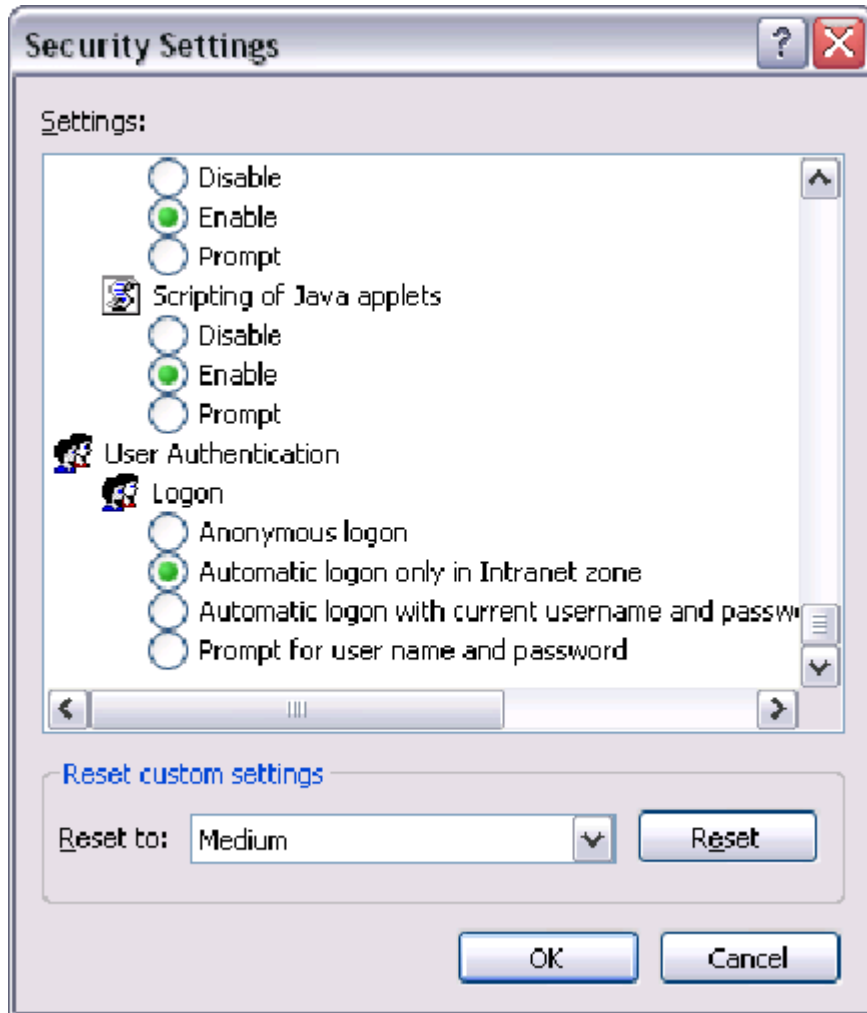
8) If your Active Directory logins are longer than the 8 characters supported by NUUPUS, the SBIUser attribute in LDAP can be used to map an AD user to a NUUPUS user. This will allow your AD login to be any size and map back to an 8 character NUUPUS login. Simply set the attribute to the corresponding NUUPUS user. The LDAP settings will vary depending on the package. Below is the properties show from ASDI Edit in Windows Server 2003:



9) Ensure that Internet Explorer is configured to use IWA. Go to the 'Tools' menu; choose 'Internet Options', then the 'Advanced' tab. Make sure the circled below is checked:



For each zone listed under the 'Security' tab in the same options as above, there are 4 settings controlling the behavior of IE and IWA:

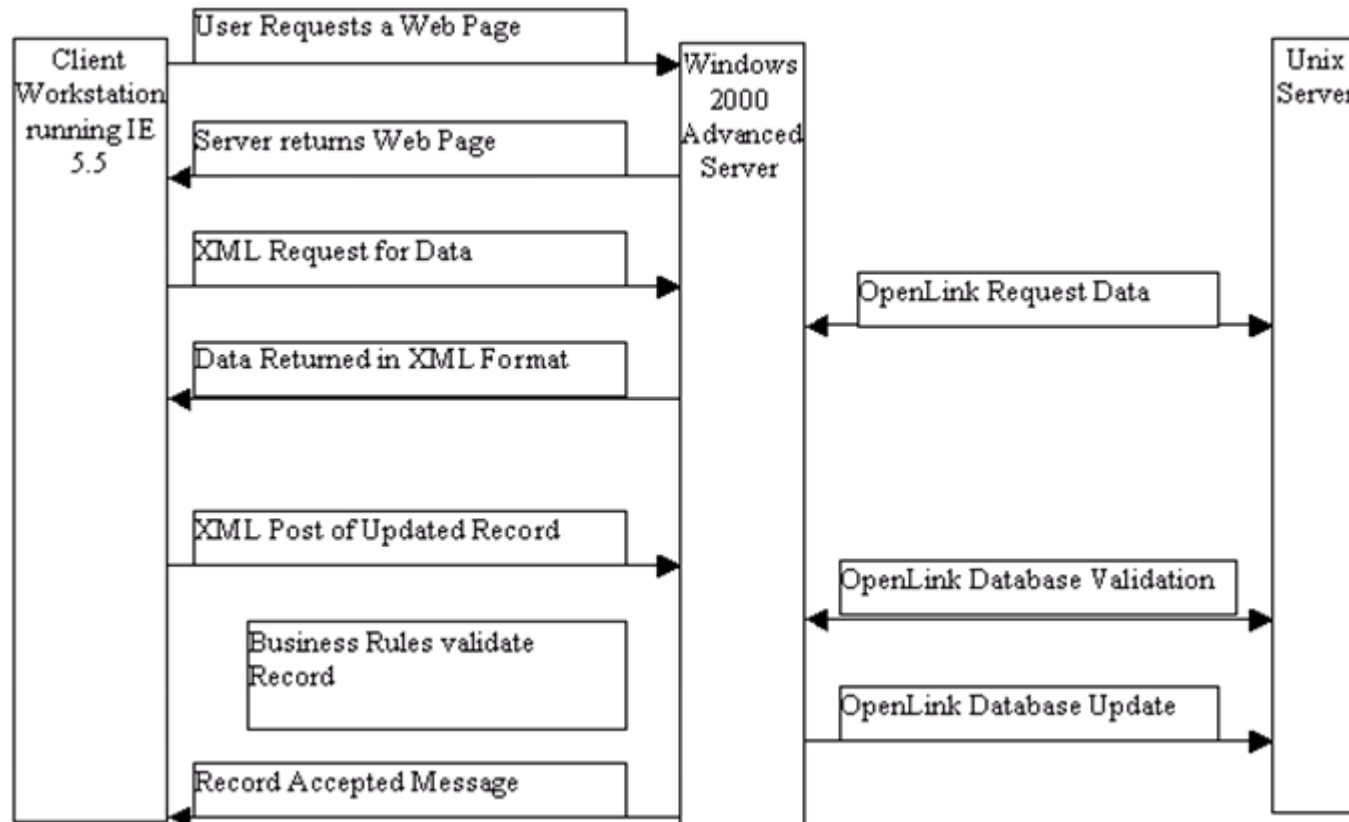


## 6 IFAS Windows 2003 Advanced Server Configuration

**Note:** The following instructions apply only when running multiple Windows 2003 server configurations. If you are using only one Windows 2003 server, skip ahead to the next sub-section (D. Installation).

### 6.1 Overview

The IFAS 7i client maintains all state information. The Windows 2003 server is state-less, and is therefore an excellent candidate for the Network Load Balancing.



Network traffic between the client and the 2003 Server is a HTTP request (whether formatted in XML or not). Traffic between the 2003 Server and the Unix Server is OpenLink (ODBC).

## 6.2 Hardware Recommendations

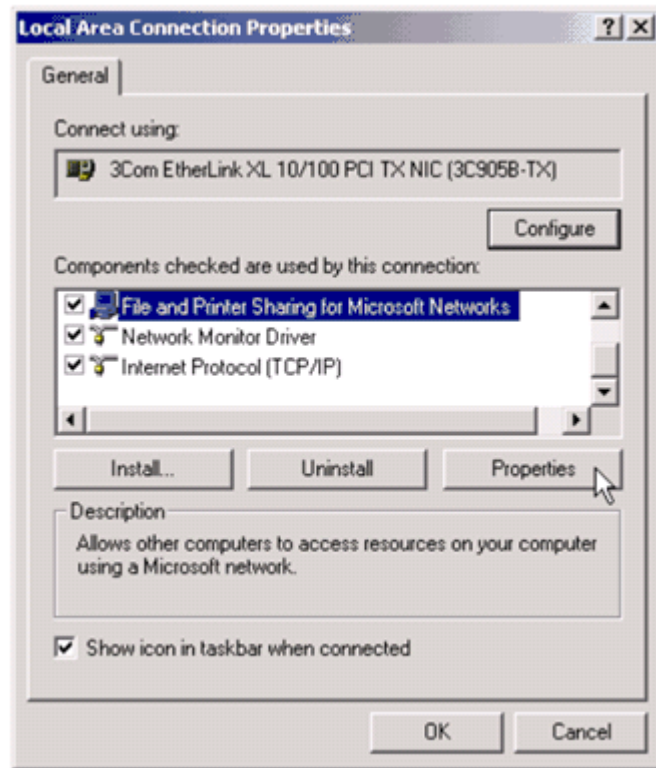
The hard drives should be mirrored (either in hardware or software) for purposes of redundancy. IFAS 7i does not place heavy demands on the I/O subsystem - any RAID solution is appropriate.

Use two network cards to allow multicast traffic to be on a separate physical network dedicated to the server farm.

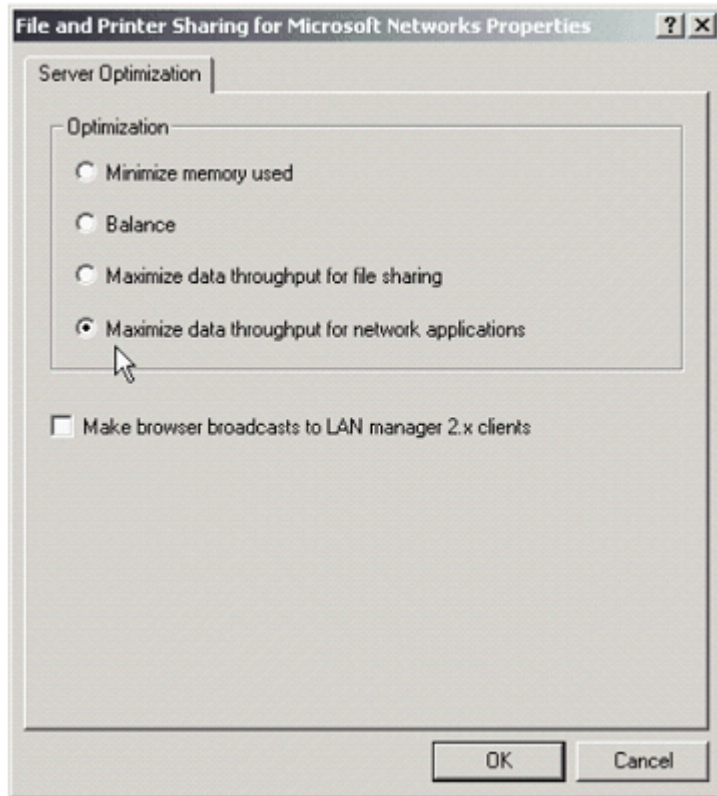
## 6.3 Win2K Recommendations

The servers should not be a primary or backup domain controller, nor should they host other applications (e.g., Exchange or SQL Server).

**Step 1:** Set "Maximize throughput for Network Applications:" Click Start/Control Panel/Network and then right-click on LAN and select Properties. Click on File and Printer Sharing and then select Properties:

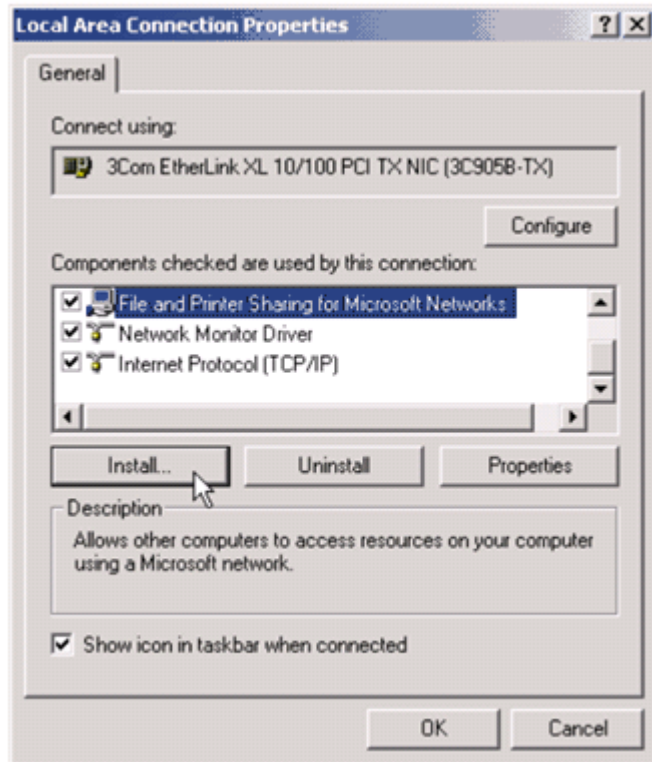


Then, select "Maximize data throughput for network applications." Click OK.

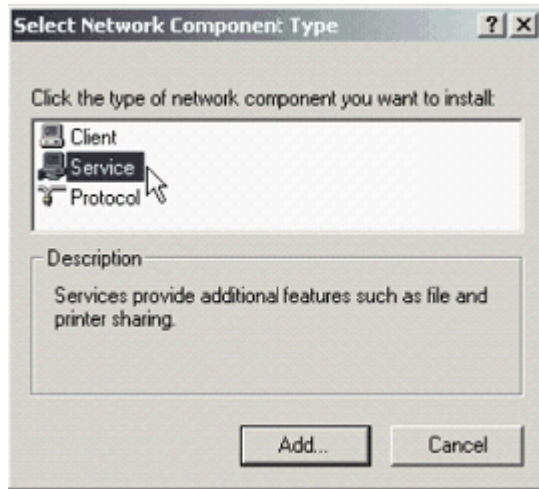




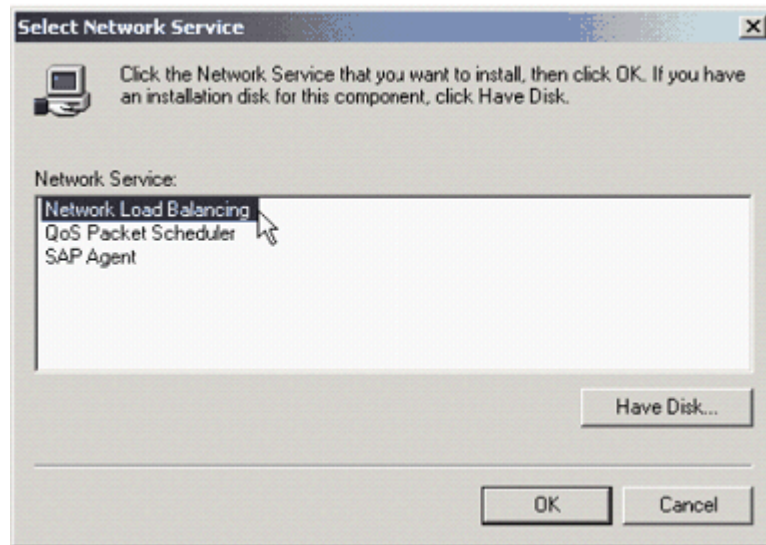
**Step 2:** Install Network Load Balancing. Click Start/Control Panel/Network and then right-click on LAN and select Properties. Click Install:

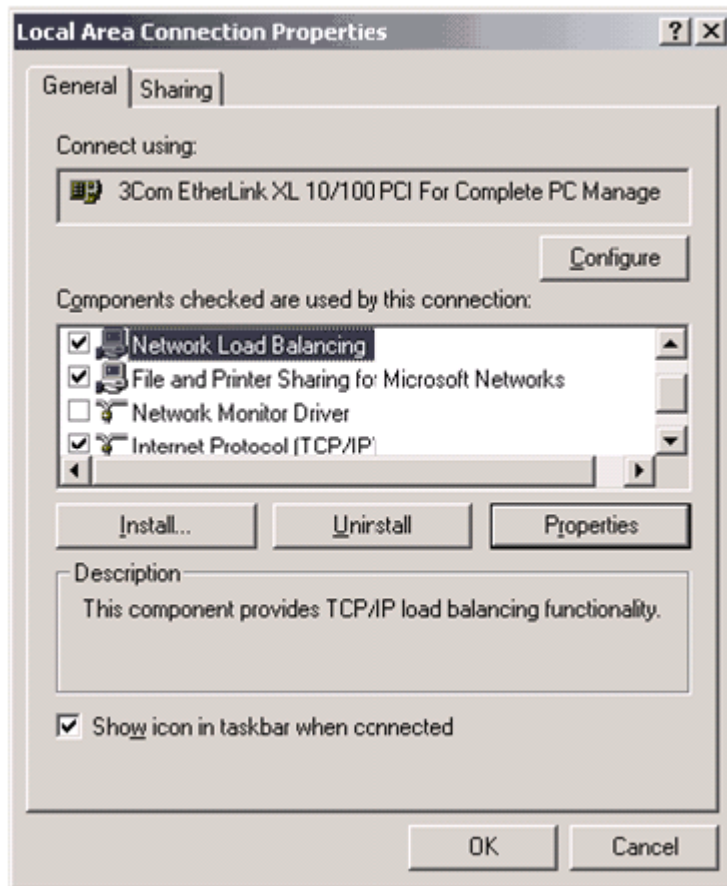


Select Service:



Then select Network Load Balancing:





### Step 3: Example for configuring a single NIC farm environment

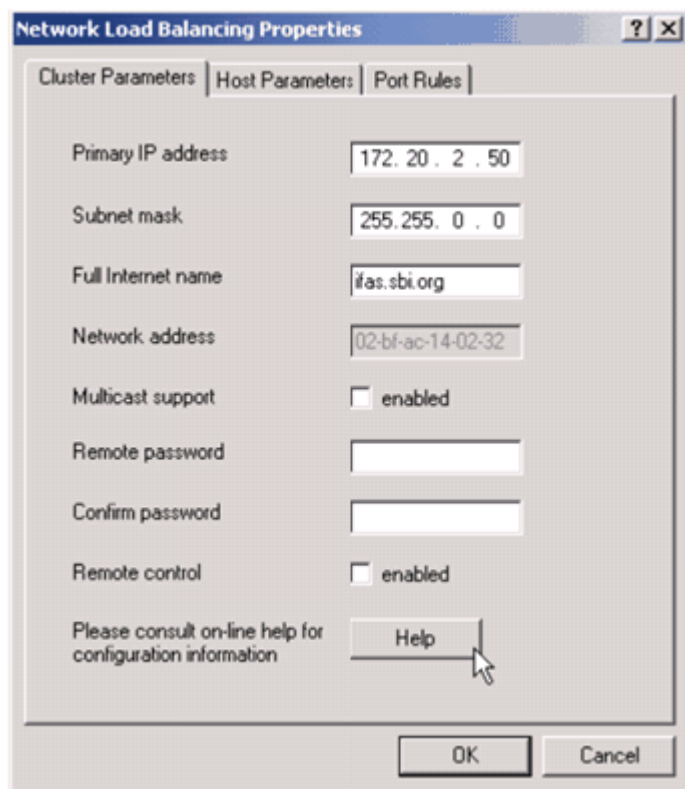
Example: You have four servers in your farm; each will have their own IP address and they will also share a single IP address. Your DNS might look like this:

```
IFAS IN A 172.20.2.50
IFAS1 IN A 172.20.2.51
IFAS2 IN A 172.20.2.52
```

IFAS3 IN A 172.20.2.53

IFAS4 IN A 172.20.2.54

Where IFAS1 through IFAS4 are the names for each server, and IFAS is the name of the farm. There is no requirement that the IP addresses be consecutive. Aside from testing, all requests should be made to the IFAS server.



Initially, don't enable multicast support. Once correct installation is confirmed, read the Microsoft online help for more information about Multicast support if desired.

**Network Load Balancing Properties** ? x

Cluster Parameters | **Host Parameters** | Port Rules

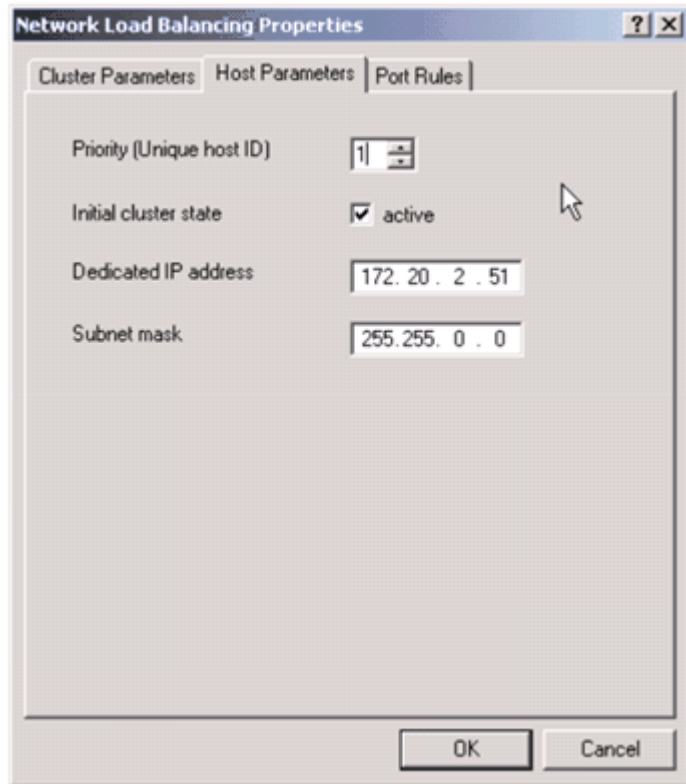
Priority (Unique host ID)

Initial cluster state  active

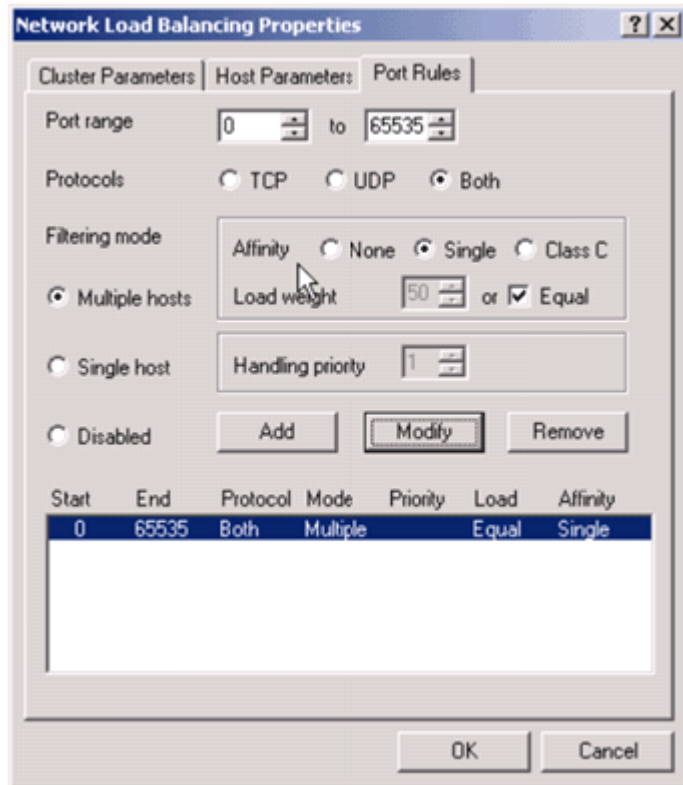
Dedicated IP address

Subnet mask

OK Cancel



Set the dedicated IP address/subnet for this host (IFAS1 using our example above).



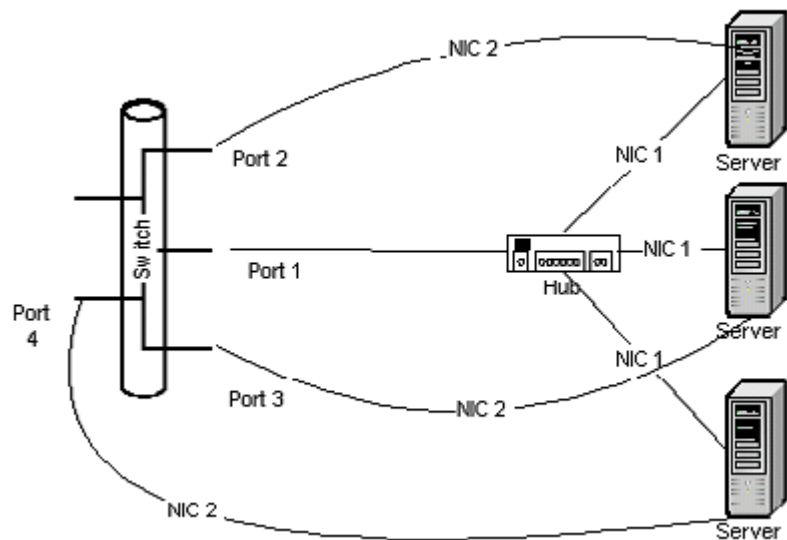
The default settings (0-65535, Both, Multiple, Equal, Single) are adequate. If only 7i applications are running on this farm, the affinity may be set to 'None', otherwise it should be 'Single'.

**Note:** The User interface is a bit awkward. Select the item you wish to modify at the bottom of the screen - this will update the values on the top of the screen. Change these values as appropriate and then press Modify.

**Hint:** See the command-line application WLBS for options on starting and stopping cluster operations gracefully.

### 6.3.1 Dual NIC for IFAS 7i

Inbound traffic (to cluster IP/MAC) is on one switched port. Outbound traffic (from dedicated adapter) goes to specific ports. NLB is on NIC 1 with Metric of 10. Outbound on NIC 2 with Metric of 1. (This information was published on-line by Microsoft Product Support Services. See Q193602 for more information.)

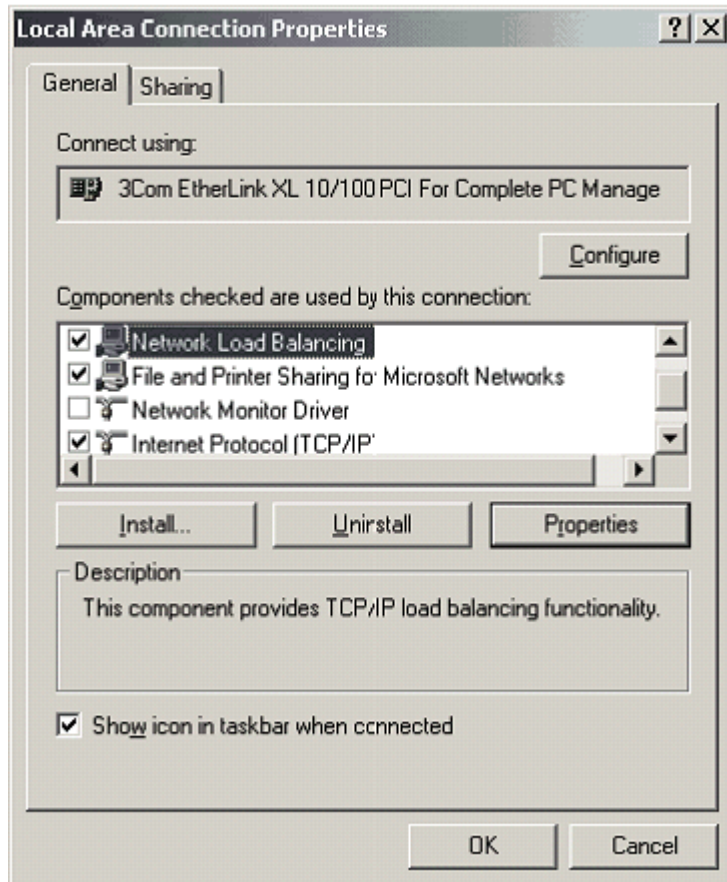


If the 7i Web servers are configured with 2 NICs, then an extra set of IP's are needed. Each Web server in a dual NIC environment should have 3 IPs. The NLB protocol should be enabled on the first NIC and the virtual IP and the NICs primary IP will be defined here. The second NIC will have its own IP. EG:



## 6.3.2 7i Server #1

### NIC1



The screenshot shows a dialog box titled "Network Load Balancing Properties" with three tabs: "Cluster Parameters", "Host Parameters", and "Port Rules". The "Host Parameters" tab is selected. The dialog contains the following fields and controls:

Primary IP address	172.20.1.10
Subnet mask	255.255.0.0
Full Internet name	QAFarm73.chico.bi
Network address	03-bf-ac-14-01-0a
Multicast support	<input checked="" type="checkbox"/> enabled
Remote password	<input type="text"/>
Confirm password	<input type="text"/>
Remote control	<input type="checkbox"/> enabled

At the bottom left, there is a text label "Please consult on-line help for configuration information" and a "Help" button. At the bottom right, there are "OK" and "Cancel" buttons.

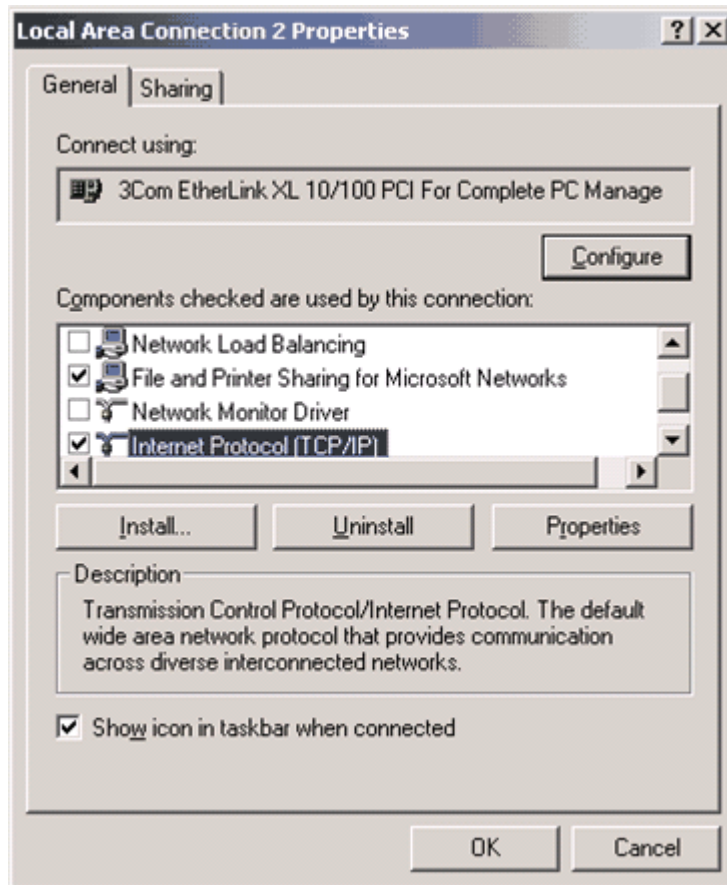
Both the farm IP and the NIC IP are defined here.

Interface metric = 10

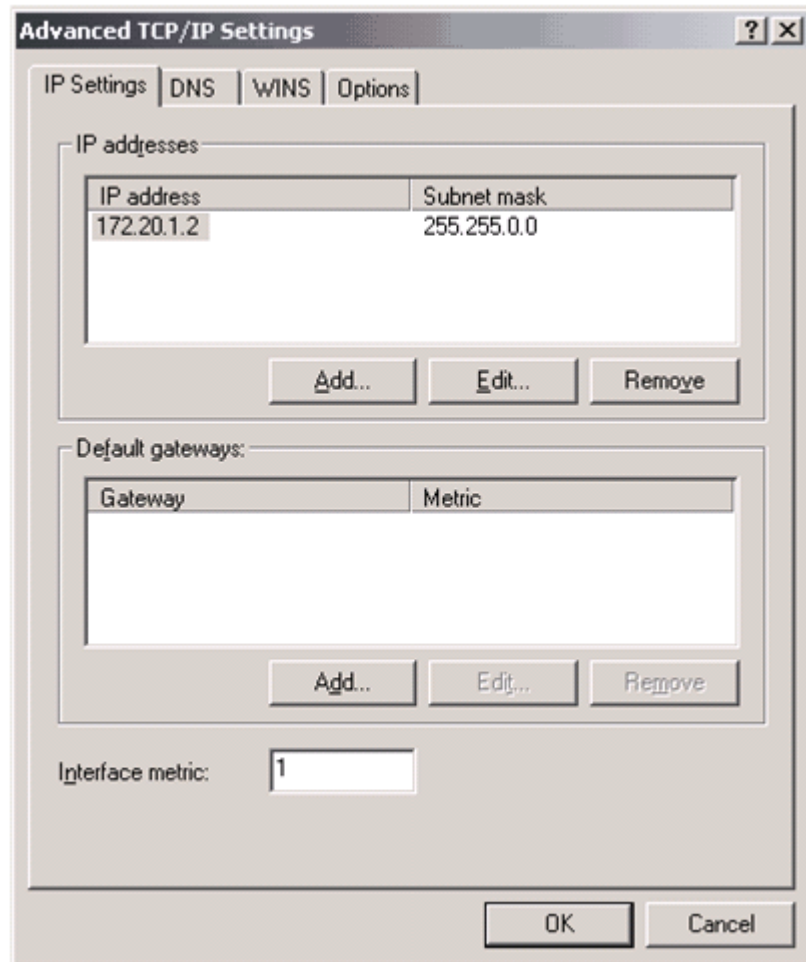
## NIC2

IP: 172.20.1.2

Metric: 1



**Note:** Network Load Balancing is NOT checked on the second NIC.



**Note:** Interface metric =1

### 6.3.3 7i Server #2

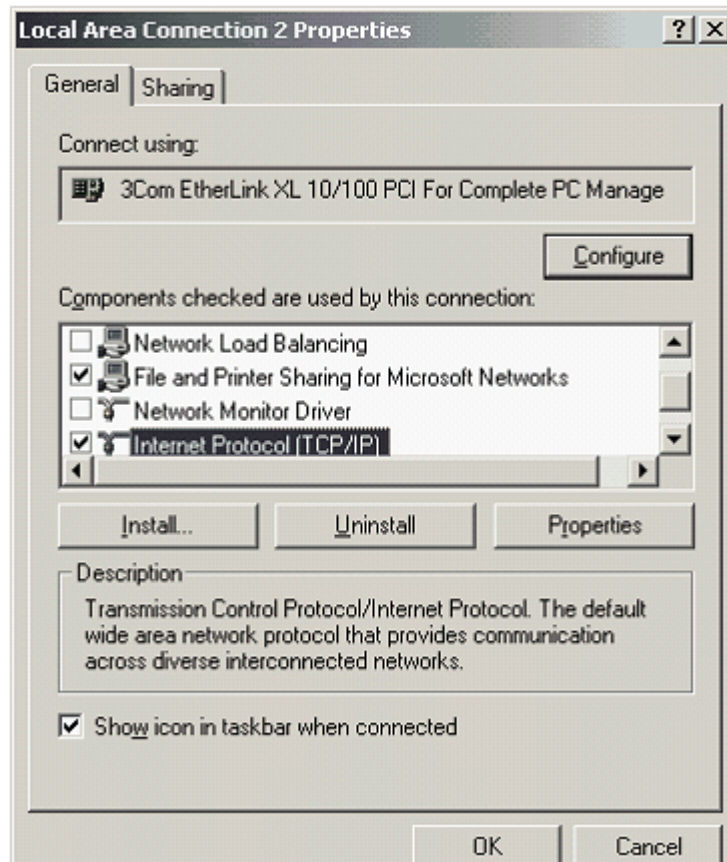
Both the Farm IP and the NIC IP are defined here.

Interface metric = 10

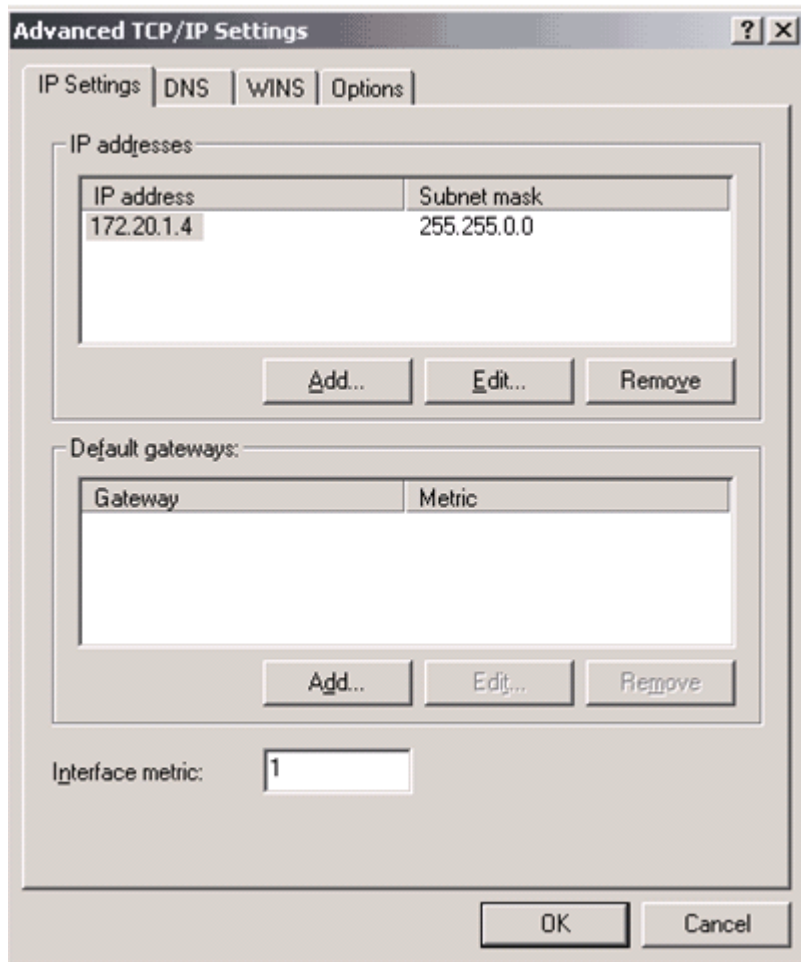
NIC2

IP: 172.20.1.4

Metric: 1



Network Load Balancing is NOT checked on the second nic.



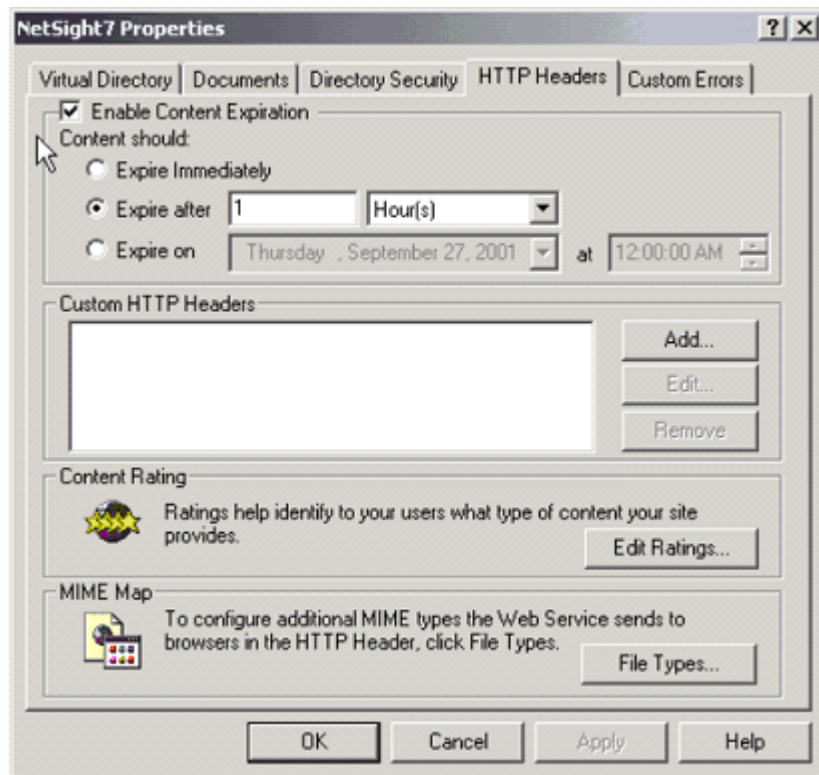
**Note:** Interface metric = 1

## 6.4 Optional

### Content Expiration

Within IIS, it is possible to send custom Content Expiration headers to Internet Explorer that will minimize the number of "Content-Modified" requests back to the server and therefore slightly decrease network usage and improve performance.

Within the Internet Service Manager, right click on the IFAS 7i virtual root and select Properties:



Click on Enable Content Expiration and then select an appropriate time interval. Be aware that setting the expiration too long will cause problems when the next software update is loaded as the client browser may end up with a mix of content, some from the update and some from prior to the update.

A good rule of thumb is that the expiration value should not exceed the time between when an update would normally start (and users prevented from accessing the system) and the time the first user would log in. For example, if an update is estimated to be start at 6:00 PM and the first user normally starts at 5:00 AM, the expiration should be less than 11 hours.

A workstation can use the Internet Explorer Tools/Options to Delete all of the Temporary Files to force current versions of all files.

## 6.5 Unix Server Load Balancing over multiple Network Cards

If conditions warrant spreading network traffic over multiple network cards in your Unix server, try using a technique called Round Robin DNS. Below is an excerpt of UNIX `/etc/named.data` DNS configuration file that illustrates how this can be accomplished:

```
IFASUX 60s      IN  A    172.20.2.51
IFASUX 60s      IN  A    172.20.2.52
IFASUX 60s      IN  A    172.20.2.53
IFASUX 60s      IN  A    172.20.2.54
```

In this scenario, each network interface is assigned a different IP address, and each IP address is mapped to the same name (IFASUX in our example). When a process requests a connection to IFASUX, the DNS server will select one of the four IP addresses to return. Subsequent requests for a new connection will use this IP address for the 60 seconds (the 60s in the example above) and then another request will be made to translate IFASUX into a new IP address.

Please note that IFAS generally asks for a new connection only when the application is starting and therefore the results may not meet expectations.

For further information, please consult the appropriate DNS documentation.



# 7 Posting Preferences and User Defaults Conversion

## 7.1 The map2vb Program

The first time the installation wizard runs, user defaults (in NUUPDF) and posting preferences will be translated from the current definition in Nucleus to a series of Visual Basic and XML script files. Any setting changes made in NUUPDF, GLUTSPSI, or GLUTSPPP after the initial installation will cause the map2vb program to run.

The map2vb program converts posting strategies and default definitions into Visual Basic and XML script files for IFAS 7i to use. The files are located on the Web server. The default location is C:\Inetpub\wwwroot\Ifas7\Connections\connection name\settings. The map2vb program will prompt for the server information if it doesn't have the settings.

## 7.2 Using map2vb

Upon completion of install of the IFAS 7i Web products, it will be necessary to manually run the map2vb program from UNIX. To do this, telnet into your UNIX account. At the Unix prompt type :

```
map2vb
```

This will generate a series of questions:

Enter IFAS7i Win2K3 Server Name:

If you have a farm set up for the Windows 2003 Servers, then enter the name of the farm here. In a single Web server setup, enter the name of the single Web server

Enter IFAS7i Virtual Root;[return=netsight7] :

This is the name supplied during the 7i install on the web servers. Default is Ifas7. Press Enter.

Enter Connection Name:

This is the connection name that you compiled the screens for on the Windows 2003 Web server. Enter that connection name here.

If all information is entered correctly, then output should be generated similar to the below:

Posting Preferences:

OHBatchDetail.vbs

SIRBatchDetail.vbs

SIMItemDetail.vbs

Default Mapping:

OHBatchDetail.vbs

POPPvDetail.vbs

POIItemDetail.vbs

SIRBatchDetail.vbs

SIOOrderMaster.vbs

Posting Logic:

OHBatchDetail.vbs

ARTrnsDetail.vbs

SIMItemDetail.vbs

OHBatchDetailDefaults.xml

POPPvDetailDefaults.xml

POIItemDetailDefaults.xml

SIRBatchDetailDefaults.xml

SIOOrderMasterDefaults.xml

ARTrnsDetailDefaults.xml

SIMItemDetailDefaults.xml

Updating Server <http://servername/netsight7>

OHBBatchDetail.vbs        OK

SIRBatchDetail.vbs        OK

SIMItemDetail.vbs        OK

POPPvDetail.vbs        OK

POIItemDetail.vbs        OK

SIOOrderMaster.vbs        OK

ARTrnsDetail.vbs        OK

OHBBatchDetailDefaults.xml    OK

POPPvDetailDefaults.xml    OK

POIItemDetailDefaults.xml    OK

SIRBatchDetailDefaults.xml    OK

SIOOrderMasterDefaults.xml    OK

ARTrnsDetailDefaults.xml    OK

SIMItemDetailDefaults.xml    OK

Flushing cache...        OK

### Saving upload settings in /ifas/bsi/pub/.map2vb.modtest6

This information is saved to a file in the bsi account. If your Web server name changes or your connection name changes, this will need to be updated. The file is stored in the /bsi/pub directory and has the name of .map2vb.databasesname where "databasesname" is the name of the database for that account.

The running of the map2vb program will need to be done in each account that you expect to run IFAS 7i Web screens against. The Data Processing Service will need to be running on the 7i Web servers for this to work correctly. If a farm name is used with map2vb, then all machines in the farm will be updated with the map2vb files.

Changes to the connection name, virtual root, and server name require the map2vb program to be run manually using the switches listed below.

Step 1: Login to your UNIX system as a user with access to the shell prompt.

Step 2: Enter map2vb at the prompt.

Switches available when running map2vb include the following:

- n xxx (xxx=connection name)
- r virtual root (e.g., IFAS7)
- s server name
- c (compatibility mode for the pre-7.1 components)
- p Specify the port number that IIS is listening on for 7i. Defaults to port 80.

## 8 CDD.net Configuration

To browse to CDD.net use the following URL

```
http://servername/IFAS7/CDD/default.asp
```

Where servername is the name of the machine on which CDD.net is installed and IFAS7 is the 7i URL.

For CDD.net to work correctly, it requires that a LOCAL printer is set up on the CDD.net server. In a farm setup, the same printer needs to be defined on each server hosting CDD.net. This printer does not need to actually be connected to the server. It can be a fake printer. CDD.net needs the printer definitions that are loaded by Windows for the printer. By default CDD.net will look for a printer named LocalPrinter. During the install of 7i, a "fake" Localprinter will be installed. If this "LocalPrinter" is removed, it will be necessary to recreate it.

Use the Windows Printer Wizard to add a LOCAL printer to each 7i server. Give this LOCAL printer the same name on each 7i server. Use the configuration page on each 7i server to add this printer to the registry so CDD.net recognizes it. Refer to Appendix E for information on the configuration page.

Specifically, the printer must be able to be accessed by IUSR\_IFAS. Creating a local printer is the easiest way to ensure IUSR\_IFAS has access to it.

## 9 Remote Configuration

### 9.1 Remote Configuration

Given that the IFAS servers may reside in a remote location, 7i provides basic configuration and management through a Web interface. To access the configuration page, you should browse to:

`http://<servername>/ifas7/admin/config/admin.asp`

#### 9.1.1 General Service Configuration

The General Service tab provides the ability to update general configuration parameters that are associated with the Data Processing Service.

The following section details the meaning of each parameter and gives recommendations for minimum and maximum values.

**Initial System Threads:** Internally the Data Processing Service maintains a thread pool to process incoming service requests. The primary request type is for a list of active connections that are being serviced on the local machine. This thread pool starts with an initial number of threads that are governed by this value.

**Maximum System Threads:** The internal system request thread pool begins with the number of threads specified by the “Initial System Threads” value. As requests come in a thread from the pool processes them. If there are more requests than threads and the maximum thread value has not been reached, a new thread will be created and added to the pool.

**Maximum Restarts:** This value controls how many times a particular queue processor will be restarted by the service before ceasing to perform the restart. This value allows you to control having a situation where the service continually restarts the queue processor when the Unix host is not available.

**Maximum Restart Failures:** This value controls the maximum number of failures that can occur during the attempt to restart a queue processor. If a queue processor loses its connection to the database and has to be restarted, there may be a time during which the Unix box will not accept new requests.

**Maximum Connection Loss:** This value controls how many times a processor may lose its connection to the database before indicating that it needs to be restarted by the service. When a database connection is lost, the queue processor will stop processing messages on the thread that lost the connection and spawn a new thread.

**Ping Interval:** This value governs the time (in seconds) between inter-group ping messages. Each instance of the Data Processing Service maintains contact with other instances in the same processor group by sending a ping message at an interval specified by this value. This is used to maintain a consistent configuration amongst the servers.

**Restart Delay:** This value specifies the number of seconds that the Data Processing Service should wait before restarting a processor that has requested restart. In some cases, the back-end database connectivity layer may wait for a particular interval to elapse before accepting new requests, this allows the software to wait for that interval before attempting to establish a new connection. The default value is 10 seconds and the minimum value is 5 seconds. For example: The OpenLink request broker will wait for a specified interval before accepting new connections. This value is controlled by the “ReceiveTimeout” configuration parameter in the OpenLink rulesbook (oplrqb.ini). If this value is set to 30, the restart delay is set to 5 and the maximum restart failure is set to 3, it is possible that the service could attempt to restart the processor 3 times, with each failing, before the OpenLink request broker accepts the request. At this point, the Data Processing Service would not attempt to restart the queue processor and would log an event to the Event Log.

**Restart Processors:** This value controls whether or not the Data Processing Service will attempt to restart a processor that has lost its connection to the database or faulted due to network-related issue, etc.

## 9.1.2 Queue Processor Configuration

The Queue Processors tab allows configuration of the queue processor settings for the server group.

To change the parameters for an active connection, first select the connection from the Active Connections combo box. After changing one or more of the configuration parameters, the server can be updated by pressing the 'Update Server' button.

To add a new connection to the list of active connections, press the “Add Connect” button and enter the name of the connection. After providing values for all of the configuration parameters, press the “Update Server” button and the connection will be activated on the server group.

**Note:** the name must match the spelling of the connection exactly and the connection must be defined in the same system catalog.

### 9.1.3 CDD Reports

**NOTE:** SunGard Public Sector Bi-Tech provides a sample CDD reports on the install DVD. These reports are found on the IFAS DVD in the Reports directory. When loading CDD Reports from the install DVD it is necessary to first copy the .CDD or .czf files to your local hard drive before loading them into your system catalog database. They can \*not\* be loaded directly from the install DVD.

The configuration parameters for the CDD Report Service are defined on the tab titled “CDD Reports”. The configuration parameters, their defaults and how they affect the behavior of the server are outlined below.

**Report Expiration:** After a report is run, there are intermediate files that are kept in the directory specified by the File Storage Path parameter. The report expiration value indicates how many seconds a file should be kept on disk before removing the file from the file system.

**File Cleanup Interval:** The file cleanup interval specifies the interval in seconds between runs of the process that removes the temporary intermediate files.

**Initial Instances:** CDD report requests that are made under IFAS 7i are queued to a request queue that is serviced by a number of instances of CDD. The initial instances specifies how many instances of CDD are launched when the first report request is received and serves as a low water mark for how many instances will remain alive.

**Maximum Instances:** The maximum instance count represents the largest number of CDD instances that will be active at any one time. If the initial instances value is set to 3, the maximum instances is set to 5 and 6 requests arrive, 1 of the requests will be queued until one of the other 5 requests is fully processed by CDD.

**Instance Timeout:** The instance timeout represents the time, in seconds, that an instance of CDD will remain running without running a report. The number of instances will not fall below the low water mark specified by the initial instances value.

**File Storage Path:** The file storage path represents the physical directory under which the temporary CDD files are created. Place this directory on a partition that has 1 GB of free disk space and configuring the directory for compression is used.

**Attachment Directory:** The attachment directory represents the physical directory on the 7i web server under which the temporary images are stored. This directory should be placed on a partition that has 1 GB of free disk space and configuring the directory for compression is used.

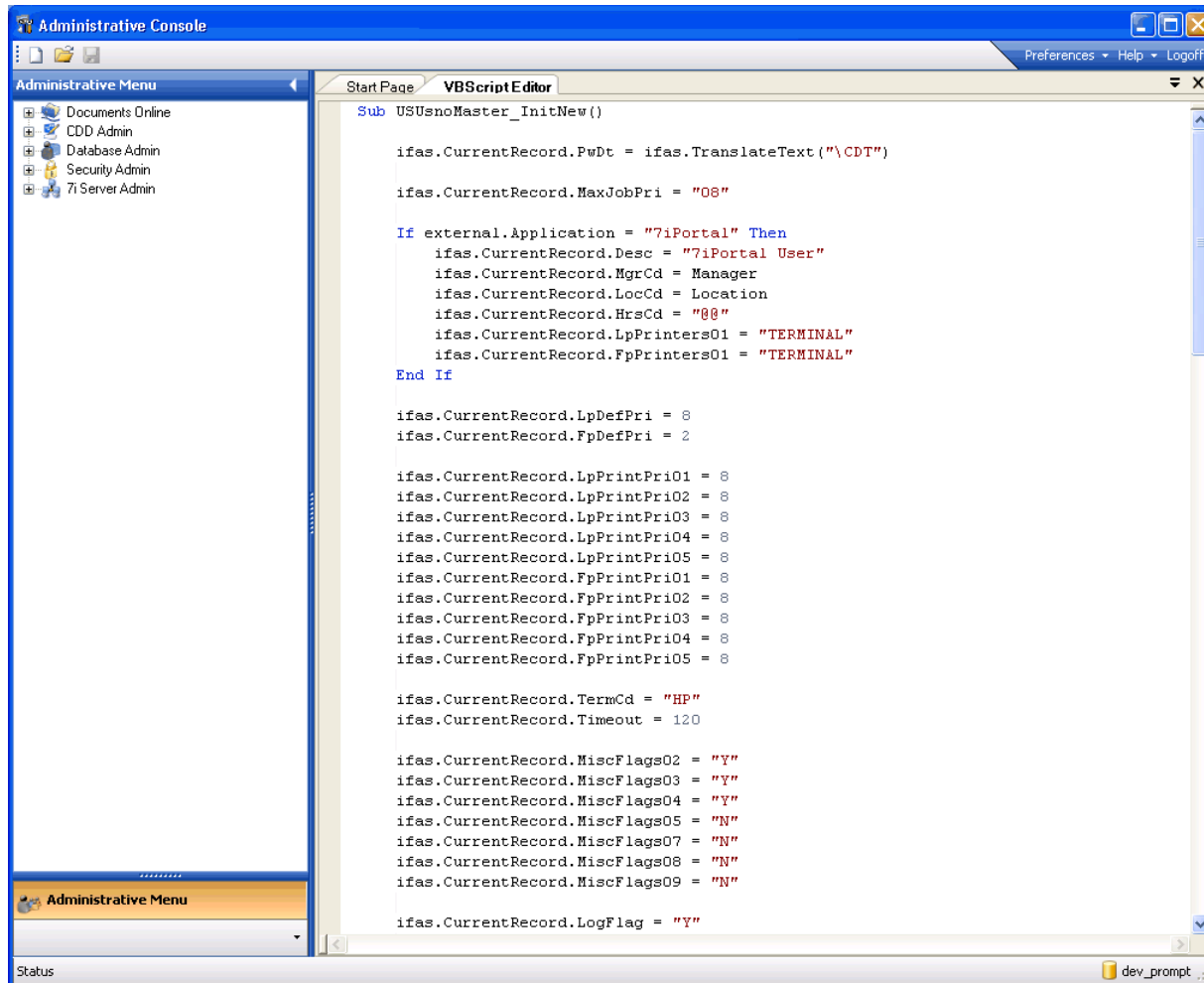
**Attachment Cleanup Interval:** The attachment cleanup interval is displayed in minutes and displays how often the attachment directory is cleaned out.



**Default Printer:** The default printer is the local printer that is defined during install and needed by IFAS 7i and CDD.net. This printer **MUST** be the same on each 7i web server in the farm.

## 10 VBScript Editor

The VBScript Editor Screen in the Admin Console is intended to assist in the editing of VBScript documents (\*.vbs) on the 7i Server.



## 10.1 Menu Options

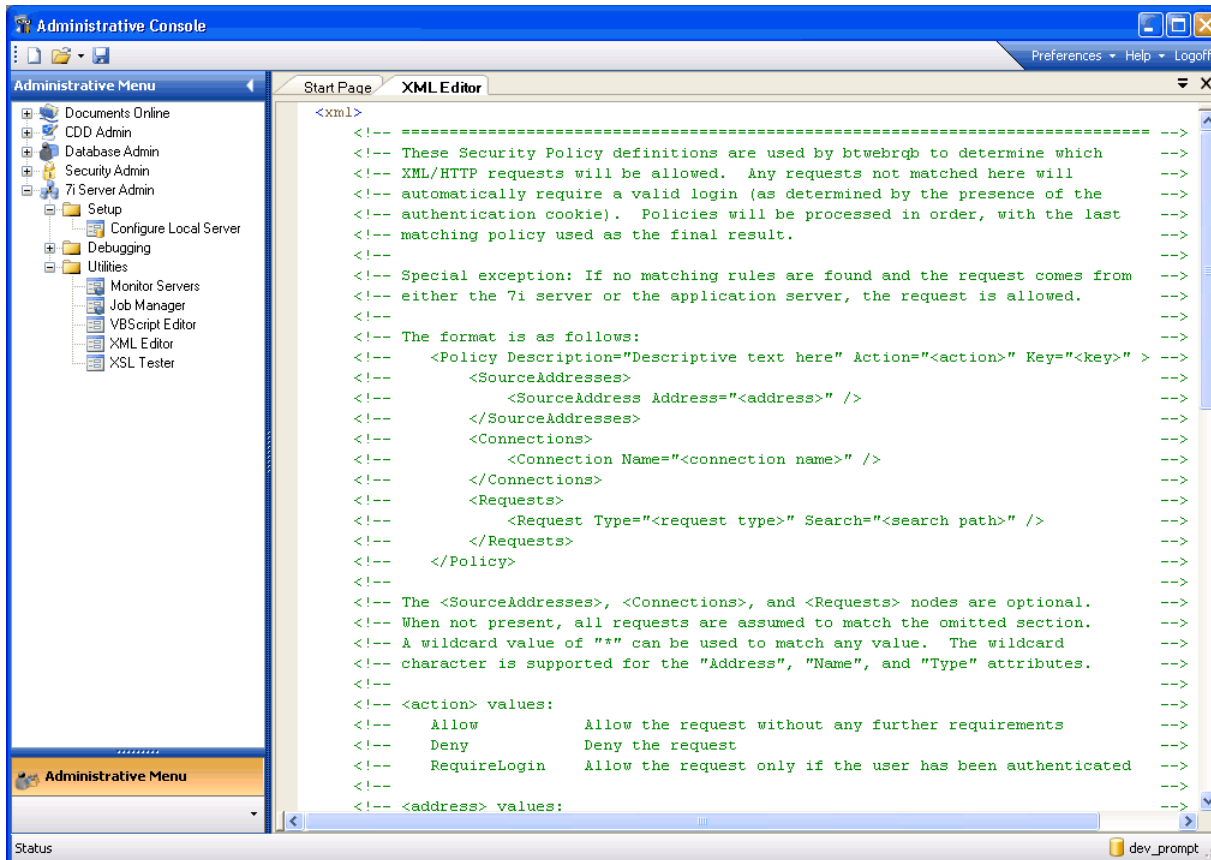
**New:** Creates a new VBScript document.

**Open:** Opens an existing VBScript document on the server.

**Save:** Brings up the Save File dialog allowing the current file to be saved or renamed.

# 11 XML Editor

The XML Editor Screen in the Admin Console is intended to assist in the editing of XML documents (\*.xml) on the 7i Server.



## 11.1 Menu Options

**New:** Creates a new xml document.

**Open:** Opens an existing xml document on the server.

Open Common Server Files

**Broker Security Policy:** The xml file used to configure the IFAS Request Broker policy on the server.

**Save:** Brings up the Save File dialog allowing the current file to be saved or renamed.

## 12 XSL Tester

The XSL Tester screen allows an XSL role such as a Security Role to be applied to a sample XML Document. This utility is used when the XSL syntax itself is in question and during the troubleshooting process the actual transformation needs to be seen. This is regarded as a very technical debugging step and is primarily used by Developers at SunGard Public Sector.

The screenshot displays the Administrative Console interface with the XSL Tester tool. The interface is divided into several sections:

- Administrative Menu:** A sidebar on the left containing various system management tools such as Documents Online, CDD Admin, Database Admin, Security Admin, 7i Server Admin, Setup, Debugging, Queue Journal, Utilities, Monitor Servers, Job Manager, VBScript Editor, XML Editor, and XSL Tester.
- XSL Transformer Code:** The main window displays the following XSL code:
 

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
  <xsl:template match="/">
    <xsl:copy>
      <xsl:apply-templates />
    </xsl:copy>
  </xsl:template>
  <xsl:template match="**">
    <xsl:copy>
      <xsl:attribute name="desc">
        <xsl:value-of select="@desc" />
      </xsl:attribute>
      <xsl:attribute name="ACCESS-R">
        <xsl:value-of select="@ACCESS-R" />
      </xsl:attribute>
      <xsl:attribute name="ACCESS-W">
        <xsl:value-of select="@ACCESS-W" />
      </xsl:attribute>
      <xsl:attribute name="ACCESS-U">

```
- XML Document Output:** The bottom pane shows the transformed XML document:
 

```
<XML desc="XML Root">
  <APPLICATION desc="Application Root" ACCESS-R="0" ACCESS-W="0" ACCESS-U="0" ACCESS-D="0" ACCESS-X="0" I
  <SUBSYSTEM_AP desc="Accounts Payable" ACCESS-R="0" ACCESS-W="0" ACCESS-U="0" ACCESS-D="0" ACCESS-X=
  <TABLE_AP desc="Accounts Payable Data" ACCESS-R="0" ACCESS-W="0" ACCESS-U="0" ACCESS-D="0" ACC
  <TABLE_AP_IMAGE desc="AP_IMAGE [Image based AP 3way match]" ACCESS-R="0" ACCESS-W="0" ACC
  <TABLE_AP_IMG_ITEMS desc="AP_IMG_ITEMS [item splits for ap imaging]" ACCESS-R="0" ACCESS-W=
  <TABLE_OHA_ACCT_DTL desc="OHA_ACCT_DTL" ACCESS-R="0" ACCESS-W="0" ACCESS-U="0" ACCESS-D="0"
  <TABLE_OHB_BATCH_DTL desc="OHB_BATCH_DTL" ACCESS-R="0" ACCESS-W="0" ACCESS-U="0" ACCESS-D='
  <COLUMN_OHB_BATCH_DTL_OH_GL_GR desc="OH_GL_GR" ACCESS-R="0" ACCESS-W="0" ACCESS-U="0"
  <COLUMN_OHB_BATCH_DTL_OH_GL_KEY desc="OH_GL_KEY" ACCESS-R="0" ACCESS-W="0" ACCESS-U="0"
  <COLUMN_OHB_BATCH_DTL_OH_GL_OBJ desc="OH_GL_OBJ" ACCESS-R="0" ACCESS-W="0" ACCESS-U="0"
  </TABLE_OHB_BATCH_DTL>
  <TABLE_OHC_CLASS_DTL desc="OHC_CLASS_DTL" ACCESS-R="0" ACCESS-W="0" ACCESS-U="0" ACCESS-D='
  <TABLE_OHD_PETTY_DTL desc="OHD_PETTY_DTL" ACCESS-R="0" ACCESS-W="0" ACCESS-U="0" ACCESS-D='
  <TABLE_OHH_BATCH_MSTR desc="OHH_BATCH_MSTR" ACCESS-R="0" ACCESS-W="0" ACCESS-U="0" ACCESS-I
  <TABLE_OHP_PETTY_MSTR desc="OHP_PETTY_MSTR" ACCESS-R="0" ACCESS-W="0" ACCESS-U="0" ACCESS-I
  <TABLE_OHS_SEC_MSTR desc="OHS_SEC_MSTR" ACCESS-R="0" ACCESS-W="0" ACCESS-U="0" ACCESS-D="0"

```
- Status:** The bottom left corner shows the status bar with the text "dev\_prompt".

## 12.1 Top Panel

The top panel is used to edit and view the XSL stylesheet.

**New:** Creates a new XSL document in the top panel.

**Open:** Opens an existing XSL document in the top panel.

**Save:** Saves the current XSL document to the desired location.

## 12.2 Bottom Panel

The bottom panel is used to view both the original XML document as well as the result of the transformation. Editing of the original XML document should be done in the XML Editor screen.

**Open:** Opens an existing XML document.

**Save:** Saves the current and possibly transformed XML document to the desired location.

**Apply XSL:** Applies the XSL stylesheet in the Top Panel to the XML Document in the Bottom Panel.

# 13 Troubleshooting

## 13.1 Request Logging

As service requests flow through the 7i server, a text log file will be created in the Log directory for each active btmqhost thread. The log directory location is configurable, but the default location is directly under the IFAS software location (for example, C:\Program Files\SunGard Bi-Tech\bin). The files will be named using this pattern:

<Queue type>\_<process id>\_<thread id>.txt

For example:

C:\Program Files\SunGard Bi-Tech\Log\Screens High\_1000\_2000.txt

The <Queue type> portion will be one of the following:

- Screens High
- Screens Low
- Nucleus High
- Nucleus Low
- Reports High
- Tools High
- Daemon High

Each log file will be removed when the corresponding request has completed, so the files in this directory should represent only pending requests or incomplete requests. Any remaining files can be examined for evidence of problems that may have been encountered. This information, in conjunction with any entries found in the application event log, can be extremely useful when tracking down problematic requests -- for example, a particular request that triggers an endless loop within a business rule.

The log file will contain the following information:

- A time stamp



- The XML request submitted by the 7i client
- The name of each business rule that has been called during processing
- The SQL for any insert, update, or delete operations

When 7i creates an "Error" event in the Application event log, the request log information for the current request will be appended to the event log. This will provide context information for most types of errors reported by 7i.

To configure this feature, including the location of the Log directory and the automatic cleanup options, see the Maintenance tab of the "Configure Local Server" screen, which is located within the Administrative Console.

## 13.2 Proper start/stop of web services

In multiple-machine web farms, it is possible to start/stop the 7i services on a particular server with minimal disruption to active users. The proper procedure is

Open a command window and type: `wlbs drainstop`

Open the Task Manager (right-click on the Taskbar, select Task Manager) and select the "Processes Tab" and click on the "Image Name" header to sort by Name

Wait until the CPU usage of the BTMQHOST process reaches 0% for at least 30 seconds. Also check the CDD QueueStatus page (see below) to make sure no active CDD reports are running.

In the command window, type: `net stop btnetsvc`

In the command window, type: `net start btnetsvc`

In the command window, type: `wlbs start`

Microsoft has acknowledged some problems with the 'drainstop' command never finishing – normally, it should complete in less than a minute. If not, after Step 4, you should restart the machine.

## 13.3 Service Packs

Service packs generally address numerous issues. It is always best to check and see if a problem you are having has been addressed on a released Service Pack before trouble shooting. Service Packs may require updates to both the host and Win server platforms.

## 13.4 Database Issues

All of the web-based applications do not react well if database service is interrupted. Recent Service Packs have improved this in some areas, but: When an unexpected interruption in database service occurs the appropriate services should be stopped and then restarted (or 'bounced'). Often, databases are brought offline for backup purposes. If converting to a 'hot' backup solution is not feasible, the Task Scheduler in Windows should be used to stop the services before the backup and start them again once the database is back online.

Anything that will adversely impact the performance of the database server (orphaned process, high CPU usage, excessive memory pressure) will directly impact 7i. If these issues exist, they should be addressed first.

Problems related to lack of disk space can also cause temporary database outages. A UNIX shell script named 'dbstat' provides an easy way to retrieve database statistics. The following are suggested commands:

Use 'dbstat -d' to show the amount of permanent disk space available.

Use 'dbstat -f' to identify fragmented database segments

Use 'dbstat -k' to identify either 'long' or 'deadly-embrace' locks. For long term monitoring, try 'dbstat -n 30 -k' which will provide updates every 30 seconds.

If running against SQL Server you may want to use those tools provided by Microsoft to review database statistics.

## 13.5 OpenLink

### 13.5.1 Version 4.2

OpenLink 3.2 does not appear to gracefully support the complex multi-threaded database access common in 7i so an upgrade to 4.2 is essential. Care must be taken that 7i actually uses the new ODBC driver once it is installed. Run the System Administrator, and select from the menu Connection/Connection Manager, and then Edit/Driver Selection. Ensure the 4.2 Driver is selected. Then, under Configure/Ifas Server Configuration, make sure the DRIVER= portion of the connect string references the proper 4.2 Driver. This must be repeated on each Server.

## 13.5.2 Multi vs Single-Threaded Agents

OpenLink 4.2 supports multi-threaded Agents – in some cases this has led to instability problems in which case the broker should be reconfigured for the single-threaded Agent. If OpenLink stability problems continue, ensure that ‘reuse=never’ is set in oplrqb.ini. **Note that if OpenLink is running fine, the above changes are not necessary.**

## 13.5.3 ODBC Tracing

In the ODBC Control Panel (it may be buried under Administrative Tools), select the Tracing tab. The button on the left should say “Start Tracing”. This means it is currently disabled and is the desired setting.

## 13.5.4 Receive Timeout

Using RegEdit, check the following registry setting:

HKEY\_LOCAL\_MACHINE\Software\ODBC\OPENLINK.INI\Communications

The value for ReceiveTimeout (which defaults to 120) should be at least 3600. SunGard’s OpenLink installation sets this value to 3600 but the standard OpenLink install does not.

## 13.6 Do’s and Don’ts

### 13.6.1 Do’s

Regularly examine the EventLog. CDD, for instance, will log those Reports that could not be run and the reason why, Workflow will identify problems with incoming or active instances, 7i will log those queries or business rules that took a ‘long’ time.

Check the CDD Queue Status page for detailed information on what CDD reports are running:

**<http://servername/ifa7/cdd/screens/queuestatus.asp>** (*<http://servername/ifa7/cdd/screens/queuestatus.asp>*)  
(<http://servername/ifa7/cdd/screens/queuestatus.asp>).

Check the Farm Status Page at **<http://servername/ifa7/admin/ServerStatus.htm>** (*<http://servername/ifa7/admin/serverstatus.htm>*)  
(<http://servername/ifa7/admin/serverstatus.htm>) (Requires 7.3 SP3 or later)

Users running CDD.Net reports should upgrade to Internet Explorer 6.0. Not only is it 3x faster than IE 5.5 in some areas, but it also supports proper page switching from portrait to landscape.

Users viewing Images (either through CDD.Net or 7i) should also upgrade to Internet Explorer 6.0 as the 5.5 version has some DirectX problems.

Ensure the Farm servers are listed in the IE Trusted Sites area. This works around a Microsoft security patch that prohibits cross-domain references.

Users that regularly run large CDD.Net Reports (> 1000 pages) should consider installing CDD directly on the workstation. Note: We have seen CDD.Net print 30,000 page reports, so it is doable.

For better uptime, ensure the both primary and secondary DNS servers are configured on each server.

In a Farm environment, the Workflow service should be enabled on only one machine in the Farm.

Microsoft introduced a problem in IE 6 SP 1 that may cause the 7i browser to hang or show a white screen. See Knowledge Base: 4320 for more information. Note that later security hotfixes are not compatible with this fix. Microsoft is working on this issue.

Sites using a proxy server **and** have Affinity set to 'Single' for the 7i farm should ensure that the 7i users bypass the proxy. 'Single' affinity will hash on the IP address to determine which server should process the request, and since all requests would come from the same IP (the proxy), all requests would therefore land on one server.

## 13.6.2 Don'ts

Do not Rebuild Security every time there is a problem. The only time a Rebuild of security is required is when security has changed, and excessive rebuilds can cause performance problems.

Do not enable IIS Compression, as it will cause random script errors in IE 6.0

## 13.7 Web Server Settings for Performance

Following are a few miscellaneous tips on settings that can help performance:

Examine the registry setting, HKEY\_LOCAL\_MACHINE\Software\BITECH\Report Processing Service for the value of FileStorageCleanup. This interval, measured in minutes, is how often CDD will cleanup its temporary files, the default is 60. If you regularly bounce the 7i services every day (e.g., for backups), ensure this value is less than the typical bounce interval.

Examine the registry setting, HKEY\_LOCAL\_MACHINE\Software\BITECH\Common\Trace and ensure that the Enable value is set to 0 (Zero).

Using the IIS Administrator, look in the /ifas7 (or /netsight7) virtual root and disable logging for all directories except bin.

If the Server is hosting either Employee Online or TimeOnline, the Server Affinity in the TCP/IP Network Load Balancing Settings should be set to 'Single', otherwise it should be set to 'None'.

Ensure the Network card (and switch and ...) is operational at 100Mb/sec FULL DUPLEX. In one case, correcting this improved the time for a query on POU\_USER\_MSTR from 1.5 seconds to 0.33.

## 14 Customizations

There are many customizations that can be made to the system to affect the behavior and appearance of any given screen. These customizations range from the definition of coded values to custom ‘after field’ rules. Likewise, implementation of some of these customizations requires a range of technical skill.

There are many fields within the system that are presented on data entry screens as combo boxes (picklists). The values presented in these lists are either generated from a static list that the end user has no control over (things like Active, Inactive), or are generated from querying a table that contains potential values. When a list is added to or removed from, the dynamic nature of these lists is automatically handled by the system. There are many such dynamic features that present themselves through standard features that react to everyday changes to the systems and their configuration.

### 14.1 Custom Business Rules

In this settings directory, additional xml and VB Script files can be created to contain business rule definitions and any associated VB Script. By naming the business rule file tableName.xml, the file will be processed when corresponding screens are compiled, and the settings will become active.


The XML example that follows shows four things:

How to setup a custom script to initialize new records presented on a 7i screen.

How to setup a custom script

How to setup a custom script to execute each time a record is added or updated from a 7i screen.

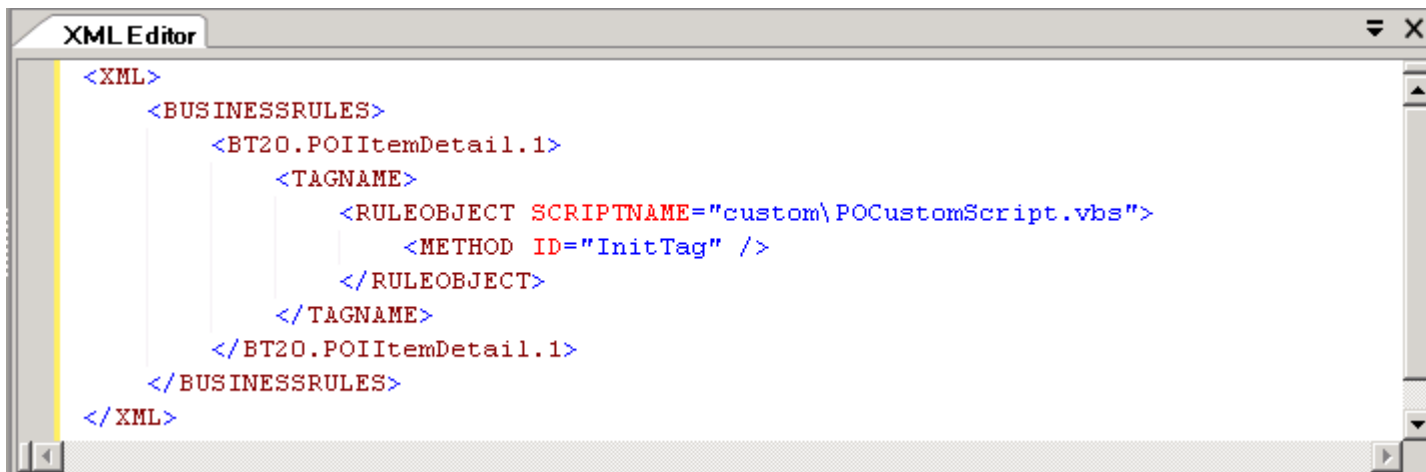
How to change a field to be disabled.



```
<XML>
  <BUSINESSRULES>
    <BT20.POPPvDetail.1>
      <INITNEW>
        <RULEOBJECT SCRIPTLOCATION="Settings" SCRIPTNAME="POPPvDetail.vbs">
          <METHOD ID="POPPvDetail_CustomInit" />
        </RULEOBJECT>
      </INITNEW>
      <INITTAG>
        <RULEOBJECT SCRIPTLOCATION="Settings" SCRIPTNAME="POPPvDetail.vbs">
          <METHOD ID="POPPvDetail_CustomInitTag" />
        </RULEOBJECT>
      </INITTAG>
      <PREACCEPT>
        <RULEOBJECT SCRIPTLOCATION="Settings" SCRIPTNAME="POPPvDetail.vbs">
          <METHOD ID="POPPvDetail_CheckOurRequirements" />
          <METHOD ID="POPPvDetail_RequireEndUse" />
        </RULEOBJECT>
      </PREACCEPT>
      <SetControlProperties>
        <POPReqCodes01 Init="1" Enabled='false' />
        <POPReqCodes01 Qbe="1" Enabled='true' />
      </SetControlProperties>
    </BT20.POPPvDetail.1>
  </BUSINESSRULES>
</XML>
```

In the above example, PREACCEPT must be in all upper case, and SetControlProperties must be in mixed case.

In the following, the default location of these scripts is assumed, as there is no SCRIPTLOCATION given. This default location is the ifas7\Scripts directory. In this example, the script will be found in ifas7\Scripts\custom.



```
<XML>
  <BUSINESSRULES>
    <BT20.POIItemDetail.1>
      <TAGNAME>
        <RULEOBJECT SCRIPTNAME="custom\POCustomScript.vbs">
          <METHOD ID="InitTag" />
        </RULEOBJECT>
      </TAGNAME>
    </BT20.POIItemDetail.1>
  </BUSINESSRULES>
</XML>
```

## 14.2 Server VB Script

The VB Script used to implement any custom business rules can reside in a couple of different places. One is the ifas7/connections/connectionName/settings directory, and another is the ifas7/scripts/custom directory. The definition in the business rules xml file will determine where we look for the VB Script file. For examples of that setup, see the previous section, Server XML.

There is a script object named ifas that is available to these custom scripted 7i business rules. This object has properties and methods that can be used by the script. These properties and methods are enumerated below:

### Objects:

**ifas.BT20Object.** This provides you access to the current record being processed. ifas.CurrentRecord provides the access to the same object. The two are equivalent.

**ifas.BT20ParentObject.** This provides you access to the parent of the current record being processed.

**ifas.BT20GrandParentObject.** This provides you access to the grandparent of the current record being processed.

**ifas.LastAddedRecord.** This provides you access to the last record added during the current session.

**ifas.CurrentRecord.** This provides you access to the current record being processed. ifas.BT20Object provides the access to the same object. The two are equivalent.



**ifas.CurrentLedger.** This provides you access to the CurrentLedgers object for the currently active ledgers. Of primary importance here are the ledger codes which can be accessed through:

**ifas.CurrentLedger.GILedger**

**ifas.CurrentLedger.JILedger.**

#### Values:

**ifas.AddMode.** Has a value of 1 when the business rule is being called while the screen is adding a new record. Has a value of 0 when the business rule is being called while the screen is updating an existing record.

**ifas.User.** Use to access the user id of the currently logged in user.

**ifas.IFASVersion.** Use to access to the version of IFAS that is currently running on the 7i server. For instance, 7.3.4.376.

**ifas.Location.** Use to access to the Location code in the current users Nucleus user definition.

**ifas.Manager.** Use to access to the Manager code in the current users Nucleus user definition.

**ifas.JobNo.** Use to access to the Nucleus Job Number associated with the users current session.

**ifas.ClientId.** Use to access the Client Id as defined in the NUUPAU section of Nucleus.

**ifas.CurrentScreenMask.** Use to access screen mask of the screen associated with a particular call to a business rule.

#### Methods:

**ifas.AddErrorMessage.** Use this method to cause an error message to be generated from a business rule script. The error message generated from this call will access both the standard error catalog and any custom error catalogs configured in the system. There are two parameters to this method, **section** (Section in the error catalog) and **number** (Error Number within the section).

**ifas.DebugTrace.** Call this method to have a custom business rule output tracing information available on the client machine when tracing is enabled using Shift-Ctrl-E.

**ifas.TranslateText.** Call this method to have text translated for you. The return value is the translated string. Supported translations follow:

\CDT. Current Date in MM/DD/YYYY format

\CD8. Current Date in YYYYMMDD format

\CTY. Current Date in YYYY format

\CDM. Current Time in HHMMSS format

\USER. Current User ID.

\GDTFYB. Fiscal Year Begin.

\GDTFYE. Fiscal Year End

\GDTQTB. Quarter Begin

\GDTQTE. Quarter End

\GDTCYB. Calendar Year Begin

\GDTCYE. Calendar Year End

\GDTMOB. Month Begin

\GDTMOE Month End

\GDTCYB Calendar Year Begin

These date related mnemonics can have calculations applied. For example, \GDTFYB+2. G – Means GL, J for JL. DT – Means Date. FYB – Means Fiscal Year Begin. +2 Means add two to the fiscal year. So, GDTFYB+2 gives you the fiscal year beginning date two years from now.

**ifas.GetBackgroundPart.** Call this method to obtain a background part for a specific Org Key or Object Code. There are four parameters to this method:

PartName – Name of the part desired

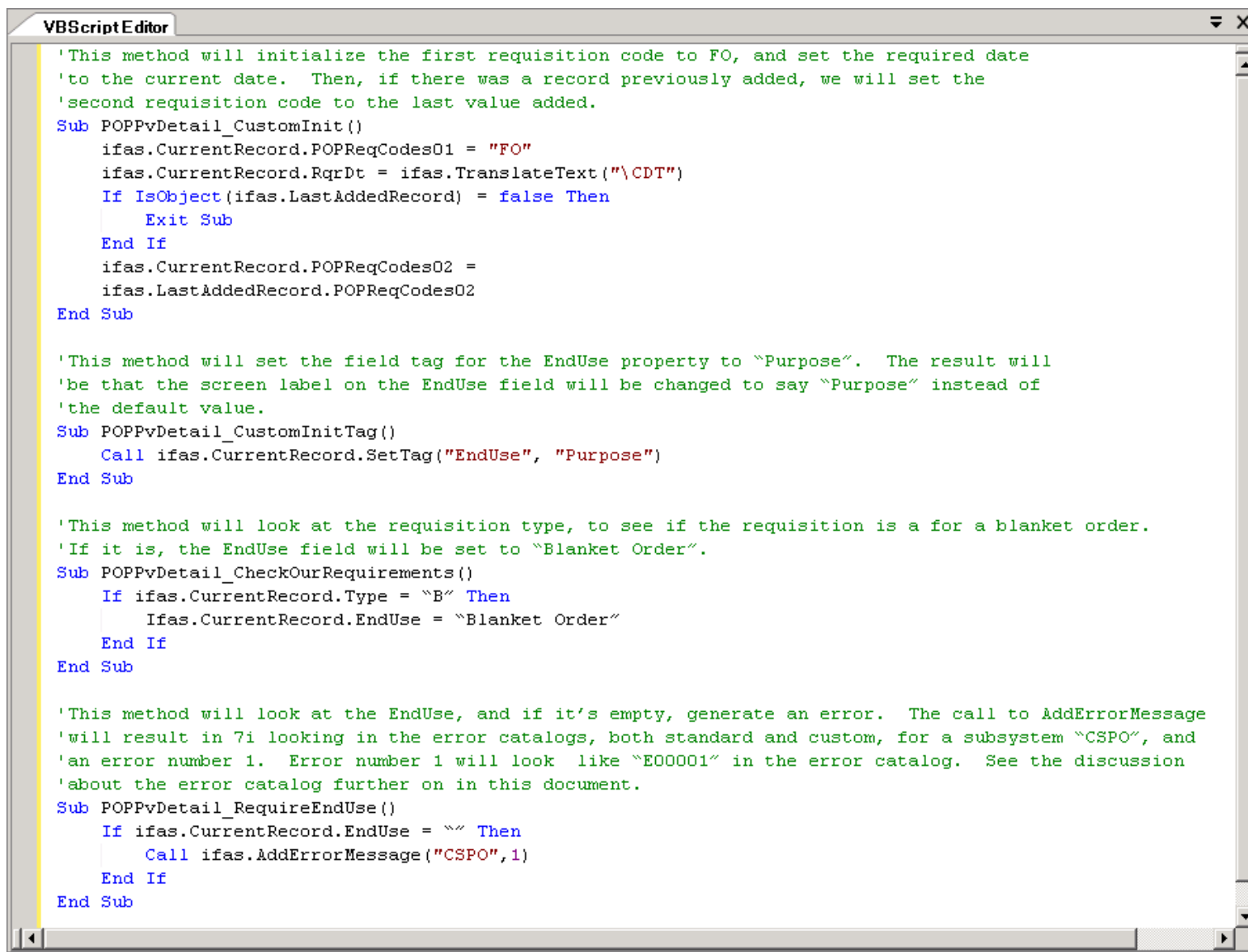
Ledger – Ledger Code

Key – Org Key

## Obj – Object Code

The PartName can be either an Org Key part name or an Object Code part name. The part value returned will be an Org Key part value or an Object Code part value based on the PartName passed in.

The XML examples in the previous section referenced a script named POPPvDetail.vbs. Examples of the methods contained within this script follow.



```
VBScript Editor
' This method will initialize the first requisition code to FO, and set the required date
' to the current date. Then, if there was a record previously added, we will set the
' second requisition code to the last value added.
Sub POPPvDetail_CustomInit()
    ifas.CurrentRecord.POPReqCodes01 = "FO"
    ifas.CurrentRecord.RqrDt = ifas.TranslateText("\CDT")
    If IsObject(ifas.LastAddedRecord) = false Then
        Exit Sub
    End If
    ifas.CurrentRecord.POPReqCodes02 =
    ifas.LastAddedRecord.POPReqCodes02
End Sub

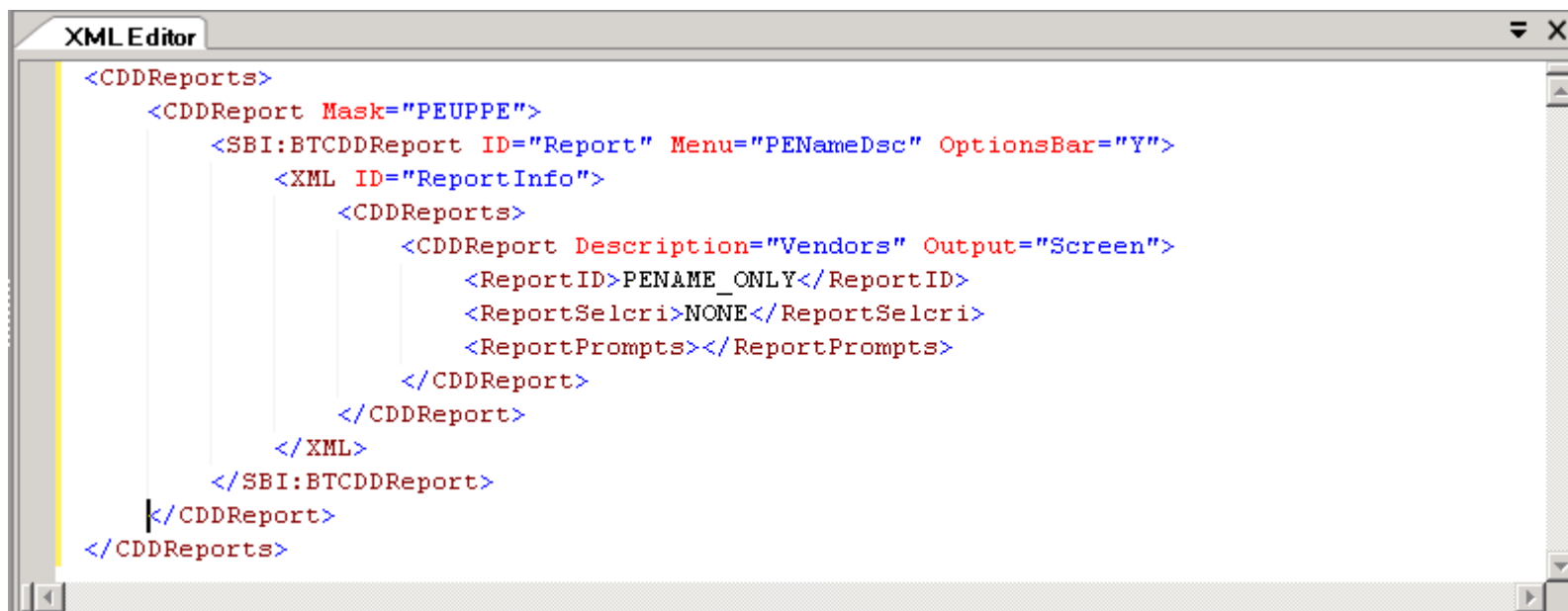
' This method will set the field tag for the EndUse property to "Purpose". The result will
' be that the screen label on the EndUse field will be changed to say "Purpose" instead of
' the default value.
Sub POPPvDetail_CustomInitTag()
    Call ifas.CurrentRecord.SetTag("EndUse", "Purpose")
End Sub

' This method will look at the requisition type, to see if the requisition is a for a blanket order.
' If it is, the EndUse field will be set to "Blanket Order".
Sub POPPvDetail_CheckOurRequirements()
    If ifas.CurrentRecord.Type = "B" Then
        Ifas.CurrentRecord.EndUse = "Blanket Order"
    End If
End Sub

' This method will look at the EndUse, and if it's empty, generate an error. The call to AddErrorMessage
' will result in 7i looking in the error catalogs, both standard and custom, for a subsystem "CSPO", and
' an error number 1. Error number 1 will look like "E00001" in the error catalog. See the discussion
' about the error catalog further on in this document.
Sub POPPvDetail_RequireEndUse()
    If ifas.CurrentRecord.EndUse = "" Then
        Call ifas.AddErrorMessage("CSPO",1)
    End If
End Sub
```

## 14.3 Adding Reports to Screens

7i data entry screens can be configured to present an option to execute one or more CDD reports. On the 7i server a file named CDDReports.xml is used to configure which CDD reports will be available on any given data entry screen. With most installations, this file can be found in c:\inetpub\wwwroot\ifas7\xml\cddreports.xml. The following is an example of the type of XML that can be added to this file to have the report named “PENNAME\_ONLY” appear on the PEUPPE 7i data entry screen with a title of “Vendors”. This example is a report with no specific criteria; there will be no specific runtime criteria applied.



```
<CDDReports>
  <CDDReport Mask="PEUPPE">
    <SBI:BTCDDReport ID="Report" Menu="PENNameDsc" OptionsBar="Y">
      <XML ID="ReportInfo">
        <CDDReports>
          <CDDReport Description="Vendors" Output="Screen">
            <ReportID>PENNAME_ONLY</ReportID>
            <ReportSelcri>NONE</ReportSelcri>
            <ReportPrompts></ReportPrompts>
          </CDDReport>
        </CDDReport>
      </XML>
    </SBI:BTCDDReport>
  </CDDReport>
</CDDReports>
```

Adding report links that include runtime selection criteria are much more involved, and require technical assistance from SunGard Public Sector Bi-Tech.

Modifications to CDDReports.xml will be overwritten on each update or service pack. Take appropriate action to re-apply desired changes after each update.

## 14.4 Adding Screen Links to Screens

The Screen Designer can be used to add screen links to a data entry screen. If you plan to customize a data entry screen in this way, it is strongly recommended that you start by making a client specific copy of the data entry screen, and add the screen links to that version.

With the Screen Designer, on the Insert Menu, Control option, ScreenLinks is listed. Choose ScreenLinks and a ScreenLinks structure will be added to the screen definition. To configure ScreenLinks, Controls Browser, open Controls, then Forms, then MainForm. At the bottom of MainForm, you will see a ScreenLinks element. Click on the ScreenLinks element and the control properties for this element will be presented. In the control properties, there is a New Screen Link option. If the presented properties are already filled out, you can click on the New option, to add an additional screen link definition. To create the screen link definition, the following needs to be filled out:

**Screen:** URL to the screen you want to configure for this linkage. When you configure this, remember that the current directory will be the screens directory for the screen you are working with. For instance, if you are working on a BudgetItemDetail screen, and wish to add a link to the General Ledger Org Key definition screen, what you would enter here is “../GeneralLedger/GLUPGN.asp”.

**Desc:** Enter the description you wish to appear on the data entry screen to represent this linkage.

**Linkage:** You can add multiple linkage elements using the New Linkage Element option. When providing a linkage element, you have to know something about the screen definition you are working with, and the screen definition you are linking to. Linkage information does not have to be provided, but it is with the linkage definition that you can configure a linkage to link from the currently presented record on the screen you are working with, to the corresponding record or records on the screen you are linking to.

**From: DataSourceName:** Enter the name of the Data Source in the current screen definition that you wish to use for source data information.

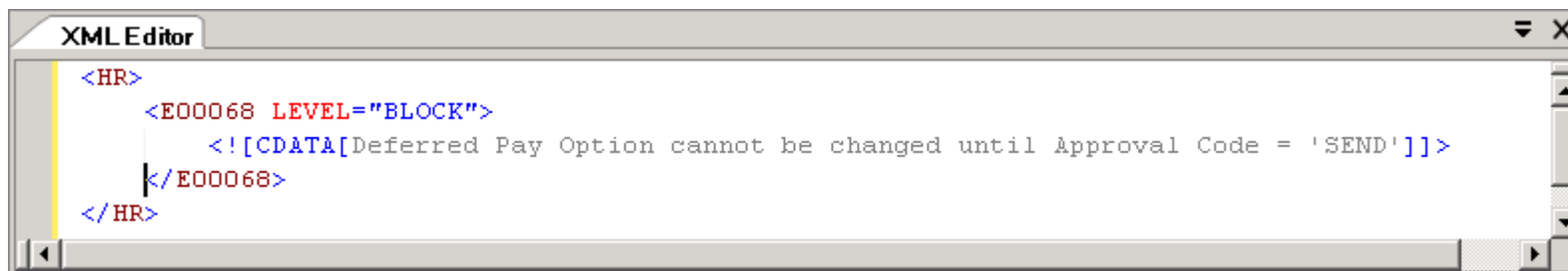
**From: DataSourceProperty:** Enter the name of the property in the from data source that you wish to provide to the new screen.

**To: DataSourceName:** Enter the name of the Data Source in the screen you are linking to that you wish to provide initial QBE information for.

**To: DataSourceProperty:** Enter the name of the property of the Data Source in the screen you are linking to that you wish to provide initial QBE information for.

## 14.5 Error Catalog

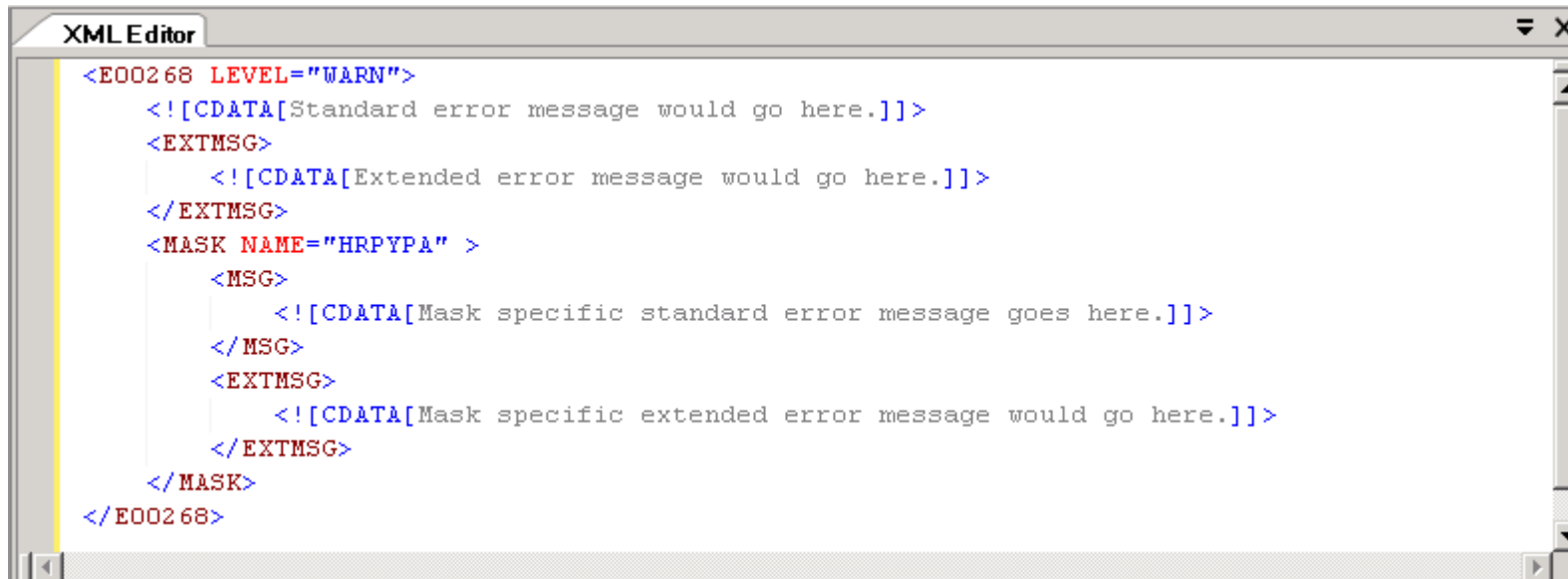
The standard error catalog is found in ifas7/xml/errorcat.xml. This error catalog should not be modified. Custom error catalogs are supported. The standard errorcat syntax is as follows:



```
XML Editor
<HR>
  <E00068 LEVEL="BLOCK">
    <![CDATA[Deferred Pay Option cannot be changed until Approval Code = 'SEND']]>
  </E00068>
</HR>
```

<HR> represents the HR subsystem. With HR, there are multiple errors defined. In this example, error 68 is defined. The format of the error number must start with 'E', and be 5 digits long. The 000 preceding 68 are necessary to fulfill the 5-digit requirement.

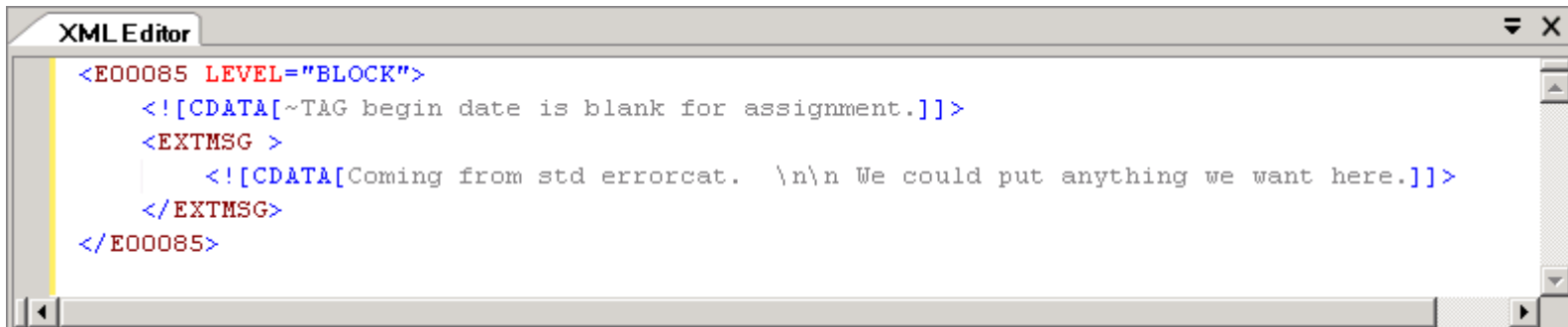
This syntax has been extended to accommodate custom errors as follows:



```
XML Editor
<E00268 LEVEL="WARN">
  <![CDATA[Standard error message would go here.]]>
  <EXTMSG>
    <![CDATA[Extended error message would go here.]]>
  </EXTMSG>
  <MASK NAME="HRPYPA" >
    <MSG>
      <![CDATA[Mask specific standard error message goes here.]]>
    </MSG>
    <EXTMSG>
      <![CDATA[Mask specific extended error message would go here.]]>
    </EXTMSG>
  </MASK>
</E00268>
```

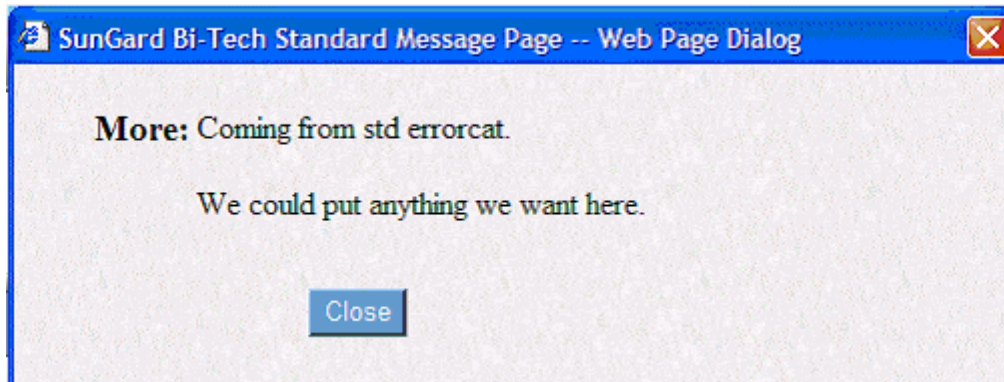
The <EXTMSG> node can hold an extended message to be displayed in a separate window. The <MASK> node can be used to define error messages specific to a particular mask. The <MSG> node is optional and can be used to denote the standard message either under the <E00xx> node or the <MASK> node. If <MSG> is not found then the CDATA element is taken as the standard message.

When an error has an extended message the user will see ‘...more’ after the standard message in the 7i Nav Bar. The default behavior when clicking on ‘...more’ will open a modeless dialog window displaying the extended messages. Here’s a simple example:



```
<E00085 LEVEL="BLOCK">
  <![CDATA[~TAG begin date is blank for assignment.]]>
  <EXTMSG >
    <![CDATA[Coming from std errorcat. \n\n We could put anything we want here.]]>
  </EXTMSG>
</E00085>
```

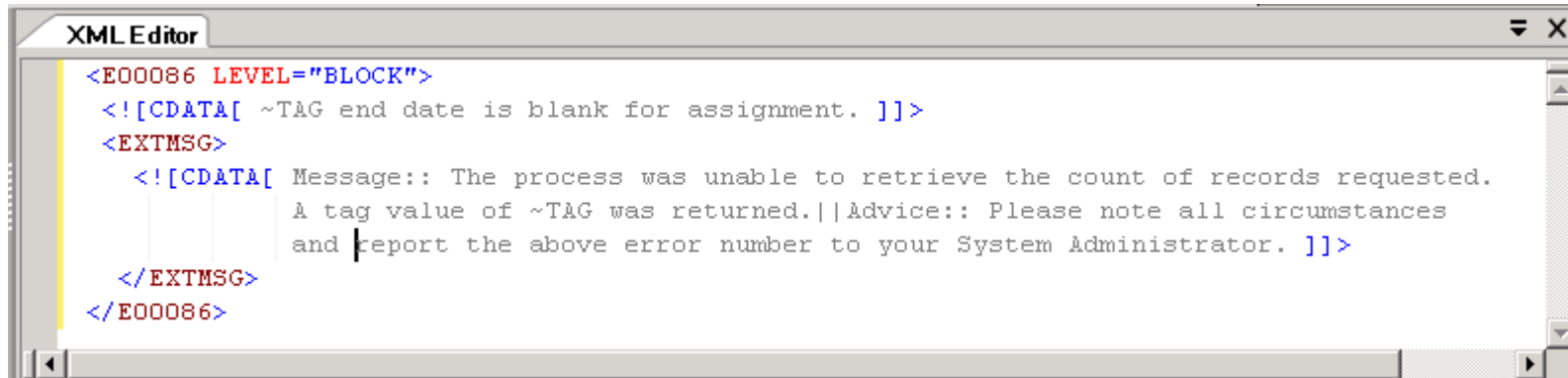
Which looks like this.



Notice that it recognized ‘\n’ as a new line and defaulted a title of More.

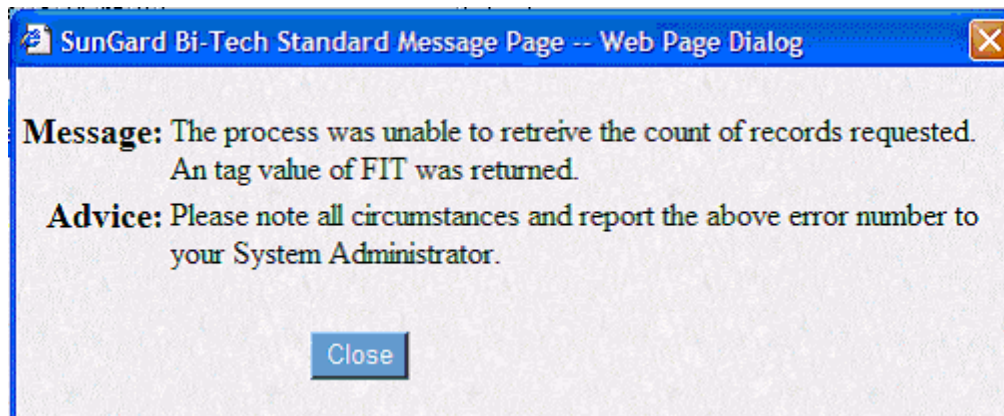


Here's another example which a message and some advice:



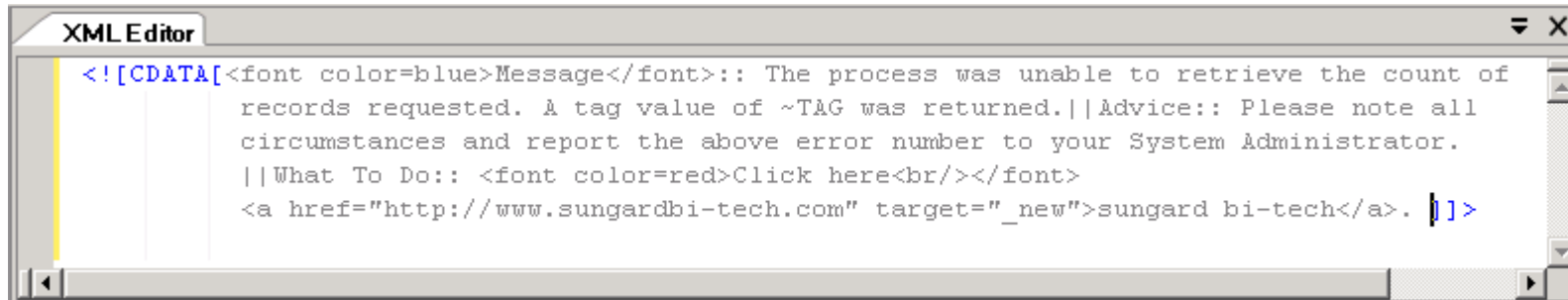
```
<E00086 LEVEL="BLOCK">
  <![CDATA[ ~TAG end date is blank for assignment. ]]>
  <EXTMSG>
    <![CDATA[ Message:: The process was unable to retrieve the count of records requested.
      A tag value of ~TAG was returned.||Advice:: Please note all circumstances
      and report the above error number to your System Administrator. ]]>
  </EXTMSG>
</E00086>
```

The displayed message looks like this:



Notice that the ~TAG replacement occurs in extended messages just like the standard message. There are several syntax conventions that help format the default display of extended messages. Use ‘||’ to denote a new paragraph. Use ‘::’ to denote a title.

Since we're displaying in a browser window that recognizes html it is also possible to use html tags to further enhance the extended message.



```
<![CDATA[<font color=blue>Message</font>:: The process was unable to retrieve the count of
records requested. A tag value of ~TAG was returned.||Advice:: Please note all
circumstances and report the above error number to your System Administrator.
||What To Do:: <font color=red>Click here<br/></font>
<a href="http://www.sungardbi-tech.com" target="_new">sungard bi-tech</a>. ]]>
```

Which results in the following:

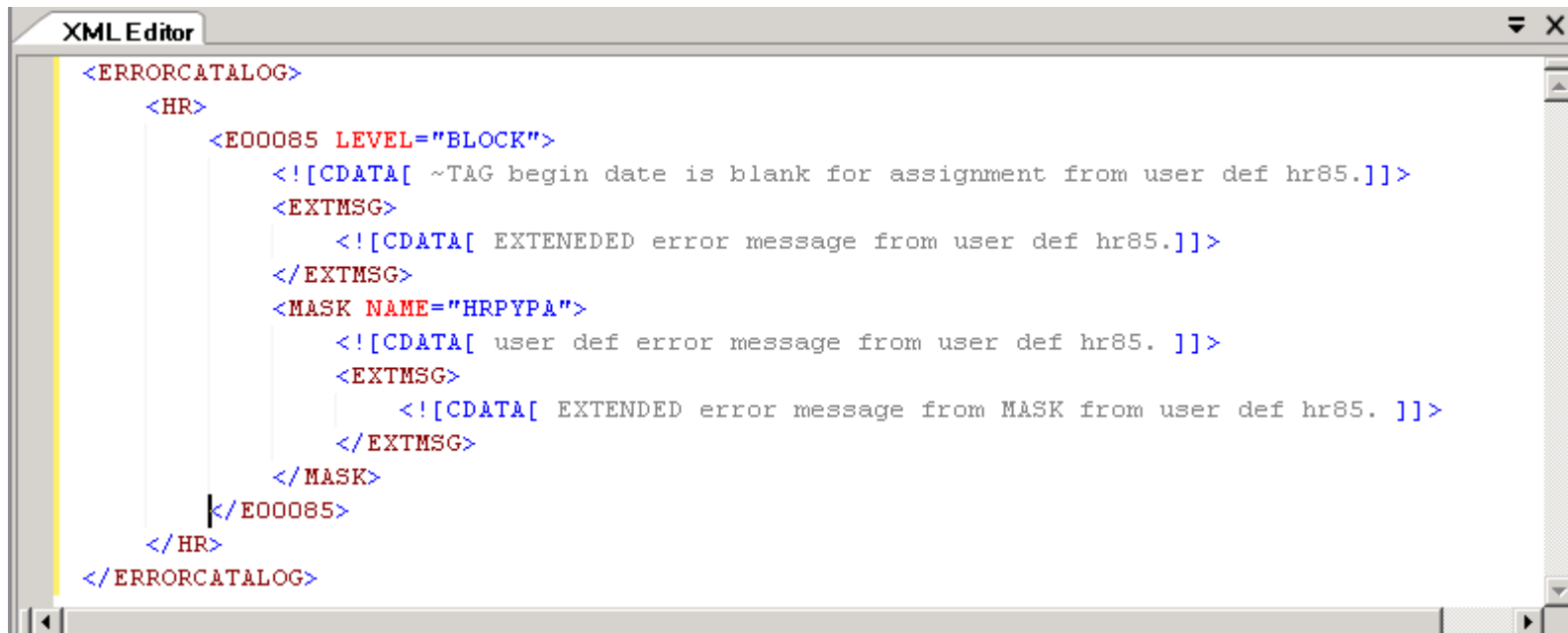


If something stronger than the standard display of extended messages is needed it is possible to create a web page specific to an error. Such pages should be named by concatenating the subsystem with the error number followed by .asp or .htm. In the case of hr error E00068 an extended message file might be named hre00068.asp. The window that opens this page will be passed any text found in the <EXTMSG> node and can be referenced in the page as window.dialogArguments. Standard SunGard Public Sector Bi-Tech extended message pages are located in ifas7/MessagePages. The default display file is named StdMsgs.htm.

When defining custom extended error messages, they should be named using the same naming convention and placed under the ifas7/MessagePages/Custom directory. For these custom pages to be used they must be defined in a custom errorcat containing entries for those error numbers which are to be overridden.

The location of the standard errorcat.xml file is the ifas7/xml directory. This file is the one provided by SunGard Public Sector Bi-Tech and should not be modified by clients. Standard error messages may be overridden by a custom errorcat.xml which should be placed in the ifas7/xml/custom directory. This file should use the xml syntax found in the standard errorcat but only include those error numbers that are to be customized. The 7i infrastructure will check this custom location first when it encounters a particular error. If an entry is found then the custom error message will be displayed. If a custom extended message is found then clicking '...more' will display a custom extended message page. If no custom page is found then the custom extended message is displayed using the standard display.

A very simple custom errorcat could look like this:



```
<ERRORCATALOG>
  <HR>
    <E00085 LEVEL="BLOCK">
      <![CDATA[ ~TAG begin date is blank for assignment from user def hr85.]]>
      <EXTMSG>
        <![CDATA[ EXTENEDED error message from user def hr85.]]>
      </EXTMSG>
      <MASK NAME="HRPYPA">
        <![CDATA[ user def error message from user def hr85. ]]>
        <EXTMSG>
          <![CDATA[ EXTENDED error message from MASK from user def hr85. ]]>
        </EXTMSG>
      </MASK>
    </E00085>
  </HR>
</ERRORCATALOG>
```

This would override HR error E00085 with a client specific message and extended message. It would provide different messages when the error occurred on the HRPYPA screen. To display the extended message the system would look for a file HRE00085.asp (or .htm) under ifas7/MessagePages/Custom. If no page was found the extended message would be displayed using ifas7/MessagePages/StdMsgs.htm.

One more note regarding the LEVEL attribute. It is possible for the to change the level in the custom error cat. However, the error level should never be changed to a lower level as this might play havoc with the business rules associated with a screen. Thus an error that was a WARN in the standard error cat could be changed to a BLOCK in the user defined error cat if the client so desired. However a BLOCK in the standard errorcat should never be changed to a WARN in the user defined errorcat.

## 14.6 Custom Tab Orders in 7i

Previously tab order was controlled by the order in which the controls appeared in the screen .asp page which is created during screencompile. Usually this order just represented the order that the controls occurred in our xml definition file. You could explicitly manage the order by setting the <TabOrder> element on a control to a number. This number was used by screen compile to sort the elements as they were placed into the screen .asp page.

Custom Tab Order leaves the old tab method in place but provides a way to define additional tab orders that a user can choose to activate at any time while working on a screen. It also allows for the use of default values which automatically populate fields when a Tab Order is in place. It is hoped that the capabilities provided by Custom Tab Orders will greatly increase the ease with which users can do certain activities on 7i screens.

These Tab Orders are defined in an xml file which lives on the 7i server. The naming convention for the file is <ScreenMask>TabOrderdefs.xml. We support a standard version of this file in which SunGard Public Sector Bi-Tech can define TabOrders that we think are useful. If it exists it will be located in <ifas7root>/screens/<moduleName>/. We also allow clients to define site specific Tab Orders. These site specific Tab Orders should be placed in <ifas7root>/Custom/screens/<moduleName>/.

To make the naming conventions concrete, using our development directory structure on my box, the standard Custom Tab Order for the TDHREMEN screen would be found in netsight7screens/screens/humanresources/TDHREMENTabOrderDefs.xml. The client specific Tab Order would be in netsight7screens/screens/humanresources/TDHREMENTabOrderDefs.xml.

If Custom Tab Orders are defined on a screen they can be selected via the Options Menu drop down. The first entry, labeled Default Tab Order, represents the old tab behavior that exists on the screen now. The other entries represent Custom Tab Order choices that have been defined for different tasks or users of this screen. An accelerator key, ctrl-o (for Order), can be used to toggle between a Custom Tab Order and the Default Tab Order. This provides a quick way of getting into and out of a Custom Tab Order.

Here's what a standard Custom Tab Order definition might look like.

## XML Editor

```
<sbixml>
  <TabOrderDefs>
    <TabOrderMask mask="ARBTCRIC">
      <TabOrderDef id="_tab1" desc="Back Order Parking Tickets $7.00" ifasUser="@">
        <Tabs>
          <Tab elementId="BTDirectCashCRController_Desc_BTEditControl" firstElement="Y" />
          <Tab elementId="BTDirectCashCRController_Price_BTEditControl" />
          <Tab elementId="BTPaymentCRController_Amt_BTEditControl" />
        </Tabs>
        <RecordDefaults />
      </TabOrderDef>
      <TabOrderDef id=" tab2" desc="Back Order Parking Tickets" ifasUser="TGM">
        <Tabs>
          <Tab elementId="BTDirectCashCRController_Desc_BTEditControl"
            setRecordDefaults="Y" firstElement="Y" />
          <Tab elementId="BTDirectCashCRController_Price_BTEditControl" />
          <Tab elementId="ARBirMasterCrippledD3C_Ref_BTEditControl" />
        </Tabs>
        <RecordDefaults>
          <Default elementId="BTDirectCashCRController__FinCd_BTComboBox" defaultValue="PARKING" />
          <Default elementId="BTPaymentCRController_Type_BTEditControl" defaultValue="CK" />
        </RecordDefaults>
      </TabOrderDef>
      <TabOrderDef id="_tab3" desc="Back Order Water Payments" ifasUser="MAFY">
        <Tabs>
          <Tab elementId="PeId" />
          <Tab elementId="BTPaymentCRController_Amt_BTEditControl" />
        </Tabs>
        <RecordDefaults>
          <Default elementId="BTDirectCashCRController__FinCd_BTComboBox" defaultValue="FC01" />
          <Default elementId="BTPaymentCRController_Type_BTEditControl" defaultValue="CK" />
        </RecordDefaults>
      </TabOrderDef>
    </TabOrderMask>
  </TabOrderDefs>
</sbixml>
```

Here's a Node and attribute dictionary:

**<sbixml>** Root node, follows our standard convention.

**<TabOrderDefs>** **<TabOrderMask mask="ARBTCRIC">** The mask attribute should contain the actual mask for the screen in question.

**<TabOrderDef id="\_tab1" desc="Back Order Parking Tickets \$7.00" ifasUser="@">**

Represents a Custom Tab Order definition. These are the nodes that define the choices that the user will see from the Tab Order selection under the Options Menu on the nav bar. If this is from the standard SunGard Public Sector Bi-tech provided Custom Tab Order then the id attribute should be given a name starting with '\_'. For Client specific Tab Order definitions the id attribute should start with an alphanumeric. The user sees the values of the desc attribute in the Tab Order selection from the Options Menu. The ifasUser attribute should be set to '@' (*mailto:'@'* (*mailto:'@'*)) if this definition is to be visible for all users. To make the definition visible only for a particular user then set ifasUser to that user's ifas login name (in upper case).

**<Tab elementId="BTDirectCashCRController\_Desc\_BTEditControl" firstElement="Y" />**

The **<Tab>** node defines a screen element that participates in the Custom Tab Order. The elementId attribute should contain the id of the element which can be found either in the screen definition xml or the screen asp page generated by screencompile.js. Currently we only support a subset of screen elements that map to our HTC control definitions. We don't support custom tabbing to BTButton elements or tabStrips.

The order of the **<Tab>** elements within a definition provides the tab transitions that will be used unless other directives are present. In other words, the Custom Tab transitions will follow the order of the **<Tab>** elements in the xml file. When the user is on the last element in the Tab Definition and hits tab then the tab order will wrap around to the first **<Tab>** element. Shift Tab is also supported and works the transitions in reverse order.

Attributes on the **<Tab>** element can change the way Custom Tab transitions are handled. One such directive is the firstElement attribute. There should only be 1 firstElement defined for a given **<TabOrderDef>**. The presence of this attribute set to "Y" tells Custom Tab Order that focus should be initially set in this element after a screen navigation event (i.e., selecting at Custom Tab Order from the menu) pressing enter, or navigating to the next record.

**<Tab elementId="BTPaymentCRController\_Amt\_BTEditControl" nextElement="BTDirectCashCRController\_Desc\_BTEditControl" />**

Here's another example of a **<Tab>** node attribute that modifies the transitions that Custom Tab Order will provide. The nextElement attribute contains the element Id of the control to which we will tab from the present control. Usually this will be an element that participates in the

Custom Tab Order (exists as a <Tab> node within the definition). However it can be any arbitrary element on the screen. If the value of the nextElement attribute is 'default' then the underlying default tab order of the screen is used to determine the transition.

If a Custom Tab Order is in effect and the user is presently on a screen element that is not part of the Tab Order definition then the normal tab transition will be used. Once the focus lands on an element that participates in the Custom Tab Order then the custom transitions will be used.

```
<Tab elementId="BTDirectCashCRController_Account_BTEditControl" defaultValue="Y" forceDefault="Y" GIGr="GL" GIKey="101201"
GIObj="1301" JIGr="GL" JIKey="101211" JIObj="1311" Wo="4" Level="10"/>
```

This <Tab> element defines some default values that will be placed into the element when a Custom Tab Order transition results in focus on that control. Normally the defaultValue attribute would hold the value that you'd like to place into the control. This particular control is an EditAccount control which requires special handling, defaultValue is set to "Y" and the actual values to default are given by the following attributes: GIGr, GIKey, GIObj, JIGr, JIKey, JIObj, Wo, and Level. Normally the default values will only be populated when the underlying control is blank (or 0 in the case of numerics). Set the forceDefault attribute to "Y" if you'd like the default values to be populated even if non trivial values are all ready in place.

The phone control, with it's phone code and number is a special default case similar to the EditAccount control. For the phone control set defaultValue to "Y" and then use attributes PhoneCode and/or PhoneNumber to define your defaults.

```
<Tab elementId="BTDirectCashCRController_Desc_BTEditControl" setRecordDefaults="Y" firstElement="Y" />
```

Default values are usually populated when tabbing into a control. If you would like a variety of fields to receive default values all at one time then you can use the setRecordDefaults attribute. When this is set to "Y" and a Custom Tab Order transition results in focus on this control then the Custom Tab Order logic looks for a <RecordDefaults> node. The children of this node define a set of elements to receive default values.

```
<RecordDefaults>
```

Children of this node define a set of elements to receive default values when setRecordDefaults is turned on for an element in the Custom Tab Order (see entry immediately above).

```
<Default elementId="BTDirectCashCRController__FinCd_BTComboBox" defaultValue="FC01"/>
```

Identifies an element to receive a default value. The actual default value is defined in the defaultValue attribute (just like in a Tab node). You can force the default value into elements that are non blank by setting forceDefault to "Y" (again just like the Tab node defaults).

Use caution when defining default values. Defining a default value on a Tab node is safest because it won't happen until you actually tab into the field. When you leave that field the afterfield rules will fire as expected with the new value in place. Defining record level defaults causes the default values to be placed in all the fields defined under the RecordDefaults node. The fields are marked as dirty and the value is placed into the control and it's related data source record if it exists. However, at present, no afterfield rules are fired for these Record Defaults.

Record defaults on custom screen controllers may have unexpected behaviors. Record Defaults should be thoroughly tested before doing real work on the screen.



## 15 Registry Settings

The registry settings that are used by the 7i services can be found in HKLM/Software/BiTech/Data Processing Service. Under this key, the following can be found. Some of these settings are only created if the default value is overridden for a particular setting.

Name	Type	Comment
Attachment Temp Directory	String	Location of the directory that is used as a temporary holding area for attachments that need to be presented to 7i users.
Catalog Connect	String	This is the ODBC connect string that 7i uses to establish it's connection to the system catalog. The Configure Local Server screen of the Admin Console application creates this registry setting.
Connection Names	String	A semi-colon separated string of connection names that 7i will accept connections for. The names given in this setting must be valid data sources configured in the system catalog configured in Catalog Connect.
Disable Database	DWORD	This setting will disable database connections for a particular 7i server. The default is 0
Disable Handle Pooling	DWORD	By default, the 7i services will use cursor pooling to minimize the number of times an SQL statement needs to be prepared. If this feature needs to be disabled, set to 1. The default is 0.
DisableAutoSort	DWORD	By default, the screen compile process looks for database indexes for the top level data sources on the creen. Sort options are then presented for each of these indexes. With some versions of Informix and Oracle, the performance of the screen compile process is poor when this option is enabled. The default is 0.

Handle Pooling Max	DWORD	When 7i is configured to use cursor pooling (Disable Handle Pooling above does not exist, or is set to 0), the Handle Pooling Max setting can be used to configure how many cursors will be held in the pool. When a pool becomes full, cursors are freed and removed from the pool when a new cursor needs to be added. The default is 64.
NetSight Root Path	String	This is the files system path of the directory holding the 7i web site.
Server Group	String	Each server in a 7i server farm should be configured with the same Server Group value. It is with this setting that we determine which servers are participating in a particular 7i server farm.

There are additional settings that you will see once the 7i Data Processing Services are up and running, but they are all configurable using the standard 7i Configuration pages discussed in the Remove Configuration portion of this document.

## 16 FAQs

### FAQ: HOW DO I PERFORM A CLEAN UNINSTALL OF PC PRODUCTS?

**Extended Text:** KB ID 2211

Add Remove Programs.

Remove all the Sungard Public Sector Bi-Tech applications.

Before performing the next step, export the registry to a file as a precaution. To do this go to start run and type, "regedit". See Figure 1.

Remove the registry key, HKEY\_LOCAL\_MACHINE\\Software\\BiTech. See Figure 2.

Remove the registry key, HKEY\_CURRENT\_USER\\Software\\BiTech. See Figure 3.

Delete the Bi-Tech Folder from the filesystem on the PC. Typically, this folder will be found directly under C:\

### FAQ: SOCKET BIND ERROR THROUGH CITRIX SERVER

**Details:** KB ID 3924

This K.B. explains how to clear up a socket bind error that occurs when you try to open up two insight sessions through the same Citrix server.

**Extended Text:**

1. Create a new Folder anywhere on the C: drive of the Citrix Server that is having the problem. This new folder should probably be named something like Citrix\_Temp.
2. Right click on this new folder and select Properties.
  - a. Select the "Security Tab"
  - b. Make sure that "Everyone" has modify privileges to this folder.

3. Open up Regedit (Start > Run > Regedit).
4. Add a new "String Value" under HKEY\_LOCAL\_MACHINE > Software > Bi-Tech > Insight. The name of this string value should be "Temp" and the data for this String Value should point to a folder on the C: drive that you just created.
5. The Citrix server should no longer get the socket bind error.

NOTE: It is wise to backup the system registry before making any changes.

## FAQ: "SQL SERVER DOES NOT EXIST OR ACCESS DENIED" ERROR

**Details:** KB ID 6117

The primary cause of this error on SQL Server is due to a large number of database calls initiated by the process and the sockets are not getting released. [DBNETLIB][CONNECTIONOPEN (CONNECT()).]SQL SERVER DOES NOT EXIST OR ACCESS DENIED

### Extended Text:

There are a couple of reasons why this error can occur:

1. The user does not have access to the SQL Server database.

First thing to check for is the DB\_SERVER variable in ifas.start.config file. The default location of this file is c:\usr\local\scripts unless specified a different location in .profile. Secondly, make sure that the IUSR\_IFAS is defined in both the database on the SQL Server and on the APP server. IUSR\_IFAS MUST have the same password on the SQL and APP servers. If you are not on SQL Server 2000 Service Pack 4 then apply it as it is a required service pack level for Ifas.

2. There are a large number of database calls initiated by the process and the sockets are not getting released.

To find out if the sockets are not getting released, open a command prompt and type "netstat -n" on the database server. This should list the local IP address of the server you are running the command on as well as a Foreign Address and State. Look for the connections with a State of TIME\_WAIT. If there are a number of connections in the TIME\_WAIT state you may need to adjust/create the MaxUserPort and TcpTimedWaitDelay registry entries on computers listed in the output's Foreign address. Generally these connections will be coming from your application server. There are two detailed KBs published on the Microsoft site:

<http://support.microsoft.com/kb/328476> (<http://support.microsoft.com/kb/328476>)

<http://support.microsoft.com/default.aspx?scid=kb;EN-US;319502> (<http://support.microsoft.com/default.aspx?scid=kb;en-us;319502>)

By default the MaxUserPort is set to 5000 and TcpTimedWaitDelay is set to 240.

We need to set them to 20000 and 30 respectively. Make sure the values set are defined as decimal instead of hexadecimal values. Here are the details:

Start registry editor:

Start --> Run --> regedit

Go to HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters

On the Edit menu, click Add Value, and then add the following registry value:

Value Name: MaxUserPort

Data Type: REG\_DWORD

Value: 20000 (decimal)

Do the same for TcpTimedWaitDelay:

Value Name: TcpTimedWaitDelay

Value Type: REG\_DWORD

Value: 30 (decimal)

Save and exit regedit. After these settings, you MUST restart Microsoft Windows for the new settings to take effect.

These need to set these even if you are using connection pooling. To see if connection pooling is enabled, go to Start --> Settings --> Control Panel --> Administrative Tools --> Data Sources (ODBC) --> Connection Pooling. Under the ODBC Drivers select SQL Server and click OK. If the "Pool Connections to this driver" radio button is enabled then connection pooling is on.

Two tasks, 335933 (763) and 322454 (774), are addressing this issue in Ifas on SQL. Both tasks require SQL Server 2005. SQL Server 2000 clients can use the registry key work around.

## FAQ: NAMING CONVENTIONS FOR CLIENT SPECIFIC DATABASE INDEXES?

**Details:** KB ID 6191

When creating client specific database indexes, it is advisable to follow a naming convention that includes using a prefix "cs\_". Not only does this convey the meaning that the index is Client Specific, it is also recognized by various database update processes as an index that should not be dropped. Also, like any computer data, a backup of the SQL that is used to create the index should be stored in back-up form that is not volatile with the state of the database. A text file that is copied to backup is a good option.

## FAQ: 7I SERVER MEMORY ALLOCATION ERROR "CANNOT ALLOCATE" IN THE EVENT VIEWER"

**Details:** KB ID 6425

(sample) Error Message in Event Viewer: btwebrqb: Cannot allocate '544' + 1 for pDataAvail 2 btwebrqb: Cannot allocate '360' for pRequestData

**Details:**

It seems that eventually the IIS process, in conjunction with the logic in btwebrqb.dll, runs into a problem allocating heap memory. It generally takes a lot of activity before the error is generated, but once it starts, it is fairly constant.

**Work Around:**

Edit registry key HKEY\_LOCAL\_MACHINE\SOFTWARE\bitech\Request Broker on the 7i server. Create new DWORD value called "UseBrokerPolicy" with a value of 0 (zero). Once this registry entry is created, you must shut down Data Processing Services, run IISRESET.exe, and restart Data Processing Services, respectively, on the 7i server (NOTE: For clients with a farm system, this process must be repeated on each server in the farm). You should also check the application logs after a request has been processed to check for an alert stating that Broker policy has been turned off.

This issue is being addressed in task 336323

## FAQ: CAN'T CONNECT TO INSIGHT WHEN LOGGED IN TO WINDOWS XP

### Question:

WHY CAN'T I CONNECT TO INSIGHT WHEN LOGGED IN TO WINDOWS XP WITH A USER WHO DOES NOT HAVE LOCAL ADMINISTRATIVE RIGHTS?

**Details:** KB ID 6628

After installing the PC-Products as a local administrator on a client PC, when a non-admin user logs on to the PC, the user cannot connect using Insight.

### Extended Text:

Non-admin users for XP who use Insight must have access to read and write to the following registry keys:

"HKEY\_LOCAL\_MACHINE/SOFTWARE/bitech"

"HKEY\_CURRENT\_USER/Software/BiTech"

"HKEY\_LOCAL\_MACHINE/SOFTWARE/Openlink Software" if Openlink is in use (for non SQL server clients)

"HKEY\_LOCAL\_MACHINE/SOFTWARE/Odbc/odbc.ini"

"HKEY\_LOCAL\_MACHINE/SOFTWARE/Odbc/openlink.ini"

Users should also have write access to the directory where the PC-products were installed (typically C:\Program Files\Sungard Bi-Tech)

Non-administrator users in Windows XP will need access to the above keys and directories.

## FAQ: ADVSETUP -- HOW TO USE THE "BUILD" COMMAND

**Details:** KB ID 6299

The "build" command has replaced the "make" command to compile most programs. The "build" command is a script that uses "make", and will take input from the command line to do things more intelligently. Build is available only for Unix Application Servers.

To see all of the parameters available to "build", simply type "build -?":

Named rules let you specify which part of the file system you would like to compile. Here are some examples:

"build all" - compile all of the IFAS core code

"build xport" - compile all of the programs in the transport subsystem.

"build cobol" - compile all COBOL source code

"build ." - compile the current directory (notice the dot ".", it signifies the current directory in Unix, and is the same as running the old "make all" command)

"build nu" - compile all programs beginning with 'nu'

If the -o parameter is not used, "build" will make a new 'run' file. If your current 'run' is working correctly, it is always wise to back it up before issuing the "build" command, as it will create a new 'run' file.

Currently, the triad subsystem is not a part of this list, and need to be done using "make" (reference KB 3057).

## **FAQ: ADVSETUP -- MAP2VB PROCESS WILL NOT UPDATE ALL SERVERS IN A FARM**

**Details:** KB ID 6381

Issue occurs with MAP2VB or after setting defaults in NUUPDF. MAP2VB is not updating all the servers in the farm. The map2vb process complete successfully but only one server in the farm received the updated .vbs files for defaults that are generated from MAP2VB.

### **Solution:**

In BrokerSecurityPolicy.xml file on each 7i server needs to modified to include IP addresses of the other servers in the farm; including the Unix server.

Additions are done in the "Local Requests" section of the BrokerSecurityPolicy.xml file (see below for an example). The BrokerSecurityPolicy.xml file should be located in the \bin directory of the installation path (typically C:\Program Files\SunGard Bi-Tech\Bin)

Example of code change:

```
<Policy Description="Local Requests" Action="Allow" >
```

```
<SourceAddresses>
```



```
<SourceAddress Address="$LOCALHOST"/>
```

```
<SourceAddress Address="$APPSERVER"/>
```

```
<SourceAddress Address="X.X.X.X"/>
```

```
<SourceAddress Address="Y.Y.Y.Y"/>
```

```
<SourceAddress Address="Z.Z.Z.Z"/>
```

```
</SourceAddresses>
```

```
<Connections>
```

```
<Connection Name="*" />
```

```
</Connections>
```

```
</Policy>
```

The X,Y,Z values are what need to be populated with IP addresses (there may be more or less SourceAddress entries depending on number of servers in farm)

Once you have made changes to the BrokerSecurityPolicy.xml file, you will need to perform an IISRESET and restart the dataprocessing service.

This should be done when users are not logged in using 7i screens.

**Notes:**

Another good test is to run ServerListQuery from BrokerDriver. If you are able to run this query and see the other servers in the farm, then the farm is setup correctly. If not, check the servergroup settings in System Admin, and the WWWServer0 registry key.

Make sure you can ping the 7i server from unix, and that the 7i servers can ping the unix box. There may be issues when dealing with DNS or a firewall in place.

If UNICAST mode is set, you may not be able to ping other 7i servers in the farm from a 7i server.

Check if SSL is being used on the webserver. Map2vb currently does not work when SSL is enabled. Refer to knowledgebase 5594 for setting up SSL and map2vb

settings.

Also check to make sure that map2vb can write to each server in the farm individually, and that the map2vb configuration file is using the farm's virtual IP address. You can either run map2vb from a prompt and pass the parameters, or create a backup of the map2vb configuration file and modify the file to use specific IP addresses. The map2vb configuration file is usually in \$BSI\_HOME/pub directory (see KB 4894). To run map2vb from a command prompt, please follow the following syntax:

```
map2vb - S (server) -N (connection ie ifas/prodcution), -r (7i virtual root, ie IFAS7).
```

To check if the files are updated on each server, go to

/inetpub/wwwrood/ifas7/connection/<connection name> /settings. Look at the time and date on each of the files to see if they are the same on all servers in the farm. Also check to make sure the clock is displaying the same time on all the servers in the farm.

Also check the btwebqrb.dll files on the server to ensure that the version of the file is the same as the version of PC Products; and the version of IFAS on the host system.

See KB6357 on network load balancing issues as well <http://so.bi-tech.com/isc/kb/external/6357.htm>. This KB will help deal with router issues and map2vb.

\* BrokerSecurityPolicy can also be disabled in the registry by going to `hkey_local_machine_/software/bitech/request broker/UseBrokerPolicy=0`. If this key is not in the registry, create a new DWORD with the value of 0 for UseBrokerPolicy. You will need to stop and start data processing services and reset IIS for the change to take effect. You should see a note in the event log whether the policy is in use.

## FAQ: COMPILING 7I SCREENS WHEN SSL IS ENABLED ON THE WEBSERVER

**Extended Text:** KB ID 6427

To allow screen compiles when SSL is enabled on the webserver you'll need to set the IIS port for 7i. To set the port go to Start > Programs > Sungard Bi-Tech > 7i Tools > Set IIS Port for 7i. Check the SSL check box and click OK. Once done test the screen compile. If that does not resolve the issue you can run the screen compile manually at a command prompt:

```
cscript screencompile.js <connection name> -v ifas7 -ssl -p <SSL port>
```

Run the above command in the directory where the 7i virtual directory is located. By default the 7i installation directory it is located at C:\InetPub\wwwroot\IFAS7. Replace the connection name with the ifas database connection and SSL port (typically 443).

-----  
One possible symptom of the screencompile where this change would need to be applied:

It is failing with: Status is 405

Error obtaining XML for BT20.ARBatchMaster status is 405

## FAQ: XLSDGEN NOT REGISTERED DURING INSTALL

**Details:** KB ID 6548

When attempting to use XLSDGEN on the 7i server the following error is generated in CDD: ActiveX component can't create object: 'ExcelGenerator.ARsTdesign' The error generated on the 7i varies, because the first call to create the object fails without warning on the server. The error will instead indicate the first attempt to use the object has failed. Such as: "Error: Object required: 'StyleDefault'" (NOTE: this is an example error only. Actual error may vary). The only true way to determine if the error is being caused by an unregistered XLSDGEN is to run the report directly from CDD running on the 7i server.

### Extended Text:

Sungard uses a scripted process to register the application components during the software installation. The script is called REGLIST.XML and is used by the BTDLLREG.exe executable. The file REGLIST.XML file is not updated to reflect the installation of XLSDGEN.

### Resolution:

The REGLIST.XML file must be updated to include a reference to XLSDGEN.DLL as follows (as always, you should make a backup copy of the original file before modifying it):

BEFORE

-- snip --

```
<DLL file="bt30al.dll" loc="BT" />
```

```
</DLL_LIST>
```

```
-- snip --
```

AFTER

```
-- snip --
```

```
<DLL file="bt30al.dll" loc="BT" />
```

```
<DLL file="xlsgen.dll" loc="BT" />
```

```
</DLL_LIST>
```

```
-- snip
```

Once the REGLIST.XML file is updated you must run BTDLLREG.EXE to complete the registration.

## FAQ: HIDING CONTENT IN DASHBOARD USING ASSOCIATIONS

**Details:** KB ID 6845

This article explains how you can use NUUPUS Associations in order to allow groups of users to view content through their Dashboard that cannot be seen by users not assigned to that group.

### Extended Text:

To hide content from groups of users:

1. Create two associations in nuupcd for the different sets of users
2. Assign each user in nuupus to one of the associations just defined. When you assign a new association, you must also explicitly assign FINANCE
3. Go to the Dashboard Customize screen: <http://<server>/ifas7/home/customize>
4. On the Groups tab, create two new Content Groups for the associations just created
5. On the Content Tab, add content to the two Content Groups just created

6. On the Default Content tab, make sure all content in the new groups is unchecked
7. Now log in as one of the users assigned to either association
8. On the Change Content tab, the user should see the content added to their association and be able to select it for display