Tech Note 640
Troubleshooting Process Graphics Client Live Data Delivery in Wonderware Information Server (WIS)

All Tech Notes, Tech Alerts and KBCD documents and software are provided "as is" without warranty of any kind. See the **Terms of Use** for more information.

Topic#: 002404
OpsManage09 Session#: TS111
Created: November 2009
Updated: December 2010

Introduction

Important! This document and its links to the utility are available on the **Wonderware Developer Network**. You must be a Customer First subscriber to download the utility.

This document complements **Tech Note 230** by explaining Troubleshooting techniques for the WIS Live-data delivery issues.

The common root cause for the live data issue is the Web Server URL. The URL is a manually-typed parameter in WinXMLExporter. This *Tech Note* explains some brief principles and provides steps for troubleshooting the issue. Wonderware Technical Support has also developed a client side Utility to make the troubleshooting easier.

Note: After the first edition of this TN published, we found another root-cause related to Internet Information Services (IIS) settings for the Live-data issue. Please refer to the **IIS Web Service Extensions Settings** section.

Application Versions

- Wonderware Information Server (WIS) 3.0 and later

WIS Client Live Data Flow Summary

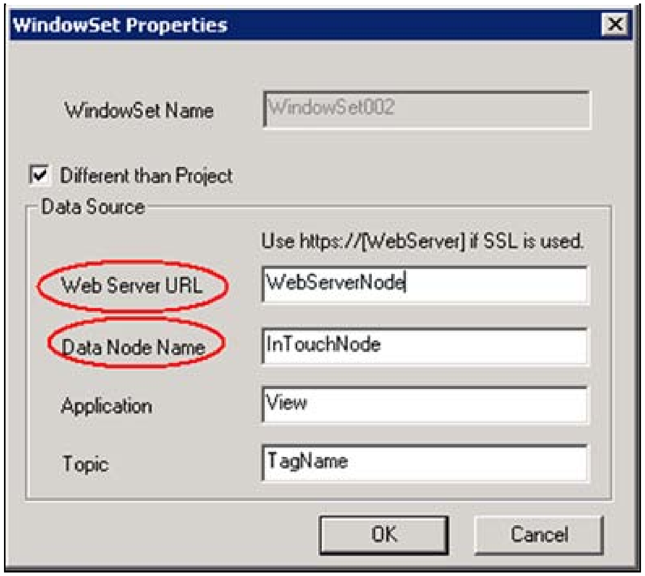
One of the WIS client components uses the **Web Server URL** value to make Internet/Intranet connection using Microsoft WinInet APIs.

When the Web Server URL contains a reachable Internet Information Services (IIS) node name or IP address, the client component sends the request to **\\IISServer\Scripts\fsoutpst.dll**. This ISAPI DLL is installed on the IIS Server side, which is responsible for making the data exchange with **Rdbhandler** service. If the HTTP communication channel is OK between the IIS and the WIS client component, the client component downloads the data from the response packet and passes through the data to other WIS client components for their live data.

Live Data Starts from WinXMLExporter

The WinXMLExporter converts the InTouch .win file into an XML file, then publishes the XML file to the Web Server in order to display Process Graphics. During the conversion, WinXMLExporter also puts the values of [Web Server URL], [Data Node Name], and other property settings into each converted XML file.

Figure 1 (below) shows a Windowset setting dialog and its corresponding XML element.



WindowSet Properties

WindowSet Name: WindowSet002

☑ Different than Project

Data Source

Use https://[WebServer] if SSL is used.

Web Server URL: WebServerNode

Data Node Name: InTouchNode

Application: View

Topic: TagName

Figure 1: WindowSet Settings

When you drag-drop the InTouch window into the above windowset, the WinXMLExporter embeds the following element into the XML file:

```
<Connection ID="1" Name=" " WebServer="WebServerNode" Node="InTouchNode" Application="View" Topic="TagName" ConnectionType="SUITELINK">
</Connection>
```

When you select this InTouch window from the Portal Process Graphics, the client-side Internet Explorer downloads its corresponding XML file to the client machine. One of the client components uses the **Connection** element from the downloaded XML file and composes a WinInet API call.

It then tries to make an Internet connection on http(s)://WebServerNode\Scripts\fsoutpst.dll.

Common Issues Summary

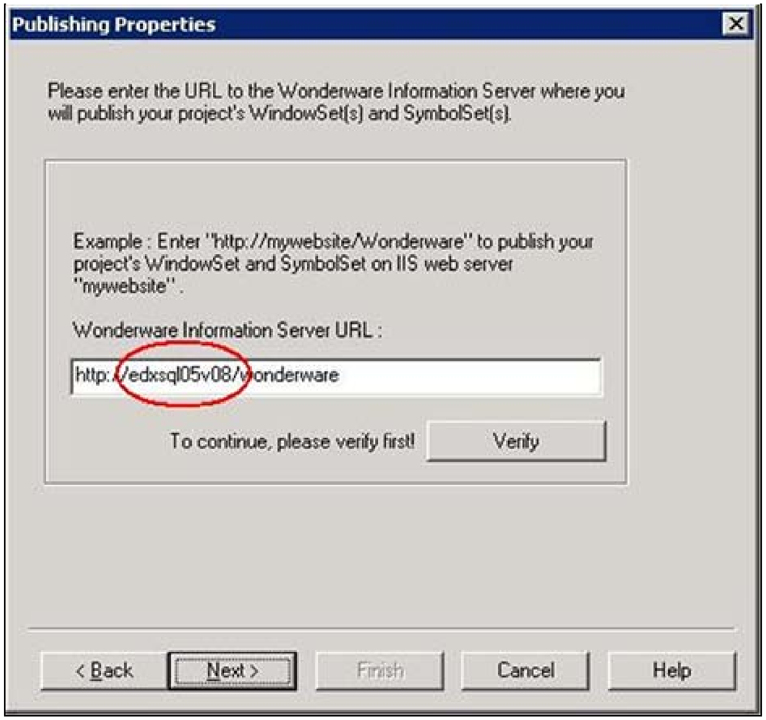
This section contains a summary of common live data delivery issues.

No Live Data at All But InTouch is Running

To troubleshoot this condition

- First, check whether the InTouchNode addressed in the Windowset is a reachable machine from WIS Portal node. The value of InTouchNode is used by **Rdbhandler.exe** in the WIS Portal machine. Rdbhandler.exe communicates to the InTouchNode using SuiteLink™.

  Refer to **Tech Note 230** about Diagnostics on Rdbhandler.exe.
- Check whether the given WebServerNode only contains machine name or IP Address. For example, the following values of WebServerNode are not correct: ServerName/Wonderware or http://ServerName. However, if the Protocol HTTPS is used, the value of WebServerNode should be https://ServerName.
- Check the value of [Wonderware Information Server URL] and [Web Server URL]. In a multiple-WIS Servers environment, the machine name or IP address in [Wonderware Information Server URL] can be different from the machine name or IP address on the WebServerNode as shown in Figure 2 and 3 (below).



Publishing Properties

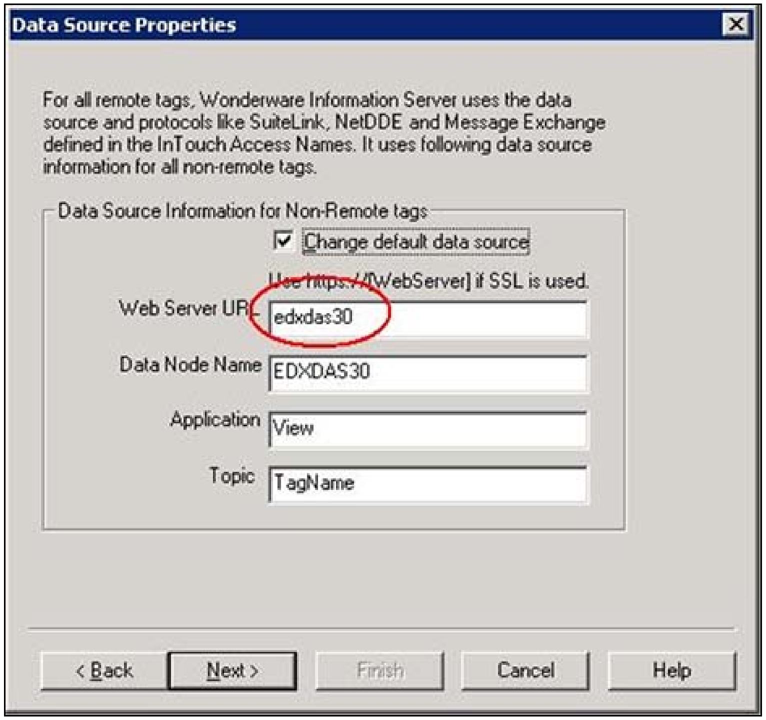
Please enter the URL to the Wonderware Information Server where you will publish your project's WindowSet(s) and SymbolSet(s).

Example : Enter "http://mywebsite/Wonderware" to publish your project's WindowSet and SymbolSet on IIS web server "mywebsite".

Wonderware Information Server URL : http://edxsql05v08/wonderware

To continue, please verify first!

Figure 2: Publishing Destination



Data Source Properties

For all remote tags, Wonderware Information Server uses the data source and protocols like SuiteLink, NetDDE and Message Exchange defined in the InTouch Access Names. It uses following data source information for all non-remote tags.

Data Source Information for Non-Remote tags

☑ Change default data source

Use https://[WebServer] if SSL is used.

Web Server URL: edxdas30

Data Node Name: EDXDAS30

Application: View

Topic: TagName

Figure 3: Web Site URL for Live Data

You Have Live Data in the WIS Portal Machine but No Live Data in the WIS Client Machines

The reason is that **LOCALHOST** is being used as the value for WebServerNode.

The WIS client component uses the value of WebServerNode to compose URL to the WIS Portal machine to get the live data. If LOCALHOST is used, the WIS client component will try to obtain the live data from itself.

You Have Live Data on the Intranet but No Live Data on the Internet

Outside the Firewall configuration, the most common reason for this issue is the machine name. For example, say that **edxdas30** is used as the value for the URL. From the Internet point of view, **edxdas30** is not a reachable name. The name **edxdas30** should be replaced by **10.2.87.169** (IP Address), or a fully-qualified (including the domain) name such as **edxDAS30.magellandev2000.dev.wonderware.com**.

Note: The Fully Qualified Domain Name (FQDN) consists of host name and domain name. In the above example, **edxDAS30.magellandev2000.dev** is the **host name** and **wonderware.com** is the **domain name.**

You Made Changes Based on the Above Recommendations but Still Have No Live Data

If the live data still cannot be seen, please download the **DebugView Utility** from Microsoft Web Site. Install the Utility to one the WIS client machine.

Use DebugView Utility to Diagnose the WIS Client Component

1. Launch the DebugView and click on Capture icon as shown in Figure 4 (below).

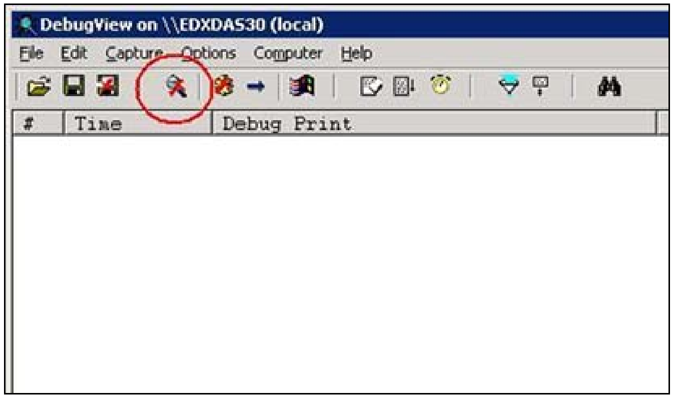
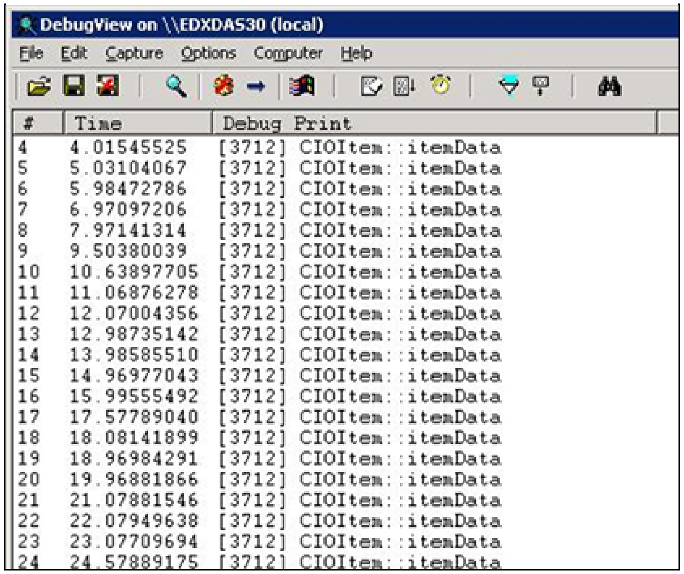


Figure 4: Start DebugView and Click on the Capture Icon

2. On the same WIS client machine, browse to the InTouch window from Process Graphics.
3. Click the Capture icon shown in the Figure 4 (Red circle) to activate the Capture function. Open the InTouch window. If the live data has been delivered to the WIS communication client components, it appears as shown as Figure 5 (below).


DebugView on \\EDXDAS30 (local)
File Edit Capture Options Computer Help

```
#   Time         Debug Print
4   4.01545525   [3712] CIOItem::itemData
5   5.03104067   [3712] CIOItem::itemData
6   5.98472786   [3712] CIOItem::itemData
7   6.97097206   [3712] CIOItem::itemData
8   7.97141314   [3712] CIOItem::itemData
9   9.50380039   [3712] CIOItem::itemData
10  10.63897705  [3712] CIOItem::itemData
11  11.06876278  [3712] CIOItem::itemData
12  12.07004356  [3712] CIOItem::itemData
13  12.98735142  [3712] CIOItem::itemData
14  13.98585510  [3712] CIOItem::itemData
15  14.96977043  [3712] CIOItem::itemData
16  15.99555492  [3712] CIOItem::itemData
17  17.57789040  [3712] CIOItem::itemData
18  18.08141899  [3712] CIOItem::itemData
19  18.96984291  [3712] CIOItem::itemData
20  19.96881866  [3712] CIOItem::itemData
21  21.07881546  [3712] CIOItem::itemData
22  22.07949638  [3712] CIOItem::itemData
23  23.07709694  [3712] CIOItem::itemData
24  24.57889175  [3712] CIOItem::itemData
```

Figure 5: Live Data Arrives at WIS client Communication Components

If you do not see a similar screen, there is still a communication issue between the WIS Portal and WIS client machines.

Using the WIS Client Diagnostic Utility

Wonderware Technical Support provides a **WIS Client Live Data Diagnostic Utilit**y that simulates the behavior of the WIS client when it sets up a connection to the WIS Portal. The Utility uses the cached XML file that WinXMLExportor converted from the InTouch win file.

The utility is available for **download on the WDN** and you must have a Customer First subscription to download it.

The Utility can test the following Connection Scenarios:

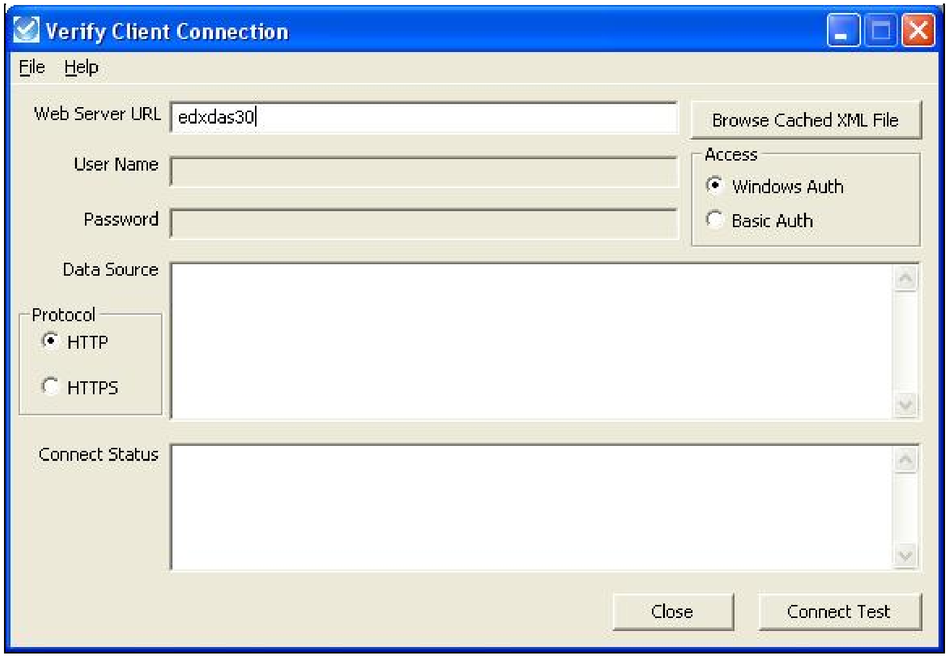
1. Windows Authentication with HTTP or HTTPS Protocols.
2. Basic Authentication with HTTP or HTTPS Protocols.

The **Connect Status** field displays either **Successful** or the detailed connection failure information.

In order to increase your troubleshooting efficiency, Wonderware Technical Support has developed a WIS Client Diagnostic Utility, which can do the following:

- Reads a cached WIS Process Graphics XML file.
- Extract the value of the Web Server URL as well as other connection property values from the XML file.
- Simulate the operations of WIS client component at runtime to communicate with WIS Server.

If there is any error during the communication, the Utility reports it. Figure 6 (below) shows the graphic user interface (GUI).



Web Server URL: edxdas30

Access: ● Windows Auth ○ Basic Auth

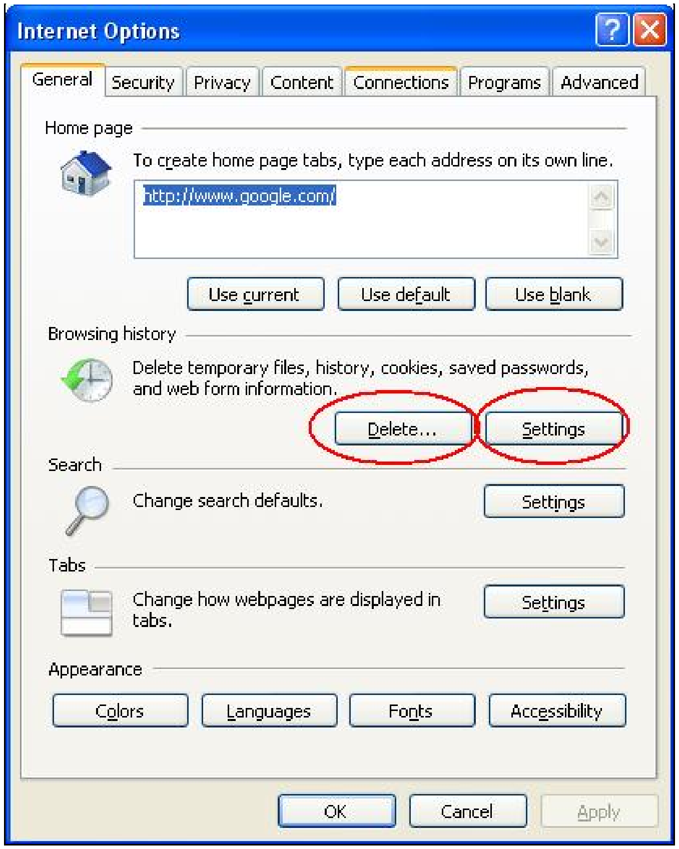
Protocol: ● HTTP ○ HTTPS

FIGURE 6: WIS Client Connectivity Testing Utility

When the WIS client selects an InTouch Process Graphics window, the corresponding converted XML file is downloaded and cached into the WIS client's Temporary Internet Files directory with type **WWXML**.

To troubleshoot

- Clean the entire Internet cached files by using Internet Options and deleting the Browsing history files (Figure 7 below).



To create home page tabs, type each address on its own line.

http://www.google.com/

Delete temporary files, history, cookies, saved passwords, and web form information.

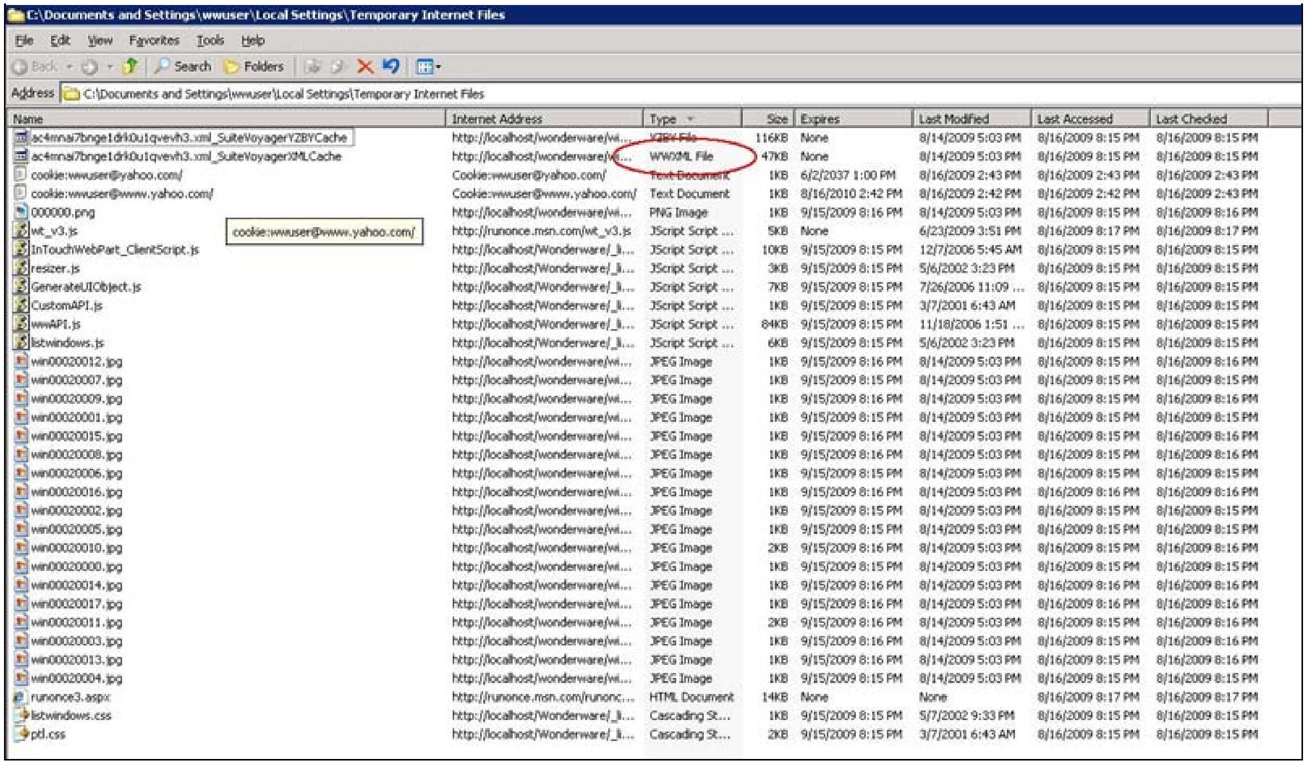
Change search defaults.

Change how webpages are displayed in tabs.

FIGURE 7: DELETE THE INTERNET CACHED FILES FOR IE 7.0

- Locate and display the Process Graphics InTouch window.

1. Click the **Settings** button shown in Figure 7 (above).
2. Click the **View Files** button in the next dialog box. You see the Temporary Internet Files directory.
3. Sort the files by Type and locate the WWXML type (Figure 8 below).

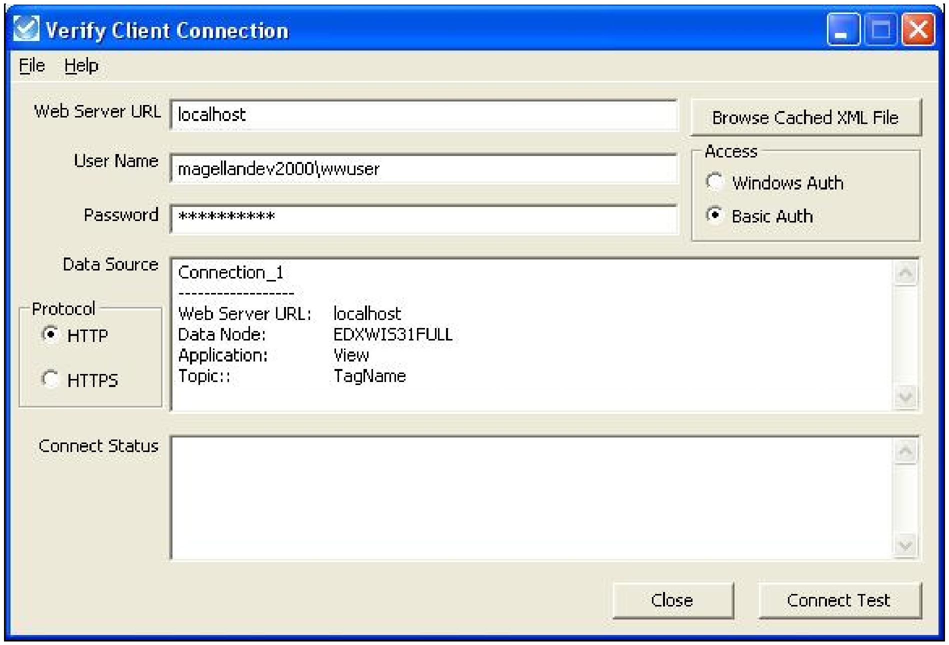

C:\Documents and Settings\wwuser\Local Settings\Temporary Internet Files
File Edit View Favorites Tools Help
Back Search Folders
Address C:\Documents and Settings\wwuser\Local Settings\Temporary Internet Files
cookie:wwuser@www.yahoo.com/

Name	Internet Address	Type	Size	Expires	Last Modified	Last Accessed	Last Checked
ac4mnai7bnge1drk0u1qvevh3.xml_SuiteVoyagerYZBYCache	http://localhost/wonderware/wi...	YZBY File	116KB	None	8/14/2009 5:03 PM	8/16/2009 8:15 PM	8/16/2009 8:15 PM
ac4mnai7bnge1drk0u1qvevh3.xml_SuiteVoyagerXMLCache	http://localhost/wonderware/wi...	WWXML File	47KB	None	8/14/2009 5:03 PM	8/16/2009 8:15 PM	8/16/2009 8:15 PM
cookie:wwuser@yahoo.com/	Cookie:wwuser@yahoo.com/	Text Document	1KB	6/2/2037 1:00 PM	8/16/2009 2:43 PM	8/16/2009 2:43 PM	8/16/2009 2:43 PM
cookie:wwuser@www.yahoo.com/	Cookie:wwuser@www.yahoo.com/	Text Document	1KB	8/16/2010 2:42 PM	8/16/2009 2:42 PM	8/16/2009 2:42 PM	8/16/2009 2:43 PM
000000.png	http://localhost/wonderware/wi...	PNG Image	1KB	9/15/2009 8:16 PM	8/14/2009 5:03 PM	8/16/2009 8:15 PM	8/16/2009 8:16 PM
wt_v3.js	http://runonce.msn.com/wt_v3.js	JScript Script ...	5KB	None	6/23/2009 3:51 PM	8/16/2009 8:17 PM	8/16/2009 8:17 PM
InTouchWebPart_ClientScript.js	http://localhost/Wonderware/_li...	JScript Script ...	10KB	9/15/2009 8:15 PM	12/7/2006 5:45 AM	8/16/2009 8:15 PM	8/16/2009 8:15 PM
resizer.js	http://localhost/Wonderware/_li...	JScript Script ...	3KB	9/15/2009 8:15 PM	5/6/2002 3:23 PM	8/16/2009 8:15 PM	8/16/2009 8:15 PM
GenerateUIObject.js	http://localhost/Wonderware/_li...	JScript Script ...	7KB	9/15/2009 8:15 PM	7/26/2006 11:09 ...	8/16/2009 8:15 PM	8/16/2009 8:15 PM
CustomAPI.js	http://localhost/Wonderware/_li...	JScript Script ...	1KB	9/15/2009 8:15 PM	3/7/2001 6:43 AM	8/16/2009 8:15 PM	8/16/2009 8:15 PM
wwAPI.js	http://localhost/Wonderware/_li...	JScript Script ...	84KB	9/15/2009 8:15 PM	11/18/2006 1:51 ...	8/16/2009 8:15 PM	8/16/2009 8:15 PM
listwindows.js	http://localhost/Wonderware/_li...	JScript Script ...	6KB	9/15/2009 8:15 PM	5/6/2002 3:23 PM	8/16/2009 8:15 PM	8/16/2009 8:15 PM
win00020012.jpg	http://localhost/wonderware/wi...	JPEG Image	1KB	9/15/2009 8:16 PM	8/14/2009 5:03 PM	8/16/2009 8:15 PM	8/16/2009 8:16 PM
win00020007.jpg	http://localhost/wonderware/wi...	JPEG Image	1KB	9/15/2009 8:15 PM	8/14/2009 5:03 PM	8/16/2009 8:15 PM	8/16/2009 8:15 PM
win00020009.jpg	http://localhost/wonderware/wi...	JPEG Image	1KB	9/15/2009 8:16 PM	8/14/2009 5:03 PM	8/16/2009 8:15 PM	8/16/2009 8:16 PM
win00020001.jpg	http://localhost/wonderware/wi...	JPEG Image	1KB	9/15/2009 8:15 PM	8/14/2009 5:03 PM	8/16/2009 8:15 PM	8/16/2009 8:15 PM
win00020015.jpg	http://localhost/wonderware/wi...	JPEG Image	1KB	9/15/2009 8:16 PM	8/14/2009 5:03 PM	8/16/2009 8:15 PM	8/16/2009 8:16 PM
win00020008.jpg	http://localhost/wonderware/wi...	JPEG Image	1KB	9/15/2009 8:16 PM	8/14/2009 5:03 PM	8/16/2009 8:15 PM	8/16/2009 8:16 PM
win00020006.jpg	http://localhost/wonderware/wi...	JPEG Image	1KB	9/15/2009 8:15 PM	8/14/2009 5:03 PM	8/16/2009 8:15 PM	8/16/2009 8:15 PM
win00020016.jpg	http://localhost/wonderware/wi...	JPEG Image	1KB	9/15/2009 8:16 PM	8/14/2009 5:03 PM	8/16/2009 8:16 PM	8/16/2009 8:16 PM
win00020002.jpg	http://localhost/wonderware/wi...	JPEG Image	1KB	9/15/2009 8:15 PM	8/14/2009 5:03 PM	8/16/2009 8:15 PM	8/16/2009 8:15 PM
win00020005.jpg	http://localhost/wonderware/wi...	JPEG Image	1KB	9/15/2009 8:15 PM	8/14/2009 5:03 PM	8/16/2009 8:15 PM	8/16/2009 8:15 PM
win00020010.jpg	http://localhost/wonderware/wi...	JPEG Image	2KB	9/15/2009 8:16 PM	8/14/2009 5:03 PM	8/16/2009 8:15 PM	8/16/2009 8:16 PM
win00020000.jpg	http://localhost/wonderware/wi...	JPEG Image	1KB	9/15/2009 8:15 PM	8/14/2009 5:03 PM	8/16/2009 8:15 PM	8/16/2009 8:15 PM
win00020014.jpg	http://localhost/wonderware/wi...	JPEG Image	1KB	9/15/2009 8:16 PM	8/14/2009 5:03 PM	8/16/2009 8:16 PM	8/16/2009 8:16 PM
win00020017.jpg	http://localhost/wonderware/wi...	JPEG Image	1KB	9/15/2009 8:16 PM	8/14/2009 5:03 PM	8/16/2009 8:16 PM	8/16/2009 8:16 PM
win00020011.jpg	http://localhost/wonderware/wi...	JPEG Image	2KB	9/15/2009 8:16 PM	8/14/2009 5:03 PM	8/16/2009 8:15 PM	8/16/2009 8:16 PM
win00020003.jpg	http://localhost/wonderware/wi...	JPEG Image	1KB	9/15/2009 8:16 PM	8/14/2009 5:03 PM	8/16/2009 8:15 PM	8/16/2009 8:16 PM
win00020013.jpg	http://localhost/wonderware/wi...	JPEG Image	1KB	9/15/2009 8:16 PM	8/14/2009 5:03 PM	8/16/2009 8:15 PM	8/16/2009 8:16 PM
win00020004.jpg	http://localhost/wonderware/wi...	JPEG Image	1KB	9/15/2009 8:15 PM	8/14/2009 5:03 PM	8/16/2009 8:15 PM	8/16/2009 8:15 PM
runonce3.aspx	http://runonce.msn.com/runonc...	HTML Document	14KB	None	None	8/16/2009 8:17 PM	8/16/2009 8:17 PM
listwindows.css	http://localhost/Wonderware/_li...	Cascading St...	1KB	9/15/2009 8:15 PM	5/7/2002 9:33 PM	8/16/2009 8:15 PM	8/16/2009 8:15 PM
ptl.css	http://localhost/Wonderware/_li...	Cascading St...	2KB	9/15/2009 8:15 PM	3/7/2001 6:43 AM	8/16/2009 8:15 PM	8/16/2009 8:15 PM

FIGURE 8: Sort by Type and Find the WWXML File

4. Copy the WWXML file into the current Diagnostic Utility's working directory and click the **Browse Cached XML File** button in the Utility. The embedded value of WebServerURL in the XML file is extracted into the Web Server URL Text Box.

5. Figure 9 (below) demonstrates the verification prompt for IIS Basic Authentication Access mode. Basic Authentication mode requires giving a User Name and Password. For a domain user, the User Name should be **DomainName\UserName**.



Verify Client Connection

File Help

Web Server URL: localhost

User Name: magellandev2000\wwuser

Password: **********

Access: ○ Windows Auth ● Basic Auth

Protocol: ● HTTP ○ HTTPS

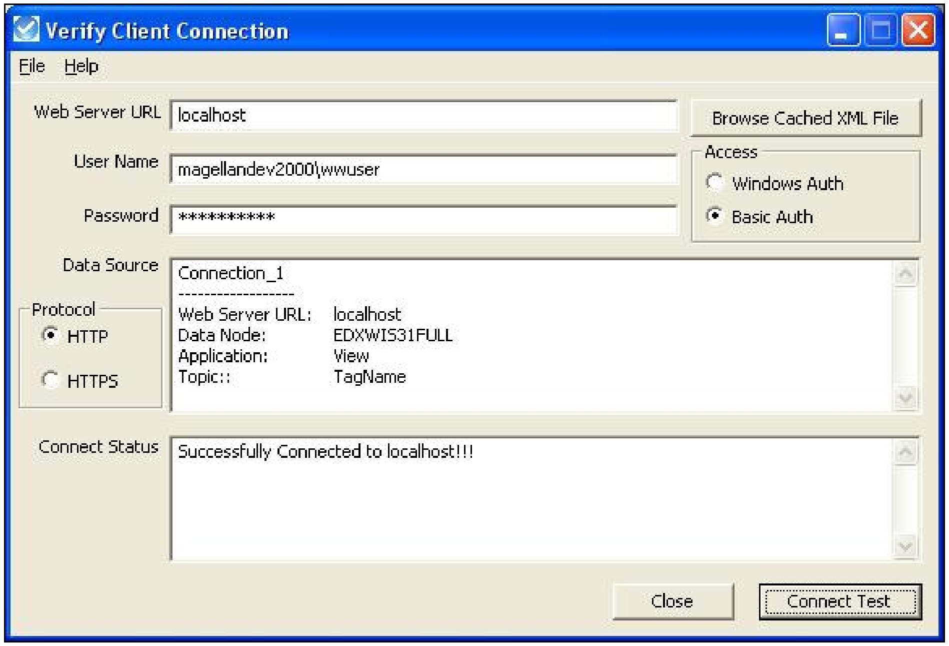
Data Source:

```
Connection_1
------------------
Web Server URL:   localhost
Data Node:        EDXWIS31FULL
Application:      View
Topic::           TagName
```

Connect Status:

Figure 9: Browse Cached XML File

6. Click the **Connect Test** button. You will see the runtime status in the **Connect Status** text box.



Verify Client Connection

File Help

Web Server URL: localhost

User Name: magellandev2000\wwuser

Password: **********

Access: ○ Windows Auth ● Basic Auth

Protocol: ● HTTP ○ HTTPS

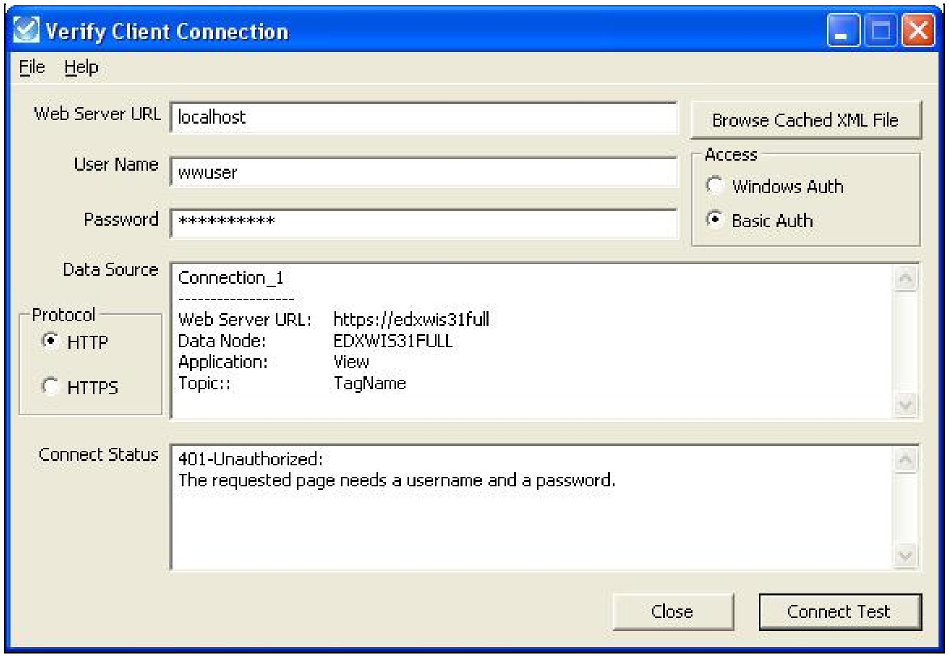
Data Source:

```
Connection_1
------------------
Web Server URL:   localhost
Data Node:        EDXWIS31FULL
Application:      View
Topic::           TagName
```

Connect Status: Successfully Connected to localhost!!!

Figure 10: Connect Test Successful

7. Use same example as above except the DomainName is missing. We will see **Unauthorized Error** in the **Connect Status Text Box** (Figure 11 below).



Web Server URL: localhost

User Name: wwuser

Password: **********

Data Source:

```
Connection_1
------------------
Web Server URL:   https://edxwis31full
Data Node:        EDXWIS31FULL
Application:      View
Topic::           TagName
```

Connect Status:

```
401-Unauthorized:
The requested page needs a username and a password.
```

FIGURE 11: 401 ERROR-UNAUTHORIZED

IIS Web Service Extensions Settings

For IIS 6.0 and IIS 7.0 or 7.5, Microsoft added a security component (Web Service Extensions) that can prevent the Dynamic Web content from attack. This component also plays a role in WIS Process Graphics Live-data.

WIS Configurator delivers a Web Service Extension (WSE) called **wwSVProcessGraphics** in **Allowed** status. The WSE holds two required ISAPI DLL file paths (path + file names) that are Allowed to visit from a Web Virtual Directory or Application. The two allowed ISAPI DLLs are:

- **fsoutpst.dll** – This is the entry point from WIS Process Graphics client call to IIS.
- **ivfw.dll** – This is the entry point from RdbHandler call to IIS (not discussed in this *Tech Note*, but we must note it).

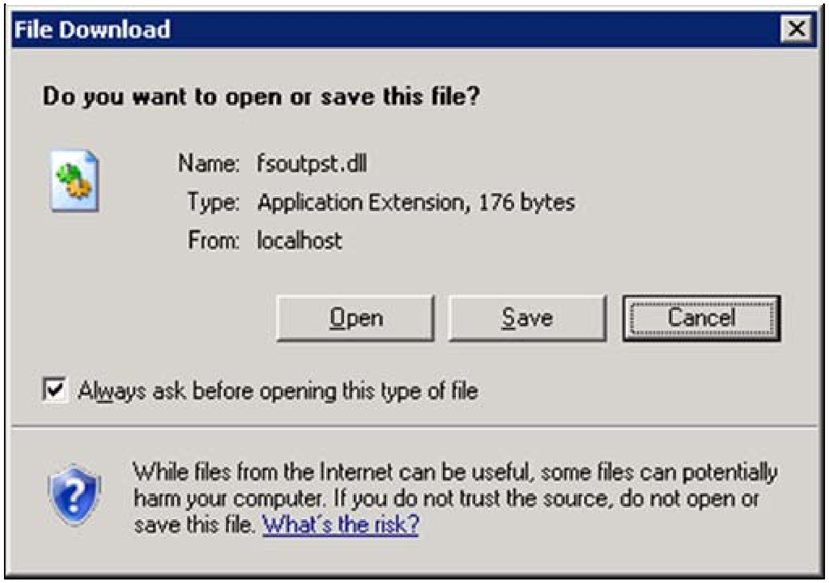
When you click on an InTouch window link under the Process Graphics node, the **rdbcmi.dll** WIS client components makes a HTTP(s) call to **http(s)://WIS_Server/Scripts/fsoutpst.dll**.

Note: The rdbcmi.dll call is based on the WIS URL you provide.

This call is to the **RdbHandler.exe** (Windows service) to get or set data from other backend components, SuiteLink and Lmx. In the above URL, **Scripts** is a virtual directory which is defined in IIS. Like other IIS virtual directories, **Scripts** must have a physical directory defined. The WIS Configurator defines by default in the physical directory as **[C]:\Inetpub\Scripts**.

If we do a test in a WIS IE client by using http://WIS_Server/Scripts/fsoutpst.dll URL, you should see the **File Download** prompt for the file..

Click **Cancel** to exit (Figure 12 below).



File Download

Do you want to open or save this file?

Name: fsoutpst.dll

Type: Application Extension, 176 bytes

From: localhost

☑ Always ask before opening this type of file

While files from the Internet can be useful, some files can potentially harm your computer. If you do not trust the source, do not open or save this file. What's the risk?

FIGURE 12: PROCESS GRAPHICS CLIENT ENTRY POINT TO IIS

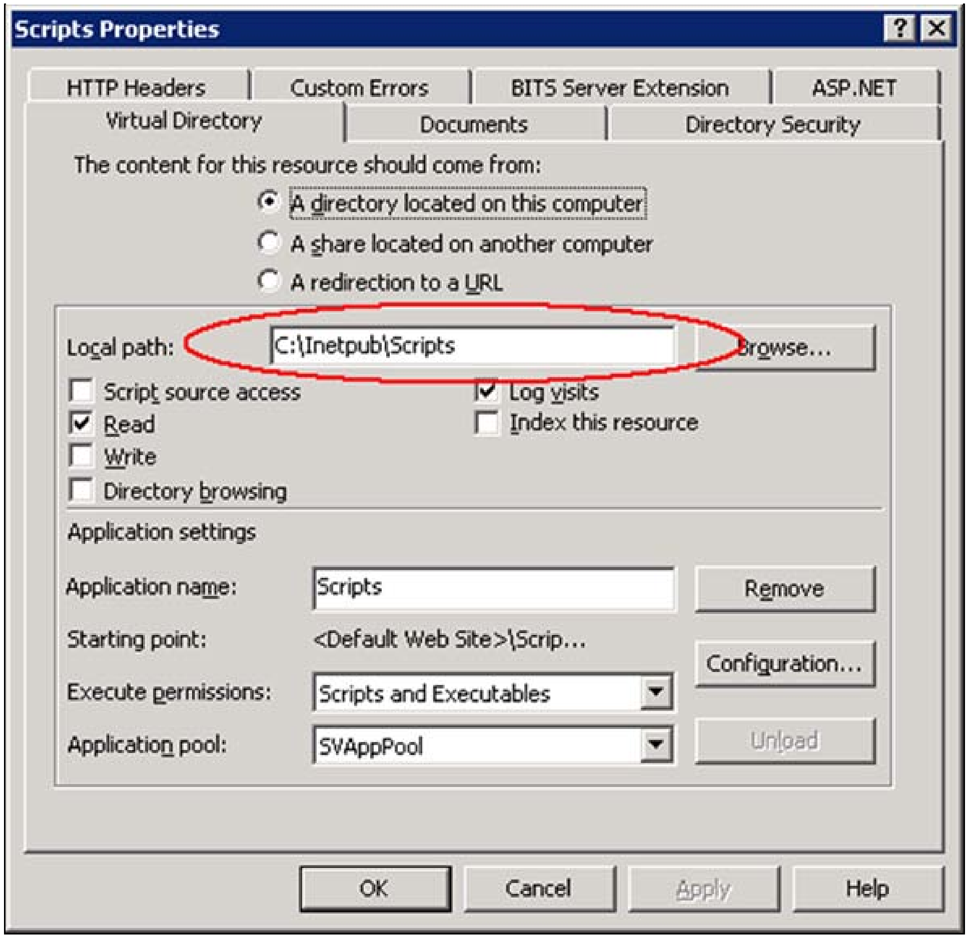
Detailed Scripts Settings

The following section shows the detailed settings on **Scripts** and its corresponding Web Service Extensions (WSE).

The physical paths defined in **Scripts** and WSE must match. Otherwise, you will not have Live-data.

For IIS 6.0

1. Check the Scripts physical directory



The content for this resource should come from:

● A directory located on this computer
○ A share located on another computer
○ A redirection to a URL

Local path: C:\Inetpub\Scripts

□ Script source access
☑ Read
□ Write
□ Directory browsing
☑ Log visits
□ Index this resource

Application settings

Application name: Scripts

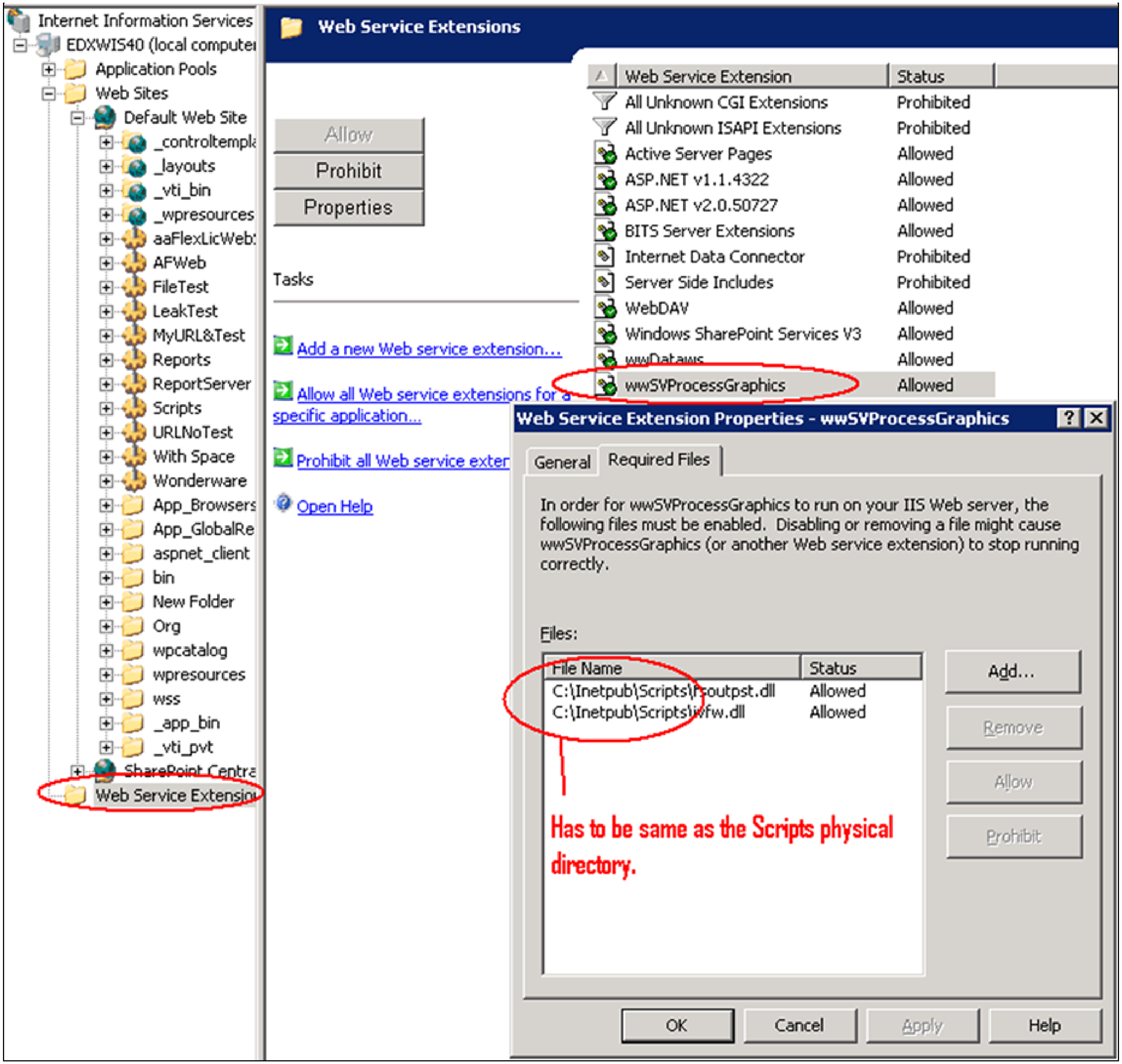
Starting point: <Default Web Site>\Scrip...

Execute permissions: Scripts and Executables

Application pool: SVAppPool

Figure 13: Physical Path for Scripts

2. Check the Web Service Extensions physical directory (Figure 14 below).



Web Service Extension	Status
All Unknown CGI Extensions	Prohibited
All Unknown ISAPI Extensions	Prohibited
Active Server Pages	Allowed
ASP.NET v1.1.4322	Allowed
ASP.NET v2.0.50727	Allowed
BITS Server Extensions	Allowed
Internet Data Connector	Prohibited
Server Side Includes	Prohibited
WebDAV	Allowed
Windows SharePoint Services V3	Allowed
wwDataws	Allowed
wwSVProcessGraphics	Allowed

In order for wwSVProcessGraphics to run on your IIS Web server, the following files must be enabled. Disabling or removing a file might cause wwSVProcessGraphics (or another Web service extension) to stop running correctly.

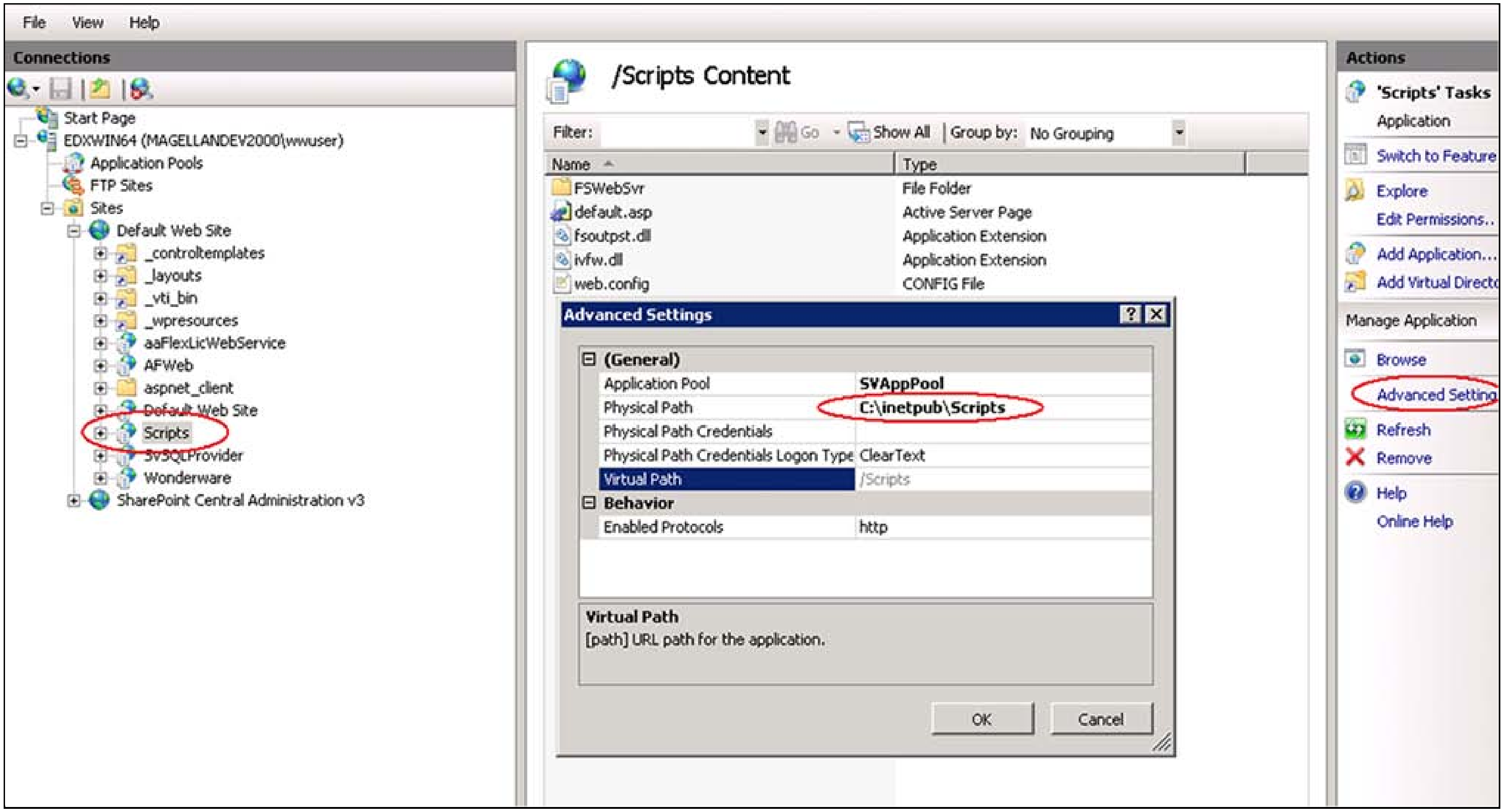
File Name	Status
C:\Inetpub\Scripts\Psoutpst.dll	Allowed
C:\Inetpub\Scripts\Wfw.dll	Allowed

Has to be same as the Scripts physical directory.

Figure 14: WSE and Scripts Physical directories Must Be the Same

For IIS 7.0 or IIS 7.5

- Check the Scripts physical directory.



/Scripts Content

Name	Type
FSWebSvr	File Folder
default.asp	Active Server Page
fsoutpst.dll	Application Extension
ivfw.dll	Application Extension
web.config	CONFIG File

Advanced Settings

(General)	
Application Pool	SVAppPool
Physical Path	C:\inetpub\Scripts
Physical Path Credentials	
Physical Path Credentials Logon Type	ClearText
Virtual Path	/Scripts
Behavior	
Enabled Protocols	http

Virtual Path
[path] URL path for the application.

Figure 15: Scripts Physical Directory

- Check the Web Service Extensions physical directory

1. In the **Feature** pane, double click **ISAPI** and CGI Restrictions (Figure 16 below).

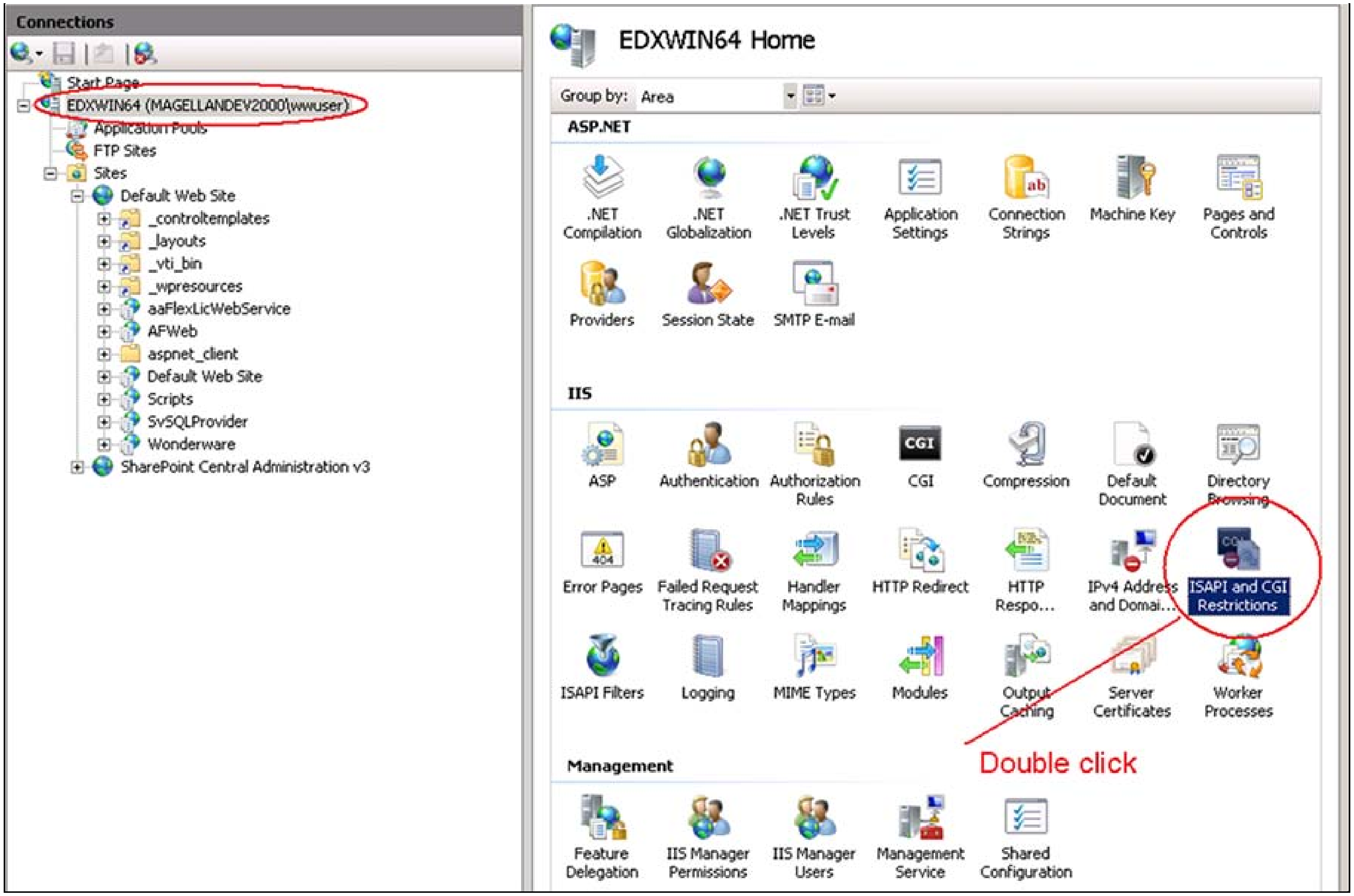
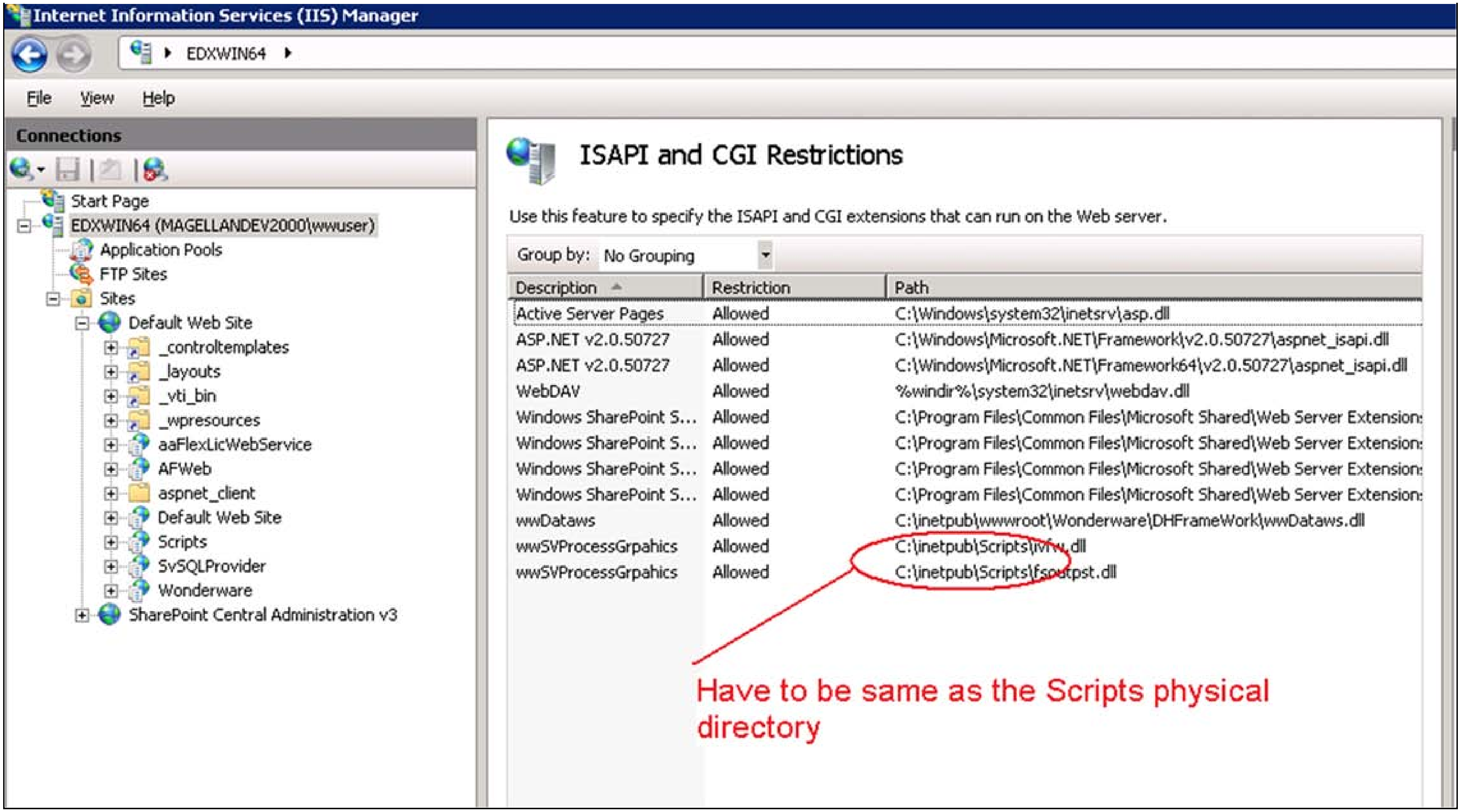


FIGURE 16: OPEN ISAPI AND CGI RESTRICTIONS

2. Check to see that the **WSE** and **Scripts** physical directories are the same.



ISAPI and CGI Restrictions

Use this feature to specify the ISAPI and CGI extensions that can run on the Web server.

Group by: No Grouping

Description	Restriction	Path
Active Server Pages	Allowed	C:\Windows\system32\inetsrv\asp.dll
ASP.NET v2.0.50727	Allowed	C:\Windows\Microsoft.NET\Framework\v2.0.50727\aspnet_isapi.dll
ASP.NET v2.0.50727	Allowed	C:\Windows\Microsoft.NET\Framework64\v2.0.50727\aspnet_isapi.dll
WebDAV	Allowed	%windir%\system32\inetsrv\webdav.dll
Windows SharePoint S...	Allowed	C:\Program Files\Common Files\Microsoft Shared\Web Server Extension:
Windows SharePoint S...	Allowed	C:\Program Files\Common Files\Microsoft Shared\Web Server Extension:
Windows SharePoint S...	Allowed	C:\Program Files\Common Files\Microsoft Shared\Web Server Extension:
Windows SharePoint S...	Allowed	C:\Program Files\Common Files\Microsoft Shared\Web Server Extension:
wwDataws	Allowed	C:\inetpub\wwwroot\Wonderware\DHFrameWork\wwDataws.dll
wwSVProcessGrpahics	Allowed	C:\inetpub\Scripts\ivfw.dll
wwSVProcessGrpahics	Allowed	C:\inetpub\Scripts\fsoutpst.dll

Have to be same as the Scripts physical directory

Summary

WinXMLExporter and Web Service Extensions (WSE) play an extremely important role in WIS Live data. A clear understanding of each setting in the WinXMLExporter will be useful in eliminating the live data issue if it occurs.

E. Xu & K. Nourbakhsh

Tech Notes are published occasionally by Wonderware Technical Support. Publisher: Invensys Systems, Inc., 26561 Rancho Parkway South, Lake Forest, CA 92630. There is also technical information on our software products at **Wonderware Technical Support.**

For technical support questions, send an e-mail to **support@wonderware.com**.

Back to top