

BonCode Apache Tomcat AJP 1.3 Connector

How to connect IIS + to Apache Tomcat via the
Apache JServ Protocol 1.3 using BonCode AJP
connector

© 2011, 2012 by Bilal Soylu

Table of Contents

BonCode AJP 1.3 Connector	4
License.....	4
The Beginning	4
BonCode Connector Solution.....	4
Prerequisites	5
Note on http headers.....	5
The Package Content	5
How to Install Automatically.....	6
How to Install Manually	8
ASP.NET Web Services Extension Enabled on IIS6:.....	12
Wildcard mappings:	12
Step 1: IIS6 Example of adding a wildcard map	13
IIS7 Example of adding a wildcard managed handler	14
Example of correctly installed files and directories on IIS on a Site basis (all):	15
OPTIONAL STEPS (INTERNATIONAL CHARACTERS)	16
Sample web.config file (IIS6 only)	16
The settings file (BonCodeAJP13.settings).....	17
Example Settings file	20
Using The Connector.....	21
The Issue with Path-Info	21
The Issue	21
The Solution	21
RFC Compliance	21
General Recommendations	23
Unblock before use	23
Connector Thoughts.....	23
Setup Recommendations.....	24
Identifying the installed Version	26
HTTP Headers and connector behavior	26
Inbound White Lists	27
Inbound Black Lists.....	27

Scripted Deployment	28
Installer Command Line Parameters.....	28
Installer Parameters File (installer.settings)	30
Troubleshooting.....	34
Q&A.....	39

BonCode AJP 1.3 Connector

License

This manual is distributed under the Creative Commons v.3 license (<http://creativecommons.org/licenses/by/3.0/>). The code is distributed under the Apache License Version 2 (<http://www.apache.org/licenses/LICENSE-2.0>). In short, you are totally and utterly at your own risk and merit if you use any samples provided.

Full source code can be downloaded from here:

<https://github.com/Bilal-S/iis2tomcat>

The Beginning

You are using Microsoft IIS 5, 6, 7, or 8 as your webserver and want to connect it effectively to an Apache Tomcat application server. You want to make this easy and do not want to install ISAPI based connectors which have their own share of issues. Or, you may want to interoperate better using multiple Tomcat applications with one IIS server.

BonCode Connector Solution

The BonCode AJP (Apache JServ Protocol version 1.3) Connector uses already existing pathways to connect to Apache Tomcat. The AJPv13 protocol details can be found here (<http://tomcat.apache.org/connectors-doc/ajp/ajpv13a.html>).

In general it is a preference question whether you use the ISAPI connection, which is the old C/C++ way of connecting IIS to other protocols or the methods implemented in the BonCode connector.

There are several advantages with the BonCode connector though:

- no ISAPI code (less risk of memory leaks, easier debugging)
- no IIS6 vestiges or backward compatibility elements needed on IIS7
- all managed code .net extensibility framework
- works on IIS5.1, IIS6, IIS7, IIS7.5, IIS8
- speed improvements
- easier control by using IIS UI and handler mappings
- no virtual directories and virtual mapping needed
- configuration can be inherited to sub-sites and virtual sites
- easy install/uninstall
- support partial stream sending to browser (automatic flushing) with faster response to client
- support both 32/64 bit of Windows with same process and files
- transfer of all request headers to servlet container
- build in simple-security for web-administration pages

Prerequisites

Besides Apache Tomcat, you will need a dot-net framework (3.51 or higher) installed on your machine. The install packages includes .net version 3.5. This is also a standard feature in Windows 2008 server and Windows 7. It can be downloaded for other versions of Windows. The standard package downloaded contains the installation stub that can be run separately if needed.

IMPORTANT: If you have already installed .net framework version 4 or 4.5 you do not need to install 3.51. If you receive a 500.21 error. Please see troubleshooting section.

Note on http headers

Http headers are important markers that are sent along with http requests on the internet. They can denote the length of content, the type of content, how to treat the response etc.

The BonCode connector will send more headers along than any previous connection technology between IIS and Apache Tomcat. Header name case as sent by browser will be preserved. However, if header names are set by IIS only they will be lower cased, e.g. X-Forwarded-For will become x-forwarded-for, this is to follow convention of previous connectors. Also, the connector will suppress headers that do not contain any data (are blank), to save on network bandwidth and processing. You can influence these and other behaviors by supplying settings that change how the connector operates. See the “The settings file (BonCodeAJP13.settings)” later in this document for details.

The Package Content

The BonCode IIS to Apache Tomcat connector consists of 2 mandatory and one optional file.

BonCodeIIS.dll

- ⇒ This is the main handler for the IIS communication process. It will intercept requests and pass them to the protocol handler for translation.

BonCodeAJP13.dll

- ⇒ This is the implementation of JServ 1.3 communication protocol. This is the main library for communicating with tomcat.

BonCodeAJP13.settings

- ⇒ This file is completely optional. It is a settings file that can be used to change behavior of the connector. It needs to be placed in the same directory as BonCodeAJP13.dll.

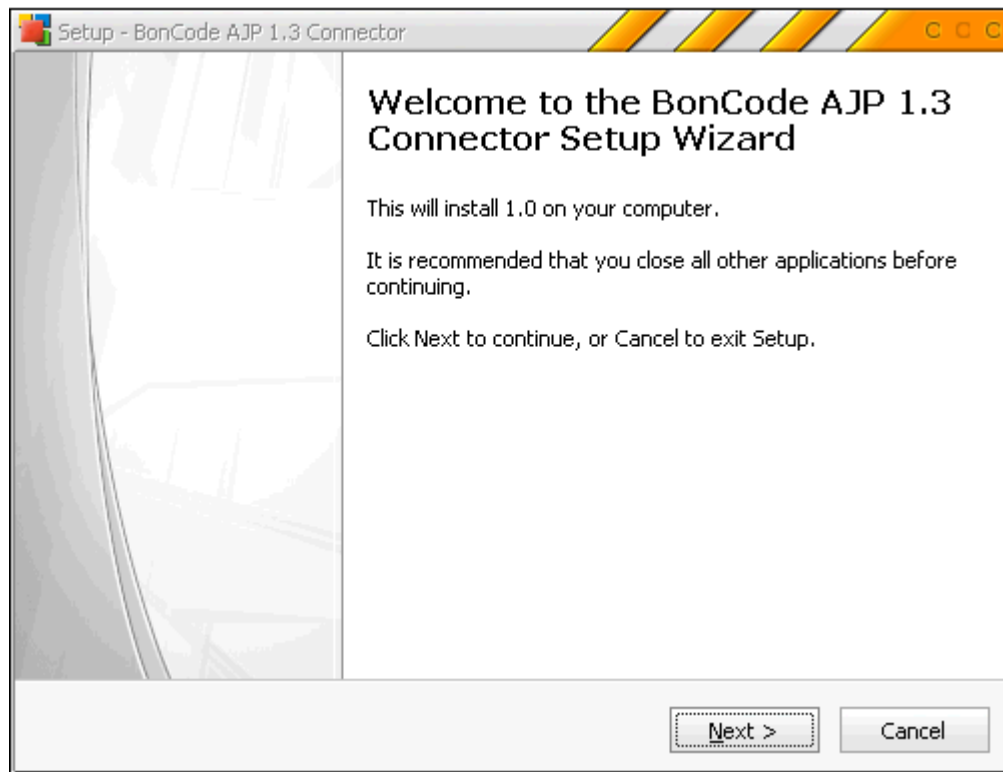
Connector_Setup.exe

- ⇒ This is the automated installation program. It will do all tasks needed (outlined in the manual installation section) to get the connection between IIS and Apache Tomcat to work.

Dotnetfx35setup.exe

- ⇒ The .net framework version 3.51 installation program. It can be used to install this framework on your computer if it is not already installed. The automated installer (Connector_Setup.exe) will check and run this if needed. For windows 2008 and later you should use the feature manager to install from the server manager.

How to Install Automatically



Before you start the automatic installation please ensure that you have unblocked the content. If you downloaded the software package from the Internet, Windows may block the execution of the program. See [Unblock before use](#) section for how to do so.

Automatic installation is the recommend way of installing the connector and configuring your webserver.

Start the **Connector_Setup.exe** program within the distribution directory and follow prompts. The automated setup will walk you through the available options.

You can also adjust the setup manually after installer has completed the initial tasks. For example, you can change the settings file manually within the “\BIN” directory of your website (or C:\windows for global installs). See below for details on the settings file.

Most choices in the installer should be self-explanatory; however, the install mode step requires further clarification. The setup program offers two choices of installs on the “Select All (Global) or Specific Site Option” screen. **These are mutually exclusive forms of installs.**

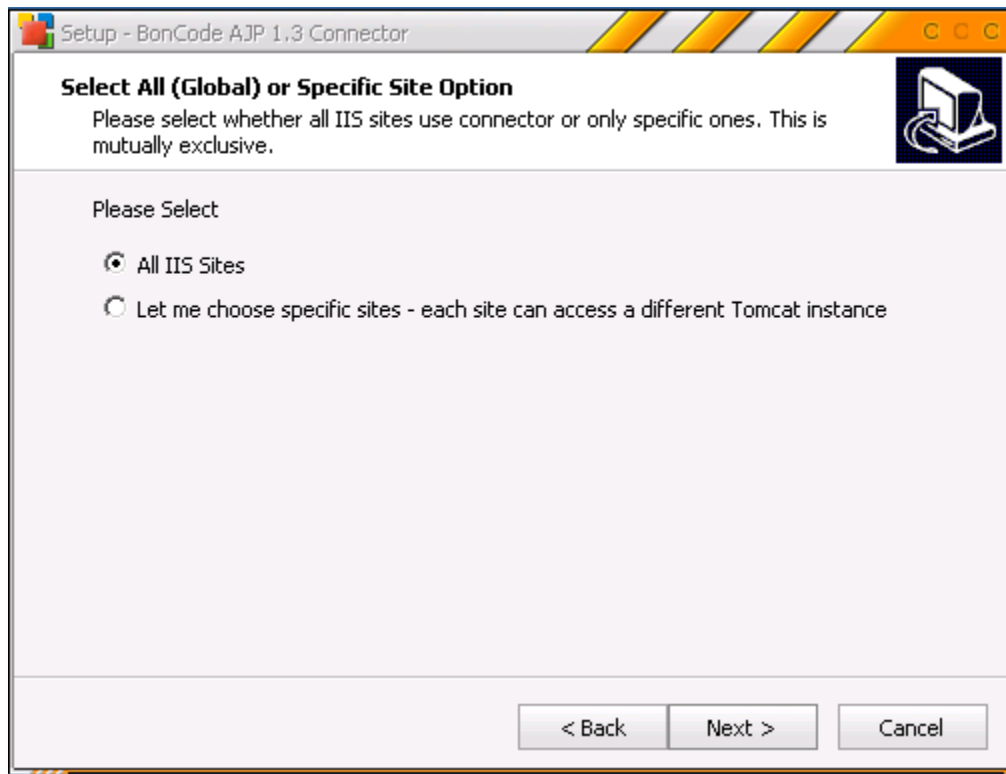
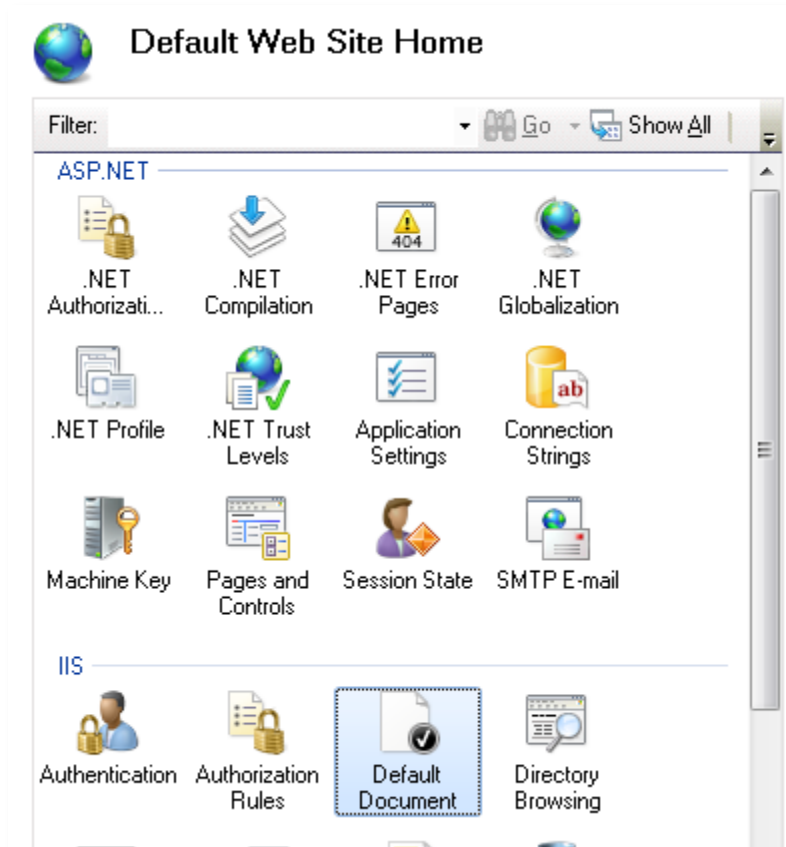


Figure 1: Installer Mode Selection Step

- a) All IIS Sites (Global Install). This is the most common use case. You wish to connect all your IIS sites to **ONE** instance of Apache Tomcat. The setup will make some changes for maximum efficiency of memory and code management. For example, this will install the connector in Windows Global Assembly Cache (GAC), using one instance of code for all your sites. There is only one configuration file located in your system directory (c:\windows). All existing and new sites will automatically inherit connector settings and be enabled for use. If you select this mode of install by accident or need to change it later, you will have to first UNINSTALL the connector, then, re-run the setup and select the other option.
- b) Let me choose specific sites. This is a good option if you do not want all your IIS sites to be enabled to communicate with Tomcat or if you need to access multiple Tomcat instances, e.g. a different Tomcat instance for each site. Each site can be configured individually. You can re-run the setup as often as needed to enter different Tomcat parameters for different IIS sites. The subsequent step in the setup will allow you to choose the affected IIS sites. In this mode, new sites will not automatically be enabled to use connector. Each instance will have its own configuration and code files.

The installer will also add the default documents such as *index.cfm* and *index.jsp* depending on options chosen. You can make manual changes to this using the “Default Document” feature of IIS



IIS 7 GUI for Default Document Feature for use after install completes

How to Install Manually

If possible avoid manual install as there are many elements to consider. The installation consists of several steps; some of these steps are defaulted and are only here for completeness. For example on tomcat 6 and 7, all the Apache Tomcat defaults will work out of the box, and only the IIS configuration is needed.

Prerequisite on the IIS side is the installation of .NET Framework 3.5.1. This can be downloaded (<http://www.microsoft.com/downloads/en/details.aspx?FamilyID=ab99342f-5d1a-413d-8319-81da479ab0d7>) or installed as feature on Windows server.

IIS Generic Configuration (both IIS6 and IIS7):

- a) Create a subdirectory under your default IIS7 document root and name it “BIN”, e.g. if your website document root is located at c:\inetpub\wwwroot, create a “BIN” subdirectory here c:\inetpub\wwwroot\BIN
- b) If you want to create a log file in this newly created directory assign the IIS_IUSRS group sufficient rights to READ/WRITE in “BIN” subdirectory. Alternately specify a different directory for logging in the settings file. By default logging is disabled and this step can be skipped.

IIS7 Configuration:

- c) Ensure that the Application pool associated with the website operates in “Integrated” Managed Pipeline Mode.



Application Pools

This page lets you view and manage the list of application pools on the server. Application pools contain one or more applications, and provide isolation among d

Filter: Default	Go	Show All	Group by:
Name	Status	.NET Framework ...	Managed Pipeline Mode
DefaultAppPool	Stopped	v4.0	Integrated

- d) Add dot net extensibility feature to IIS. You can do so using the computer management UI or running this command in a command window (important not to use a power-shell (blue background) window):
- e) Add managed handler mapping(s). You can do so using the IIS manager UI or you can run this appcmd command (it adds a handler for jsp pages). You can add as many handlers as you need, simply change the name attribute as needed.

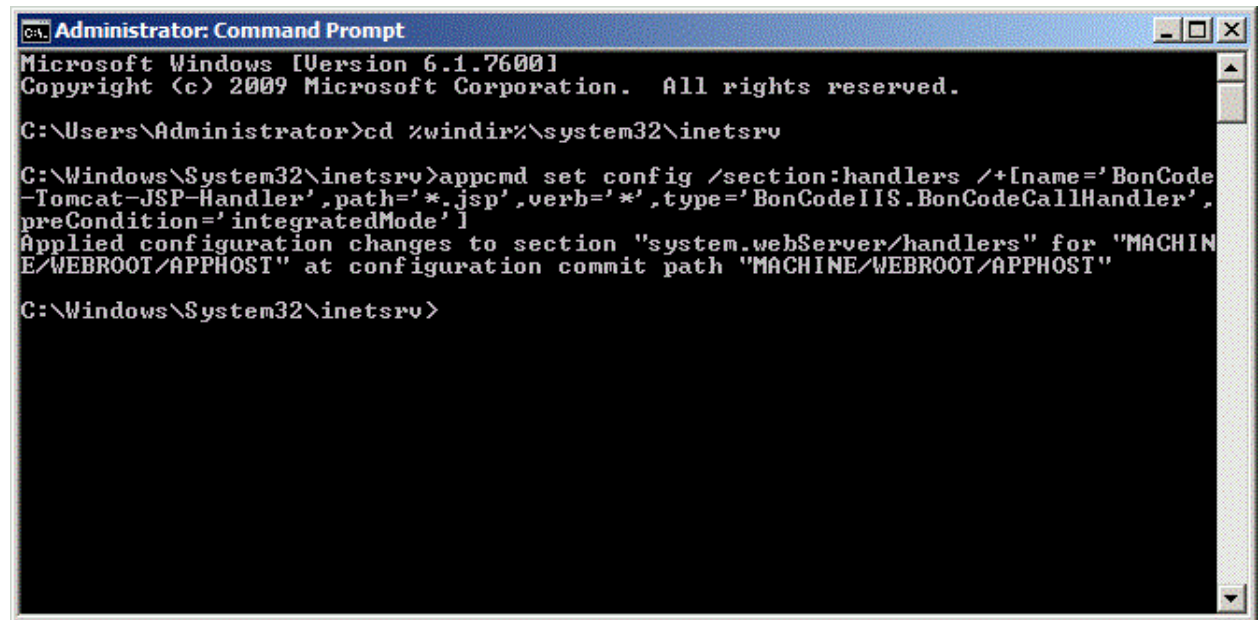
```
start /w pkgmgr /iu:IIS-NetFxExtensibility
```

appcmd.exe is located here: %windir%\system32\inetsrv

Paste the next lines as one line into the command:

```
%windir%\system32\inetsrv\appcmd set config /section:handlers  
/+[name='BonCode-Tomcat-JSP-  
Handler',path='*.jsp',verb='*',type='BonCodeIIS.BonCodeCallHandle  
r',preCondition='integratedMode']
```

Example Command Prompt with executed commands:



```
Administrator: Command Prompt
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>cd %windir%\system32\inetsrv

C:\Windows\System32\inetsrv>appcmd set config /section:handlers /+[name='BonCode-Tomcat-JSP-Handler',path='*.jsp',verb='*',type='BonCodeIIS.BonCodeCallHandler',preCondition='integratedMode']
Applied configuration changes to section "system.webServer/handlers" for "MACHINE/WEBROOT/APPHOST" at configuration commit path "MACHINE/WEBROOT/APPHOST"

C:\Windows\System32\inetsrv>
```

IIS6 Configuration

- c) Ensure that you allow ASP.NET in web service extension within your website (see image below).
- d) Add a managed handler mapping to your web.config file. If you do not have a web.config file in your website, create one in your document root. A sample is available later in this document.
- e) Add an HTTP Handler extension to IIS. To do so follow the remainder of the steps outlined here:
- f) Right click on site within site manager and select **Properties**.
- g) Click the **Home Directory** or **Virtual Directory** tab, and then click **Configuration**.
- h) On the **Mappings** tab, click **Add**. The **Add/Edit Application Extension Mapping** dialog box is displayed.
- i) In the **Executable** box, type or browse to the file **Aspnet_isapi.dll**. This file is normally located here: %windir%\Microsoft.NET\Framework\version\version v2.0.50727 or later should work.
thus: %windir%\Microsoft.NET\Framework\v2.0.50727\aspnet_isapi.dll
For Windows 2003 64 bit or 64 bit Windows XP you will need to reference the 64 bit version of aspnet like so:
%windir%\Microsoft.NET\Framework64\v2.0.50727\aspnet_isapi.dll
- j) In the **Extension** box, type the file-name extension, such as .jsp or .cfm. If you need to enable servlets, you probably will need to add a wildcard map as well (see example below). The easiest way to do this is to create a virtual directory using your IIS6 manager UI that points to your Tomcat application's servlets path. You may have to experiment with the "Verify that files exist" option. Commonly you should leave it checked.
- k) Click **OK**

Global Deployment in Global Assembly Cache:

GLOBAL DEPLOYMENT IS A COMPLETELY OPTIONAL STEP. IT IS NOT REQUIRED. YOU WILL NEED TO BE AN EXPERT ON WINDOWS SOFTWARE DEPLOYMENT. NO SUPPORT IS AVAILABLE FOR THIS OUTSIDE THIS SECTION.

The automated installer will perform a global deployment of the connector on IIS7 and above when you select the “All IIS Sites” option. This has the advantage that even new sites that you create will inherit the connector without any special steps.

You have the option with version 0.9.2.8 onward to deploy the connector one time per server, rather than one time for each web site. If you want to do this manually, you have to add the dll files to the windows global assembly cache (GAC). The libraries are signed and have strong names so are ready to be deployed in such a fashion.

You should be familiar with the process of adding assemblies to the GAC. This involves special tools that are outside the scope of this manual. You can either use the gacutil.exe or the .NET Framework Configuration Tool (Mscorcfg.msc), both of these are separate downloads.

Once you have added the libraries to the GAC, you will need to change the “Type:” reference in the Managed Handler or web.config file like so:

from:

```
BonCodeIIS.BonCodeCallHandler
```

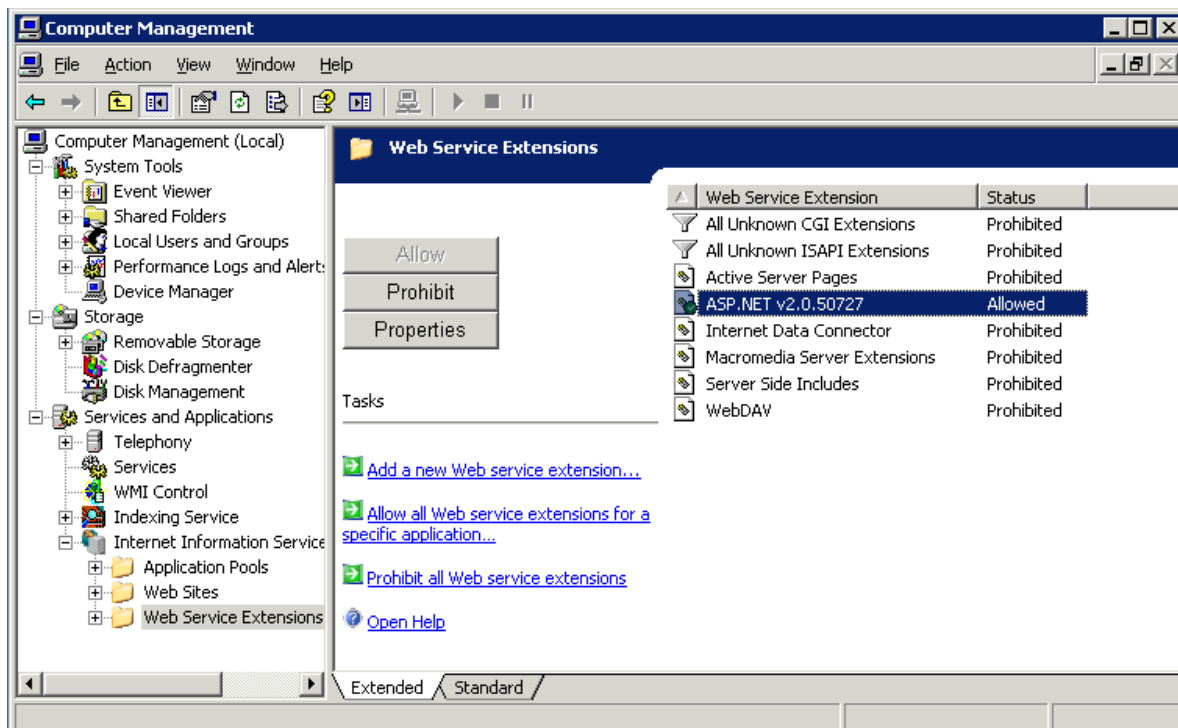
to:

```
BonCodeIIS.BonCodeCallHandler, BonCodeIIS, Version=1.0.0.0,  
Culture=neutral, PublicKeyToken=ad590a40d40745cf
```

Also, the default directories for the setting and log file are changed to the system windows directory, normally c:\windows. If a setting file is placed there it will become effective for all IIS sites.

You can still perform a site specific install after you installed the connector in global mode. The site specific install will override the global properties.

ASP.NET Web Services Extension Enabled on IIS6:



Wildcard mappings:

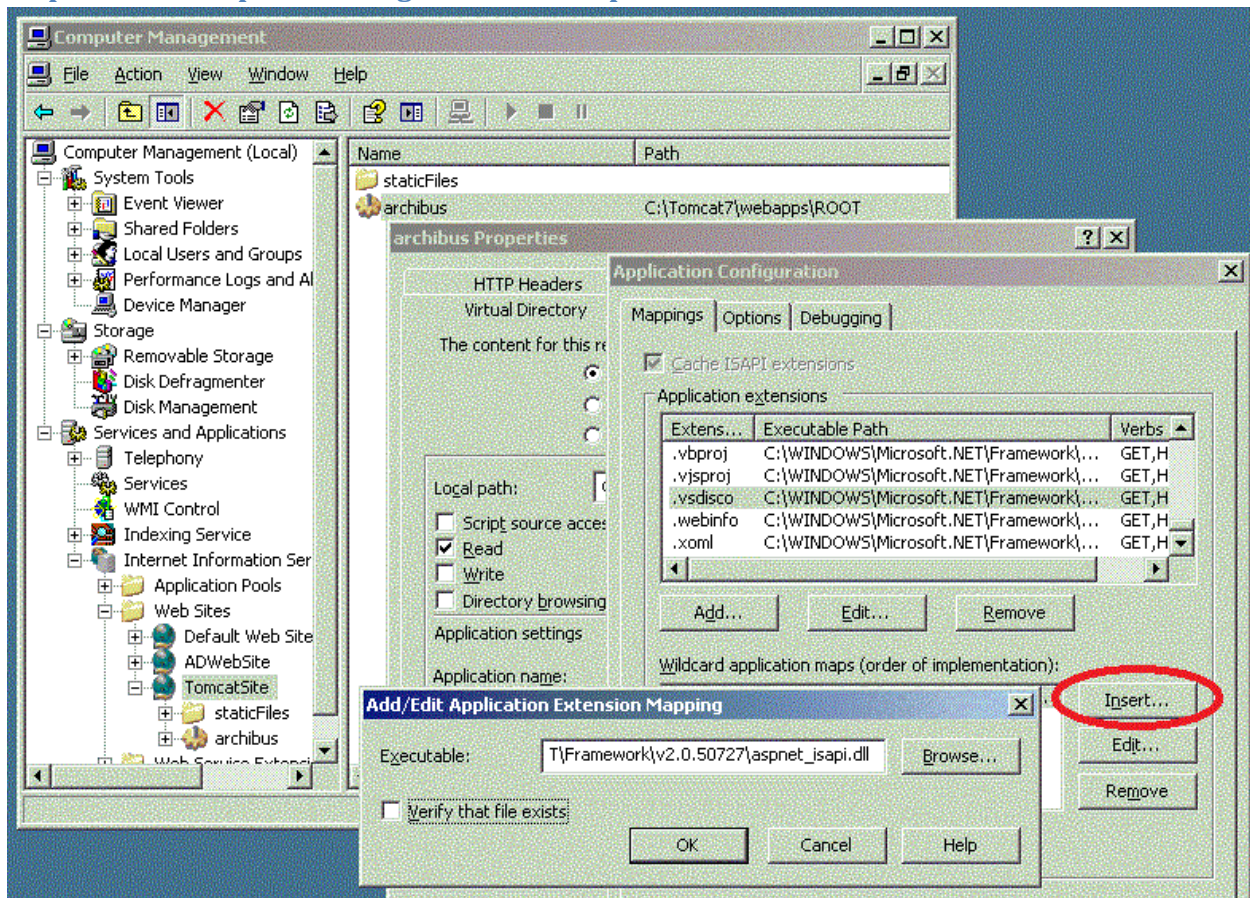
If you are using Java servlets or want to designate either a whole site or a site- path in IIS to forward every request with a matching URI pattern to tomcat for processing you will need to add Wildcard mappings to IIS. This will also need to be done for SES URLs. However, this is simple to do.

Here are examples of Wildcard mappings. Additional videos are available online/youtube that walk you the process step-by-step.

The wildcard mapping for IIS6 has two steps, i) Using the IIS6 UI and ii) changing the web.config file.

The wildcard mapping for IIS7 can be done in one step either via the UI or web.config changes. The example shows the use of UI.

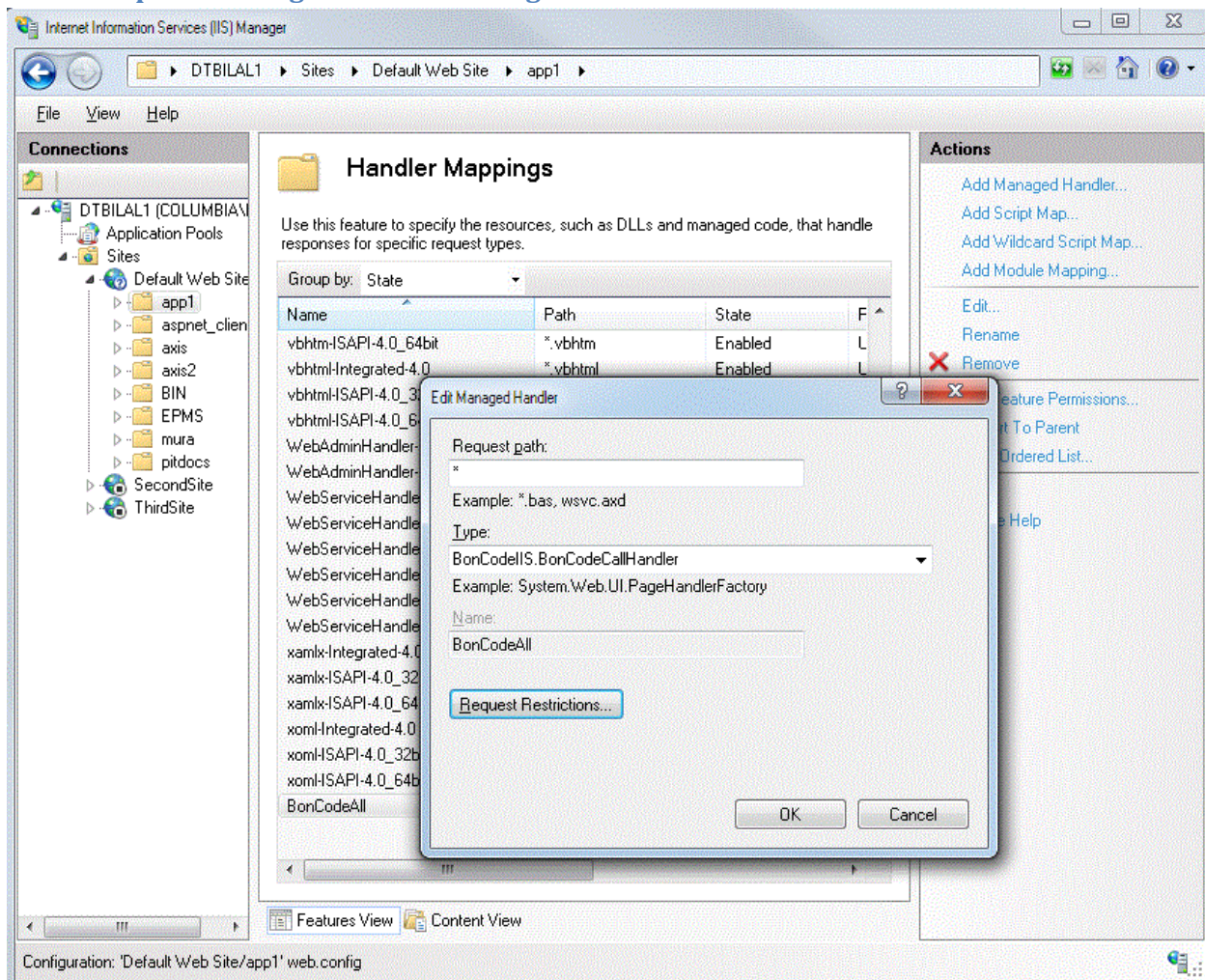
Step 1: IIS6 Example of adding a wildcard map



Step 2: To complete the wild-card map for IIS6 you will also need to change the web.config file. Add this line as last line of your <httpHandlers> section:

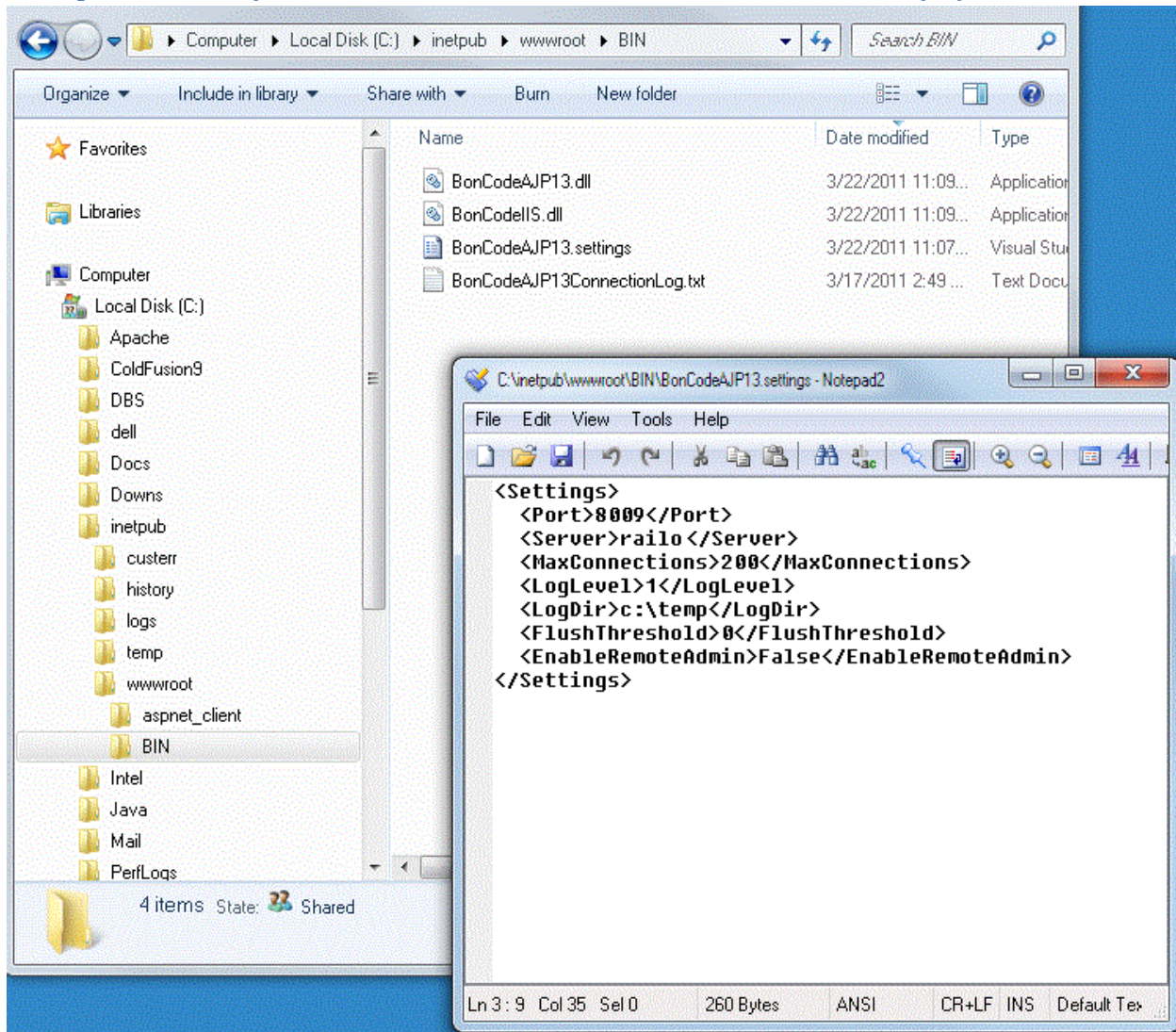
```
<add verb="*" path="*" type="BonCodeIIS.BonCodeCallHandler"/>
```


IIS7 Example of adding a wildcard managed handler



The IIS7 configuration does not require the change of any configuration file directly. You can do the mapping within IIS Administrator UI. In the example case the overall site already has the connector installed under the BIN directory. Any subdirectory can inherit it. In this case all requests for any part of the “app1” directory will be passed on to tomcat. The actual documents/files do not need to be visible to IIS for this to work correctly.

Example of correctly installed files and directories on IIS on a Site basis (all):



In this case the connection log file would be created in the `C:\temp` directory.

OPTIONAL STEPS (INTERNATIONAL CHARACTERS)

SKIP THIS STEP IF YOU JUST INSTALLED APACHE TOMCAT AND YOU DO NOT NEED UTF-8 SUPPORT. THERE IS NOTHING FOR YOU TO DO. THE DEFAULT INSTALLATION OF TOMCAT HAS ALL YOU NEED.

But, if you already have manually configured your Apache Tomcat instance to disable the AJP protocol, you will need to re-enable it. You should consult the Apache Tomcat manuals if you need to make these changes.

You do not need to worry about making changes to `worker.properties` or `uriworkermap.properties`, these files are not used with the BonCode connector.

Apache Tomcat configuration only requires that you enable the AJP protocol in the `server.xml` file. This is the same place internationalization is configured, e.g. you use URLs with double byte characters. To make this work you will need to configure UTF-8 support on the Tomcat side. UTF-8 encoding is enabled by default on the BonCode connector side.

```
<Connector port="8009" URIEncoding="UTF-8" enableLookups="false"
redirectPort="8443" protocol="AJP/1.3" />
```

This allows Apache Tomcat to accept and translate characters in UTF-8 format in the Uri and Query parameters. If you work with UTF-8 character anywhere else the BonCode connector will still transfer this correctly but there will be no automatic translation on the Apache Tomcat side. You will have to correctly decode the traffic before you can use it in your application.

Java example of retrieving a custom header containing UTF-8 characters:

```
String pathInfo = request.getHeader("path-info");
String decodedPath = new String(pathInfo.getBytes(), "UTF-8");
```

Sample web.config file (IIS6 only)

The following is a sample file that can be used in IIS6 managed handler configurations. It will prepare IIS6 to send jsp and cfm requests to tomcat via the connector.

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <system.web>
    <httpHandlers>
      <add verb="*" path="*.jsp" type="BonCodeIIS.BonCodeCallHandler"/>
      <add verb="*" path="*.cfm" type="BonCodeIIS.BonCodeCallHandler"/>
    </httpHandlers>
  </system.web>
</configuration>
```

The settings file (BonCodeAJP13.settings)

This file (BonCodeAJP13.settings) needs to be located in the same directory as the library (dll) files. By default this is most likely the c:\inetpub\wwwroot\BIN directory. In global (server wide) install, this file will be located in the Windows directory of your server, most likely c:\windows. You can configure several settings for the connector. All of these are optional, thus this file is not required for the connector to work. The setting file is a simple XML formatted file and contains the following keys.

Setting Name	Impact
Port	Default: 8009. You can set a different port for AJP communication.
Server	Default: localhost. If the Apache Tomcat server is located on a different server you can set the server IP or DNS name here.
MaxConnections	Default 200. How many connections are allowed through this channel at one time. This should correspond to Max JK threads on the tomcat side. If you set this to zero, the connector will stop maintaining a connection pool and instead will re-connect for every request.
Log Level	Default 0. Determines what will be logged by the connector. Log levels are 0=none, 1=Errors and Exceptions only, 2=Basic (startup/shutdown/error), 3=Headers (including HTTP Headers), 4=Debug and packet Contents. If you set logging above 2 the log file can get large quickly; please monitor connector carefully. Also ensure that IIS has write permission in the designated Log Directory (see below).
LogDir	Default: installation directory of connector or when using global install the Windows system directory (c:\Windows). Determines in which path the connector will create a log file. Please ensure that the IIS_IUSRS group has permission to read/write in the designated directory and that the directory exists. You will see an error in your browser otherwise. You do not need to specify a LogDir setting for logging to work.
LogFile	Default: BonCodeAJP13ConnectionLog.txt. This is the name of the log file that will be populated with log data. If you have multiple instances of the connector, e.g. multiple sites running on IIS with individual instances of connector, you should either specify a different file name or directory to avoid log file contention errors. You do not need to specify a LogFile for logging to work.
EnableRemoteAdmin	Default: True. Whether remote administration of Apache Tomcat or railo servers is allowed. Will block access to certain URI signatures if set to False.
FlushThreshold	Default: 0. This is the count in time-ticks that determine whether a flush has occurred on the Apache Tomcat side. If we receive a partial response after this many tick counts, we will assume a flush is desired and thus the webserver should start transferring the output cache to the browser. A reasonable tick count for this setting is 20,000 or more. This allows a response to browser while processing is still on-going. You can experiment based on your expected processing time. If you set this to zero, content flushing will be disabled. With flushing disabled we will only transfer content to browser after all of tomcat processing is completed. If you experience problems with content transfer (garbled text, or image

EnableHeaderDataSupport	files not transferring it is recommended to turn this feature off). This is for dynamic host configuration on Tomcat side working with the special mode_cfml_valve open source project by Jordan Michaels (http://www.modcfml.org/). The connector will insert the x-tomcat-docroot http header into the communication stream.
HeaderWhitelist	If this setting is enabled, only HTTP headers on this comma separated value (CSV) list will be transferred to tomcat. These headers have to be in the IIS specific notation (most likely with HTTP_ prefix). Follow this URL for default list: http://msdn.microsoft.com/en-us/library/ms524602(v=vs.90).aspx Example of valid headers are: HTTP_CONNECTION,CONTENT_LENGTH,HTTP_ACCEPT,HTTP_ACCEPT_ENCODING,HTTP_ACCEPT_LANGUAGE,HTTP_COOKIE,HTTP_HOST,HTTP_USER_AGENT
HeaderBlacklist	CSV list of HTTP headers that will not be transferred to Tomcat even if populated. These will need to be expressed in valid IIS specific notation (see above).
ReadTimeOut	Number of milliseconds to wait before we stop listening for more traffic. 1000ms=1s. Default: 120000 (20 min).
WriteTimeOut	Number of milliseconds to wait until we stop writing to tomcat. 1000ms=1s. Default: 30000 (30s).
ForceSecureSession	Forces SSL to tomcat; forces secure jsession cookie. Standard HTTP connection will be accepted but responses will be redirected to secure port. True/False. Default: False.
ResolveRemoteAddrFrom	If another HTTP header contains valid IP instead of REMOTE_ADDR, it should be provided here. The connector will extract the first IP from list of IPs in the alternate header. Set to blank if this is to be disabled. Commonly the HTTP_X_FORWARDED_FOR header is the standard populated header when intermediaries such as load balancers or proxy servers are used. In those cases REMOTE_ADDR contains the intermediary's IP rather than the client's IP. This feature corrects for this. Defaults to blank (disabled).
AllowEmptyHeaders	By default the connector only sends HTTP headers that contain a value. If you need to see all headers all the time, you need to change this to True. Default: False.
PathInfoHeader	PATH-INFO HTTP header is not implemented in Apache Tomcat AJP acceptor. The connector will send path info in this header instead. If set to blank this will be disabled. Default: xajp-path-info
TomcatConnectErrorURL	If Tomcat is not available, e.g. shutdown, the user will be redirected to URL provided in this setting. If this setting is not set, a default message is displayed instead.
EnableHTTPStatusCodes	True or False. HTTP status codes that are set by Tomcat will be set for IIS. Default: True
TCPClientErrorMessage	If TCP communication is experiencing an error, a default message is displayed to users. An alternate message can be provided in this setting. Default: blank.

TCPStreamErrorMessage	If there is a problem with the TCP stream, a default message is displayed to users. An alternate message can be provided in this setting. Default: blank.
PathPrefix	<p>To make it easier to map Tomcat deployed application to the root of IIS sites you can specify a path prefix. A path prefix has to start with a forward slash (/). For example if you deployed axis2 as war archive in tomcat webapps directory, it would be available for IIS using a URL pattern such as http://[server]/axis2. If the desire is to map it to the site root of a new IIS (e.g. http://[server] or http://web.service.mysite.com) then a PathPrefix setting must be set for the translation to take place between Tomcat and IIS.</p> <p>In this example a PathPrefix of <code>"/axis2"</code> can be used on the IIS site's connector:</p> <pre><PathPrefix>/axis2</PathPrefix></pre> <p>Using the connector IIS will redirect all inbound request from <code>"/"</code> to <code>/axis2</code> on tomcat transparently.</p> <p>Default: blank</p>
PacketSize	<p>Integer</p> <p>This defines the size of packets exchanged between IIS and Tomcat in bytes. Normally there should be no need to change the default. However, in some cases, this results in errors when long SSL certificate chains are used.</p> <p>You will need to change the Apache Tomcat packetSize attribute on the AJP protocol to correspond to the PacketSize setting. The maximum value possible is 65536 bytes. Default: 8192.</p> <p>To change the BonCode connector, add or alter the PacketSize setting in the BonCodeAJP13.settings file, e.g.:</p> <pre><PacketSize>16384</PacketSize></pre> <p>Correspondingly in the Apache Tomcat conf directory, change the server.xml file for the AJP protocol and add or alter the packetSize attribute , e.g.:</p> <pre><Connector port="8009" protocol="AJP/1.3" redirectPort="8443 packetSize="16384"/></pre>
EnableAdobeMode	<p>True or False.</p> <p>This is a convenience setting that will switch the connector into supporting Adobe ColdFusion in "Server Install" scenarios. When switched into this mode, the connector will respond to non-standard extensions to the AJP protocol introduced by Adobe.</p> <p>This will also automatically override the following settings and values accordingly:</p> <p>PacketSize → 65536</p> <p>PathInfoHeader → path-info</p>

EnableClientFingerPrint

Default: False

True or False.

The connector will use a combination of HTTP header data to attempt to uniquely identify a client (browser). The result of this calculation is a base 64 encoded MD5 hash. It will be passed in separate HTTP header xajp-clientfingerprint.

The data in the xajp-clientfingerprint header can be used to identify invalid session access in web application (OWASP TOP 10 A-3).

Default: False

Example Settings file

```
<Settings>
  <Port>8009</Port>
  <Server>myTomcatServer</Server>
  <MaxConnections>5000</MaxConnections>
  <LogLevel>1</LogLevel>
  <LogDir>c:\temp</LogDir>
</Settings>
```

Using The Connector

The Issue with Path-Info

The Issue

Using an AJP based connector, like the BonCode connector, you have many advantages, including binary transfer of data, efficient reuse of connections, and SSL data support. However, one of the elements that causes repeated hiccups is the use of the PATH_INFO CGI variable. The AJP protocol currently does not have a method to transfer this information to Apache Tomcat in a native format. Thus references to *request.getPathInfo()* or *CGI.PATH_INFO* will return null or empty string in all cases.

The Solution

The current work-around the BonCode connector is offering to developers is that it will transfer the PATH_INFO data from IIS to Apache Tomcat via an alternate HTTP Header variable. This variable is labeled by default: **xajp-path-info**. You can change this, please see previous section on setting file on how to do this.

Several application servers (Railo (gettrailo.org), Open Blue Dragon (openBlueDragon.org)) automatically look for this alternate value and place it correctly into the CGI scope PATH_INFO. Thus, using one of these application servers on top of Tomcat, will avoid code changes.

If you cannot make any code changes, and you absolutely require the use of CGI.PATH_INFO you will have to use an alternate connector (HTTP) or layer a URL rewrite engine on top of AJP connector. Both of these options are outside the scope of this document.

RFC Compliance

If you have been using a different web server than IIS, you will also need to note, that the PATH_INFO data IIS supplies is not RFC compliant (<http://www.ietf.org/rfc/rfc3875>). Rather it is a repeat of the SCRIPT_NAME/PATH_INFO combination.

The BonCode connector, changes the xajp-path-info data automatically to be RFC 3875 compliant. If this is not desired, the connector also supplies the original IIS determined data in path-header http variable that can be used instead.

Using this example URL:

```
http://somehost.com/cgi-bin/somescript.jsp/this%2eis%2epath%3binfo?Qn=Yes
```

Using RFC compliant formatting the PATH_INFO would be: **/this.is.the.path;info**

Otherwise, the IIS default designation would be: somescript.jsp/this.is.the.path;info

Extending the Timeouts for Long Running Requests

I am including this for completeness as some users were wondering about available options.

If you have very long running requests you need to be aware that IIS has built in timeouts that you may need to adjust. For example the default execution timeout is 110 seconds.

If you need to extend this you need to change the appropriate web.config file in each site.

The *httpRuntime* section will need to be adjusted:

The complete option set for httpRuntime:

```
<httpRuntime useFullyQualifiedRedirectUrl="true|false"
    maxRequestLength="size in kbytes"
    executionTimeout="seconds"
    minFreeThreads="number of threads"
    minFreeLocalRequestFreeThreads="number of threads"
    appRequestQueueLimit="number of requests"
    versionHeader="version string"/>
```

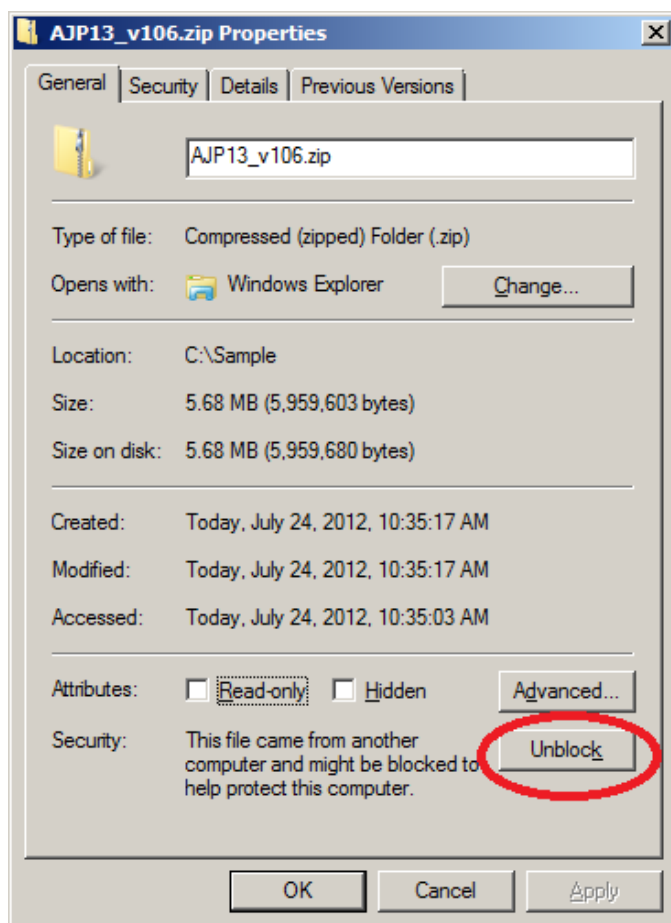
A sample web.config with httpRuntime execution timeout defined:

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <system.web>
    <httpRuntime
      executionTimeout="180"
    />
    <httpHandlers>
      <add verb="*" path="*.jsp" type="BonCodeIIS.BonCodeCallHandler"/>
      <add verb="*" path="*.cfm" type="BonCodeIIS.BonCodeCallHandler"/>
    </httpHandlers>
  </system.web>
</configuration>
```

General Recommendations

Unblock before use

Most likely you have received this package via internet download. To avoid multiple issues with using the zip file content I would recommend that you unblock the package before unzipping. This avoids security issues with Windows Vista and later operating systems as well as Windows 2008 and later server operating systems. Simply right click on the zip file and click “Unblock” on the “General” tab.



Connector Thoughts

The BonCode connector has been engineered for throughput in concert with IIS7. Thus, it will efficiently create and destroy threads; however, it will also maintain a larger number of threads for re-use than the, previous, ISAPI based connector. This will allow it to handle demand spikes gracefully.

Consequently, it is recommended that some of the Apache Tomcat default parameters are adjusted to allow higher level of concurrency in production as well.

For example, in the tomcat server.xml file add the maxThreads attribute to the AJP1.3 protocol like so:

```
<!-- Define an AJP 1.3 Connector on port 8009 -->  
<Connector port="8009" protocol="AJP/1.3" redirectPort="8443"  
maxThreads="1000" />
```

It is also recommend that IIS and Apache Tomcat point to the same document directories. Thus for each additional website that is added on IIS a similar entry need to be made in the Apache Tomcat configuration.

Setup Recommendations

This is my personal preference on setup of the Tomcat and IIS configurations.

I setup two new web-sites in IIS, before the installation of the connector. The first site is for deployed applications using Tomcat (regular application), the second one for tomcat administration.

Configuration of Application Site:

In this setup I point the IIS website's physical path to the Tomcat [tomcat]\webapps\ROOT folder. Thus Tomcat's references do not change. I can deploy all WAR files as usual and they should become available through IIS as apps as well. Depending on whether I need servlet support, I will go into the IIS GUI to add a wildcard mapping for a subfolder. For example, if I deployed axis2, the path axis2 and below will need a wildcard mapping.

Configuration of Admin Site:

This is a slightly more complex piece as I also setup an alternate DNS name so I can map a host header in IIS. For example, if I wish to access the tomcat administration through IIS, I do the following setup:

- Add a host reference in either host file (c:\windows\system32\drivers\etc) or in DNS, e.g. tomcatadmin.local
- Create new IIS web site with: Map IIS-physical path to [tomcat]\webapps
- Host header name: tomcatadmin.local

I enable wildcard mapping on the site level.

I have, now, options to use Windows integrated security or IP based security to control access to the admin site in addition to all other tools. And, I can access tomcat manager through IIS.

Configuration of the Application Pool:

I am also recommending a lower idle-timeout for the application pool. This will allow the connections to be recycled sooner. This is a good idea with low, medium load sites as it releases resources faster. You do this by accessing the "Advanced Settings..." of the application pool. In the "Process Model" section of the properties change the "Idle Timeout (minutes)" to 10.

Also in the same configuration section, the “identity” of the application pool should be changed to “NetworkService” to avoid the common “System.Web.AspNetHostingPermission” error that occurs with IIS 7.5 and beyond.

Identifying the installed Version

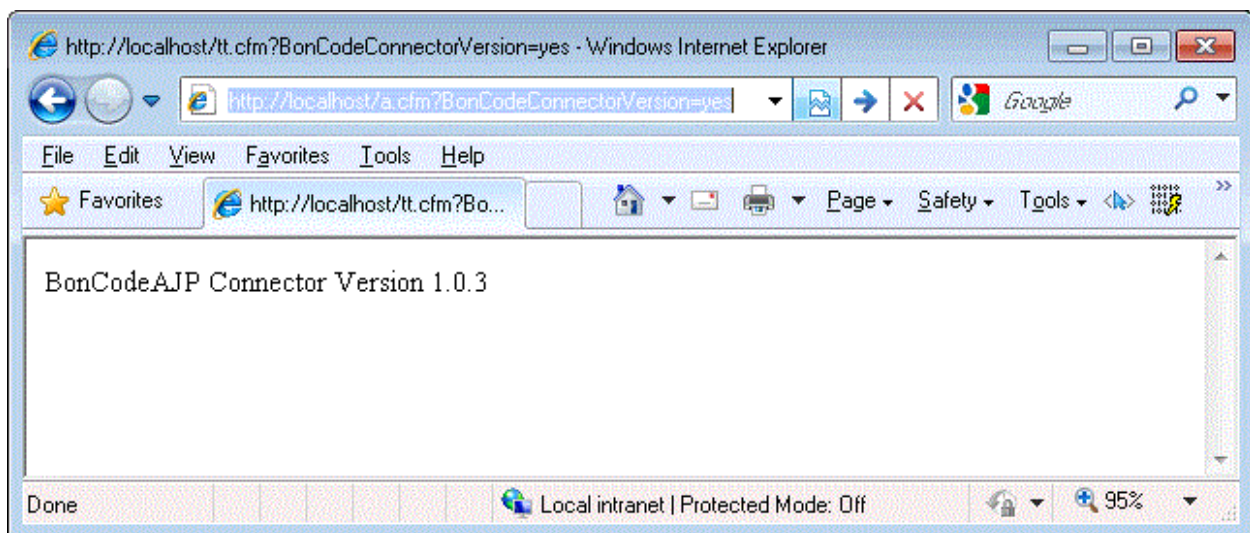
With version 1.0.3 and later, the BonCode connector will support a special URL parameter that can be used from a local IP address. The connector will show the installed version and the source of setting file, if any, to the user.

Add the `BonCodeConnectorVersion` Query Parameter to any request. The main body of the request will be ignored.

An example identifier call:

```
http://localhost/a.jsp?BonCodeConnectorVersion=true
```

This has no effect when used from a remote client.



Also to note is that when a log file is used (log level is set between 1 and 3) the version number will be appended to any system error in the log.

HTTP Headers and connector behavior

There are generally two streams of HTTP headers to consider. An inbound stream, the headers set by an HTTP client such as a browser in concert with headers set by the webserver, i.e. IIS, vs. an outbound stream, headers set by Apache Tomcat, as response to the initial client request. In the latter, the outbound stream, no changes are made by the connector. Everything is passed on as is. However, during the inbound stream some optimizations take place.

During the inbound phase, the connector normally passes through headers as they are provided by clients (browsers). There are exceptions though. You can set up black lists and white lists to restrict how the connector behaves in this regard. There are also headers that the connector will block by default and which cannot be controlled by any setting. These fall generally into two groups themselves. They are either IIS generated headers and are not assumed to be useful in the tomcat context, or they already exist in some form in the communication stream and are duplicated by IIS.

This is the list of always suppressed headers:

INSTANCE_META_PATH

APPL_MD_PATH

ALL_HTTP

ALL_RAW

AUTHORIZATION

CONNECTION

HTTP_CONTENT_TYPE

HTTP_CONTENT_LENGTH

PRAGMA

Inbound White Lists

When using white lists only headers on that list will be transferred regardless of how many headers the client actually submits to webserver. This can be adjusted in the setting file for the connector.

Inbound Black Lists

When using black lists, headers that are on this list will be suppressed and will not be transferred to Tomcat. This can be adjusted in the setting file for the connector.

Scripted Deployment

If you got this far, you have proven to have a curious mind. The information in this section is meant for users who want to deploy the connector on many machines and thus need to create some sort of automated way to do this. You can stop reading otherwise.

To deploy the BonCode connector in an automatic fashion across multiple servers you have two basic options. Firstly, you can use the guidelines provided in this manual installation steps section to build your installers and scripts. Secondly, you can use the distribution files and the installer provided.

We will focus on the second alternative. The installer provided (Connector_Setup.exe) can operate in silent mode to fully install the connector. In order to provide the inputs needed you will need to provide both command line parameters and a setting file (installer.settings) to the installer.

Installer Command Line Parameters

The recommended combination of command line parameters is:

```
/VERYSILENT /SUPPRESSMSGBOXES /LOG /SP- /NOCANCEL /NORESTART
```

Thus the full command line would most likely look like this:

```
Connector_Setup.exe /VERYSILENT /SUPPRESSMSGBOXES /LOG /SP- /NOCANCEL /NORESTART
```

The command line options in detail:

/VERYSILENT

Instructs Setup to be very silent. When Setup is silent the wizard and the background window, and the progress window are not displayed. Everything else is normal so for example error messages during installation are displayed and the startup prompt is (if you haven't disabled it with the '/SP-' command line option explained below) displayed as well.

If a restart is necessary and the '/NORESTART' command isn't used (see below) it will reboot without asking.

/SUPPRESSMSGBOXES

Instructs Setup to suppress message boxes. Only has an effect when combined with '/VERYSILENT'.

The default response in situations where there's a choice is:

- Yes in a 'Keep newer file?' situation.
- No in a 'File exists, confirm overwrite.' situation.
- Abort in Abort/Retry situations.
- Cancel in Retry/Cancel situations.

- Yes (=continue) in a DiskSpaceWarning/DirExists/DirDoesntExist/NoUninstallWarning/ExitSetupMessage/ConfirmUninstall situation.

- Yes (=restart) in a FinishedRestartMessage/UninstalledAndNeedsRestart situation.

/LOG

Causes Setup to create a log file in the user's TEMP directory detailing file installation and [Run] actions taken during the installation process. This can be a helpful debugging aid. For example, if you suspect a file isn't being replaced when you believe it should be (or vice versa), the log file will tell you if the file was really skipped, and why.

The log file is created with a unique name based on the current date. (It will not overwrite or append to existing files.)

The information contained in the log file is technical in nature and therefore not intended to be understandable by end users. Nor is it designed to be easily parseable; the format of the file is subject to change without notice.

/SP-

Disables the This will install... Do you wish to continue? prompt at the beginning of Setup.

/NOCANCEL

Prevents the user from cancelling during the installation process, by disabling the Cancel button and ignoring clicks on the close button. Useful along with '/VERYSILENT'.

/NORESTART

Prevents Setup from restarting the system following a successful installation, or after a Preparing to Install failure that requests a restart. Typically used along with /VERYSILENT.

Installer Parameters File (installer.settings)

Most likely in addition to the command line parameters you will want to provide installation parameters for the installer to use. These will need to be placed in a file with the designated name of “installer.settings” and located in the same directory as the executable file (Connector_Setup.exe). The format of this file follows a standard Windows “.ini” file structure, i.e. it is a text file with section delineated by headers (in square brackets).

If the installer finds such a file during execution it will read it and determine the parameters from the data contained in the file. All sections and keys are optional. If not provided the default values are assumed.

DON'T LEAVE AN ACTUAL “INSTALLER.SETTINGS” FILE IN THE SAME DIRECTORY AS THE INSTALLER IF YOU DO NOT MEAN TO USE IT. IF YOU START YOUR SETUP WHEN A PARAMETER FILE IS PRESENT, USER INPUTS WILL BE IGNORED.

A sample installer.settings file

```
[Setup]
installType=global
acceptLicense=1
enableRemote=0

[Handlers]
installCF=1
installJSP=1
installWildcard=0

[Tomcat]
server=localhost
ajpPort=8009
configureServerXml=0

[Settings]
allowEmptyHeaders=1
```

The Setup Section Options

Parameter Name	Impact
installType	Default: global. Options: global or sites Either “global” or “sites”. If you specify global, the installer will install connector server wide for all IIS sites found. The connector libraries will also be installed in Global Assembly Cache (GAC). If you specify “sites”, you will also need to add a [Sites] section to the parameter file. The installer will, then, determine by site order which IIS

acceptLicense	sites will have the connector installed.
	Default:1 Options: 0 or 1 (0=No, 1=Yes) Indicates that you accept license.
enableRemote	Default:1 Options: 0 or 1 (0=No, 1=Yes) Indicates whether you wish the connector to allow remote administration of tomcat or Railo servers through this channel.
uninstallDir	Default: [blank] The directory in which uninstall logs and references should be placed. If left blank the installer will determine this based on operating system automatically. A valid existing path should be used.

The Handlers Section Options

Parameter Name	Impact
installCF	Default:1 Options: 0 or 1 (0=No, 1=Yes) Installs connector support for cfm/cfc pages.
installJSP	Default:1 Options: 0 or 1 (0=No, 1=Yes) Installs connector support for JSP pages.
installWildcard	Default:0 Options: 0 or 1 (0=No, 1=Yes) Install wild card connector. All traffic will be passed to Tomcat.

The Tomcat Section Options

Parameter Name	Impact
server	Default: localhost Name of server on which Tomcat instance is installed.
ajpPort	Default:8009

configureServerXml	Port on which Tomcat is listening to AJP traffic
	<p>Default:0</p> <p>Options: 0 or 1 (0=No, 1=Yes)</p> <p>If Apache Tomcat is installed on the local server as Windows service and you have opted to do a global install, then the installer can also allow an automatic configuration of Tomcat.</p> <p>It will change the Tomcat server.xml file. It will redirect the webapps directory to the same directory as the Default IIS site's physical path (normally c:\inetpub\wwwroot). It will also copy any files contained in the existing Tomcat webapps directory to the IIS site's directory.</p>

The Settings Section Options

Parameter Name	Impact
allowEmptyHeaders	<p>Default:0</p> <p>Options: 0 or 1 (0=No, 1=Yes)</p> <p>The connector does not transfer HTTP headers that do not contain data. If you like to change it during setup provide this in parameter file.</p>
forceSecureSession	<p>Default:0</p> <p>Options: 0 or 1 (0=No, 1=Yes)</p> <p>Will set the ForceSecureSession option. See settings file description for impact of this selection</p>
enableHeaderDataSupport	<p>Default:0</p> <p>Options: 0 or 1 (0=No, 1=Yes)</p> <p>Enable Header Data support setting (set to True). See setting description for impact of this selection.</p>
enableFlush	<p>Default:0</p> <p>Options: 0 or 1 (0=No, 1=Yes)</p> <p>Enable IIS based HTTP flushing. The contents of cache are send to browser. This will set the setting for Flush Threshold</p> <p><FlushThreshold>50000</FlushThreshold></p>
enableRemoteIpDetection	<p>Default:0</p> <p>Options: 0 or 1 (0=No, 1=Yes)</p> <p>Enable translation of headers to determine the correct client IP address behind load balancers. This will set the setting for Resolve Remote Addr like so:</p> <p><ResolveRemoteAddrFrom>HTTP_X_FORWARDED_FOR</ResolveRemote</p>

The Sites Section Options

The [Sites] section is a special section that is available when the “installType=sites” is used. This is a dynamic section containing an ordered list of entries. It contains a site reference and a flag (0 or 1) to indicate whether the connector should be installed. It is important to know the order of the sites setup in IIS to correctly supply information. For example, the third IIS site would be referenced as “Site3”. In this schema the IIS Default Site will always be referenced as “Site1”.

In the following example we would setup the connector for Site 1 and Site 6:

```
[Sites]
Site1=1
Site6=1
```

You may omit sites for which you do not want to setup the connection. We recommend that you specify at least one site though.

Troubleshooting

No page response	<p>You are not getting a response via IIS, but when accessing tomcat web- port the page is displayed correctly.</p> <p>➔</p> <p>You may have run into the maxThreads restrictions for tomcat, please see above for general recommendations. You should ensure that both tomcat and the BonCode AJP connector use the same maximum connections.</p>
J2EE Sessions dropped	<p>It appears like all your requests are creating new sessions, rather than reusing the session that is present.</p> <p>➔</p> <p>Check whether IIS is using keep-alive directive. On IIS7: IIS Admin select server > HTTP Response Headers > Actions > Set Common Headers > check Enable HTTP keep-alive (by default) > ok</p>
<p>Error Message:</p> <p>No connection could be made because the target machine actively refused it 127.0.0.1:8009</p>	<p>Tomcat may not be listening/ running on the assigned machine and port.</p> <p>➔</p> <p>Start tomcat, or if you need to change the target machine and port to connect to you will need to add a settings file (BonCodeAJP13.settings) normally into the /BIN directory. See above for instructions.</p>
<p>Error Message:</p> <p>HTTP Error 500.21 - Internal Server Error</p> <p>Handler "BonCode-Tomcat-CFM-Handler" has a bad module "ManagedPipelineHandler" in its module list.</p>	<p>SEVERAL ORIGINS ON THIS ERROR:</p> <p>i)</p> <p>You are not able to retrieve mapped extensions from IIS7.</p> <p>➔</p> <p>You may not have installed the ".net extensibility" for IIS7. See above for instructions or use the Server Manager GUI to install this IIS7 feature.</p> <p>ii)</p> <p>You have installed a later version of the .net framework, for example .net version 4.0, before you installed the connector and .net version 3.51. If you also have ISAPI role features enabled, you may have created a conflict to resolve.</p> <p>➔</p> <p>To keep using ISAPI and .net version 4:</p> <p>a) Re-register asp.net under the correct .net version, run command: <code>C:\Windows\Microsoft.NET\Framework64\v4.0.30319\aspnet_regiis.exe -i</code></p> <p>b) Switch the Application pool connected to your site to "integrated mode" and the .net version 4</p>

	<p>iii) Conflict with existing .net framework setup and ISAPI Filters and ISAPI Extensions Role Sub service.</p> <p>➔ If you do not need ISAPI support (the connector does not need this with .net 3.5.1), you can remove ISAPI extensions and filters. Remove: ISAPI Extension, ISAPI Filter support</p> <p>iv) Conflict between .net frameworks installed. If you do not use .net 4: ➔ a) uninstall .net 4, b) uninstall .net 3.51, c) then, rerun the connector installer</p>
<p>Error Message: <i>Could not load type 'BonCodeIIS.BonCodeCallHandler'.</i></p>	<p>Most likely you have registered the handlers correctly but probably either not created a /BIN subdirectory under your website root and/or forgot to copy the library files (BonCodeAJP13.dll and BonCodeIIS.dll) into it.</p> <p>➔ Double check your webroot and ensure that you have the BIN subdirectory. Double check that library files (BonCodeAJP13.dll and BonCodeIIS.dll) are in your BIN subdirectory.</p>
<p>Error Message: Security Exception Description: The application attempted to perform an operation not allowed by the security policy. To grant this application the required permission please contact your system administrator or change the application's trust level in the configuration file.</p> <p>Exception Details: System.Security.SecurityException: Request for the permission of type 'System.Web.AspNetHostingPermission, System, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089' failed.</p>	<p>You are using IIS7.5 on Windows 2008 R2 or later. A change was introduced that changes security requirements for dll files to be deployed.</p> <p>You will need to unblock the BonCodeIIS.dll and BonCodeAJP13.dll libraries. See this blog for details: http://blogs.msdn.com/b/friis/archive/2010/06/09/system-security-securityexception-request-for-the-permission-of-type-system-web-aspnethostingpermission-failed.aspx</p> <p>Simply browse to the files in your webfolder, right-click, then, click the "Unblock" button on the General-Tab. See example earlier in this document.</p> <p>Or, you can change the Application Pool Identity back to "NetworkService" which was used in IIS7.</p>

Advanced Settings

(General)

.NET Framework Version	v2.0
Enable 32-Bit Applications	True
Managed Pipeline Mode	Integrated
Name	SecondSite
Queue Length	1000
Start Automatically	True

CPU

Limit	0
Limit Action	NoAction
Limit Interval (minutes)	5
Processor Affinity Enabled	False
Processor Affinity Mask	4294967295

Process Model

Identity	NetworkService
Idle Time-out (minutes)	20
Load User Profile	False
Maximum Worker Processes	1
Ping Enabled	True
Ping Maximum Response Time (seconds)	90
Ping Period (seconds)	30
Shutdown Time Limit (seconds)	90
Startup Time Limit (seconds)	90

Process Orphaning

Enabled	False
Executable	

Name
[name] The application pool name is the unique identifier for the application pool.

OK Cancel

Microsoft MVC:
You are receiving a 404 message and have Microsoft MVC installed prior to installing the connector.

By default the MVC handler will receive all communication and thus the BonCode connector will be ignored.
You will have to add the "IgnoreRoutes" directive to your code.
For example:

```
public static void RegisterRoutes(RouteCollection routes)
{
    routes.Ignore("{resource}.jsp/{*pathInfo}");
}
```

Or

```
routes.IgnoreRoute("{resource}.jsp/{*pathInfo}");
```

<p>Invalid content length. Last header processed [HTTP_COOKIE]. Please reconfigure BonCode Connector and Apache Tomcat to allow larger transfer packets. Your max allowed content length is 8192 bytes. Provided content length would be at least 8422 bytes.</p>	<p>A message similar to this (different bytes) indicates that the transfer packet size used to communicate information between Apache Tomcat and IIS is too small.</p> <p>Most likely issue here is the use of long certificate chains with SSL enabled or encountering of larger than allowed client cookies.</p> <p>You have two principal options:</p> <ol style="list-style-type: none"> you should limit the size of client cookies used and/or clear existing ones change the packet size <p>To change packet size, you will need to adjust both the BonCode connector as well as the Apache Tomcat side.</p> <p>To change the BonCode connector, add or alter the PacketSize setting in the BonCodeAJP13.settings file, e.g.:</p> <pre><PacketSize>16384</PacketSize></pre> <p>Correspondingly in the Apache Tomcat conf directory, change the server.xml file for the AJP protocol and add or alter the packetSize attribute , e.g.:</p> <pre><Connector port="8009" protocol="AJP/1.3" redirectPort="8443 packetSize="16384"/></pre>
<p>Upload Error 500</p>	<p>You may receive this error message when you attempt to upload files that are larger than 4MB in size. This error is not related to the connector but is included here for completeness.</p> <p>IIS has a built in size limit for file uploads which is set to 4096 KB / 4MB. You can change this either in machine.config file for the whole server or web.config file for the individual web site.</p> <p>For example in web.config, change the maxRequestLength attribute of httpRuntime tag. In this example setting it to 8MB:</p> <pre><system.web> <httpRuntime executionTimeout="90" maxRequestLength="8096" useFullyQualifiedRedirectUrl="false" minFreeThreads="8" minLocalRequestFreeThreads="4" appRequestQueueLimit="100" enableVersionHeader="true" /> ...</pre>

<p>On screen display of: "Error in transfer of data from tomcat to browser."</p>	<p>You may have enabled Flush Detection, but set a value that is too low for the <FlushThreshold> setting. IIS will only allow that we transfer HTTP headers one time. Setting a low Flush Threshold will cause the headers, even if partial, to be streamed to browser and when you receive additional headers, this message is displayed. Increase the value, recommended is in 10000 tick increments until the message goes away.</p> <p>If the FlushThreshold was zero when this message was received there was an issue with the protocol order of packets. Please look at updates to connector and or Apache Tomcat update to resolve this issue.</p>
<p>Repeated on screen display of: "Connection to Tomcat has been closed. Tomcat may be restarting. Please retry later."</p>	<p>Two scenarios:</p> <p>a) If this message does not go away after you retry (and you are certain Tomcat is configured and listening on AJP port), you may have a connector configuration issue. Most likely is a mismatch of .net framework version. Please set the application pool for your IIS site to the latest .net version that is installed on the server and retry. If this does not resolve the issue, enable logging with level 1 or higher and report the exception to the support site (tomcat@iaforge.org)</p> <p>b) This message occurs occasionally and you have not restarted Tomcat. Please ensure that you have not set a connectionTimeout parameter on the AJP connection definition in Tomcat's server.xml that is smaller than the Application Pool idle timeout in IIS (Advanced settings). By default the IIS timeout for this is 20 minutes. If the Tomcat timeout occurs before the IIS timeout it will cause the channel to be shut down unnecessarily. Recommended is that you remove the timeout on the Tomcat side:</p> <pre><Connector port="8009" protocol="AJP/1.3" redirectPort="8445" tomcatAuthentication="false" maxThreads="500" connectionTimeout="60000" /></pre>

Q&A

Feel free to contact me on my blog for questions or comments. If you want a mailed response, please use the <http://tomcatiis.riaforge.org> site to contact the project so I can mail you back.

Best of luck,

Bilal