

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 by Bilal Soylu

Table of Contents

BonCode AJP 1.3 Connector	3
License.....	3
The Beginning	3
BonCode Connector Solution.....	3
Prerequisites	4
Note on http headers.....	4
The Package Content	4
How to Install Automatically.....	5
How to Install Manually	5
ASP.NET Web Services Extension Enabled on IIS6:.....	8
Wildcard mappings:	8
IIS6 Example of adding a wildcard map	9
IIS7 Example of adding a wildcard managed handler	10
Example of correctly installed files and directories on IIS (all):	11
OPTIONAL STEPS	12
Sample Tomcat worker.properties file:	12
Sample Tomcat uriworkermap.properties file:.....	12
Sample web.config file (IIS6 only)	13
The settings file (BonCodeAJP13.settings).....	14
Example Settings file	15
General Recommendations	15
Setup Recommendations.....	16
Scripted Deployment	18
Installer Command Line Parameters.....	18
Installer parameters file (installer.settings).....	20
Troubleshooting.....	23
Q&A.....	25

BonCode AJP 1.3 Connector

License

The information and the related code are distributed under the Creative Commons v.3 license (<http://creativecommons.org/licenses/by/3.0/>). 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 or 7 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.

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
- no IIS6 vestiges or backward compatibility elements needed on IIS7
- all managed code .net extensibility framework
- works on IIS5.1, IIS6 and IIS7
- speed improvements
- easier control by file type on IIS side
- 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 the dot-net framework 3.5 installed on your machine. This comes as 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.

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. All header names 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 3.5 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

Start the Connector_Setup.exe program within the distribution directory and follow prompts. The automated setup currently assumes that your tomcat server is located on the same machine as IIS and that it communicated on port 8009.

If this is not accurate you can still use the automated installation, you can make changes using the setup screens. You can also add or change the settings file manually within the “/BIN” directory of your website. See below for details on the settings file.

How to Install Manually

The installation consists of several steps; most of these steps are defaulted and are only here for completeness. For example on tomcat 6 and 7, all the tomcat side 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):

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

- 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.

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

Paste the next line as one line into the command prompt after you navigated to the inetsrv directory:

```
appcmd set config /section:handlers /+[name='BonCode-Tomcat-JSP-Handler',path='*.jsp',verb='*',type='BonCodeIIS.BonCodeCallHandler',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
- 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.

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:

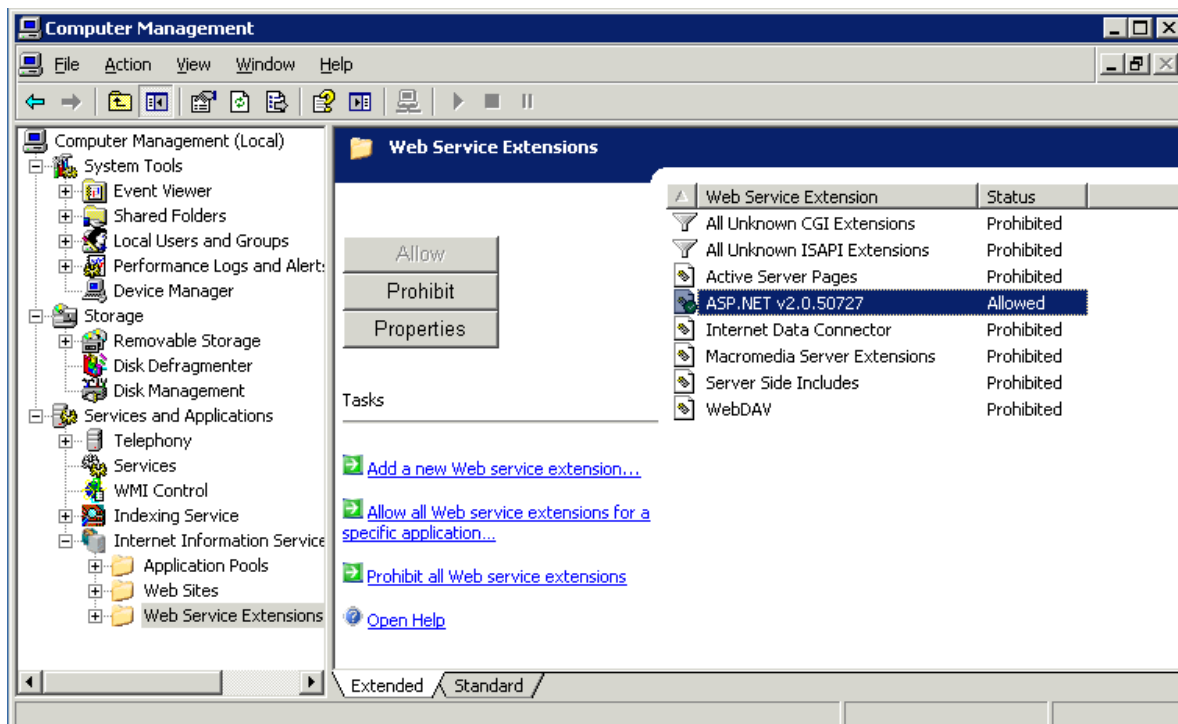
`BonCodeIIS.BonCodeCallHandler`

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 root directory, normally c:\windows\system32. If a setting file is placed there it will become effective for all IIS sites.

ASP.NET Web Services Extension Enabled on IIS6:

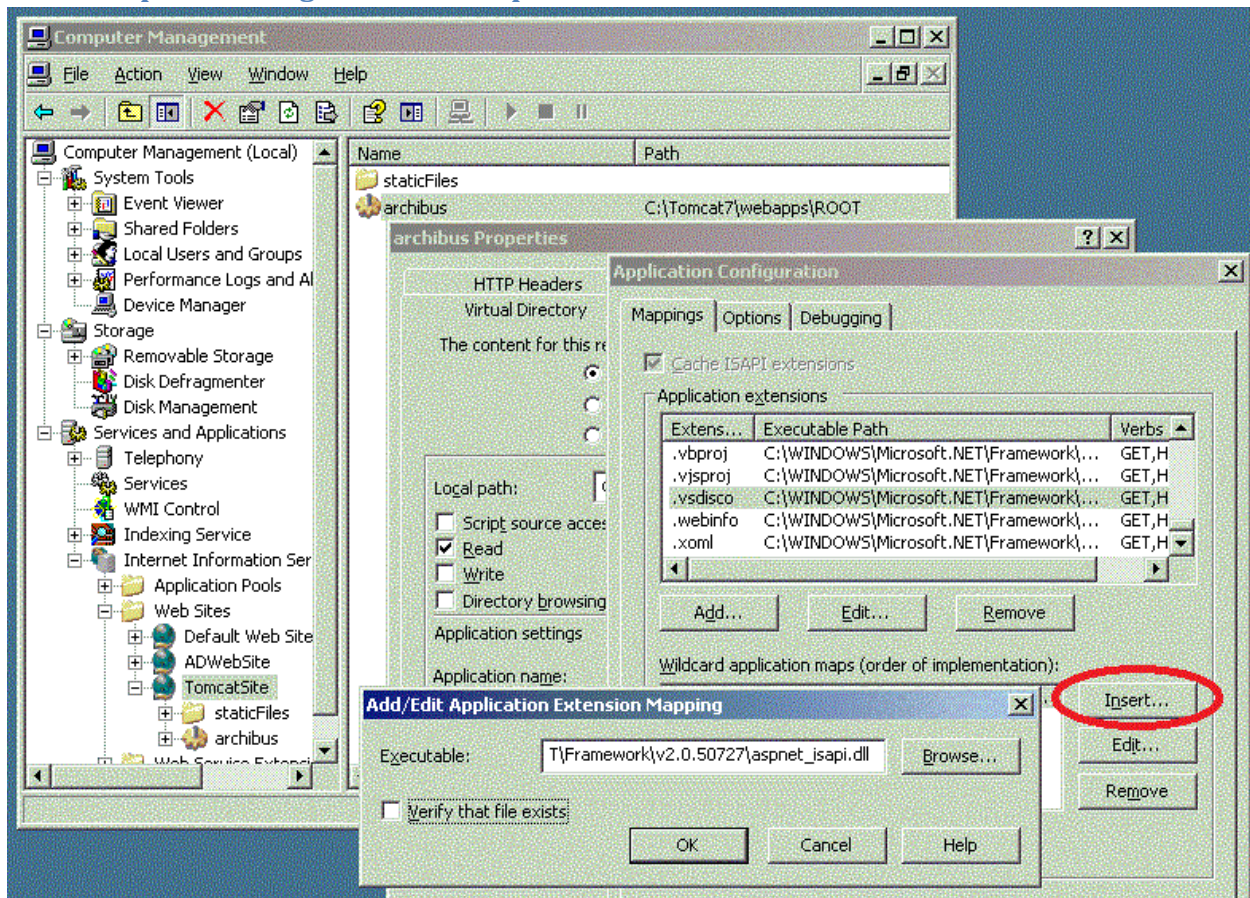


Wildcard mappings:

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.

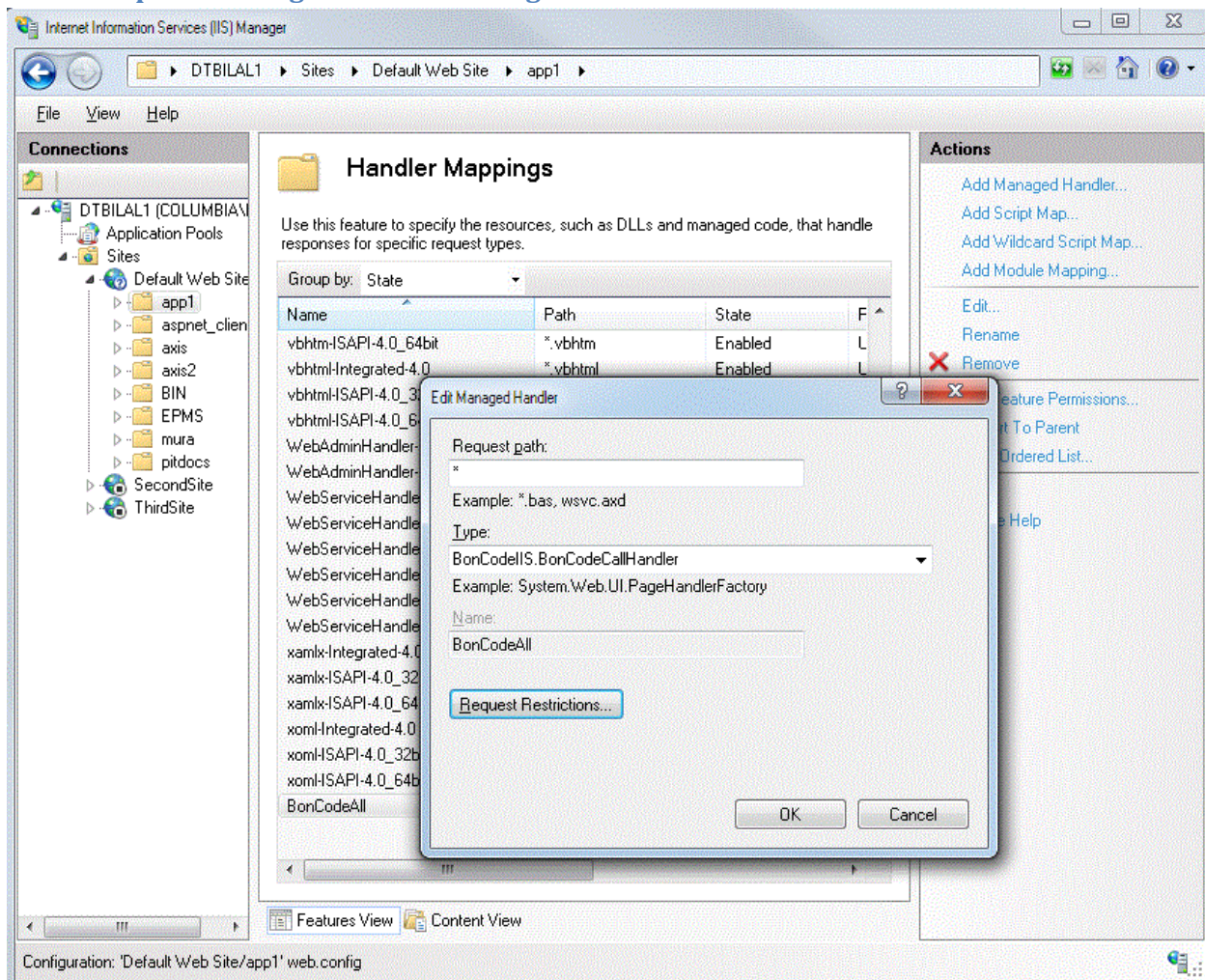
IIS6 Example of adding a wildcard map



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:

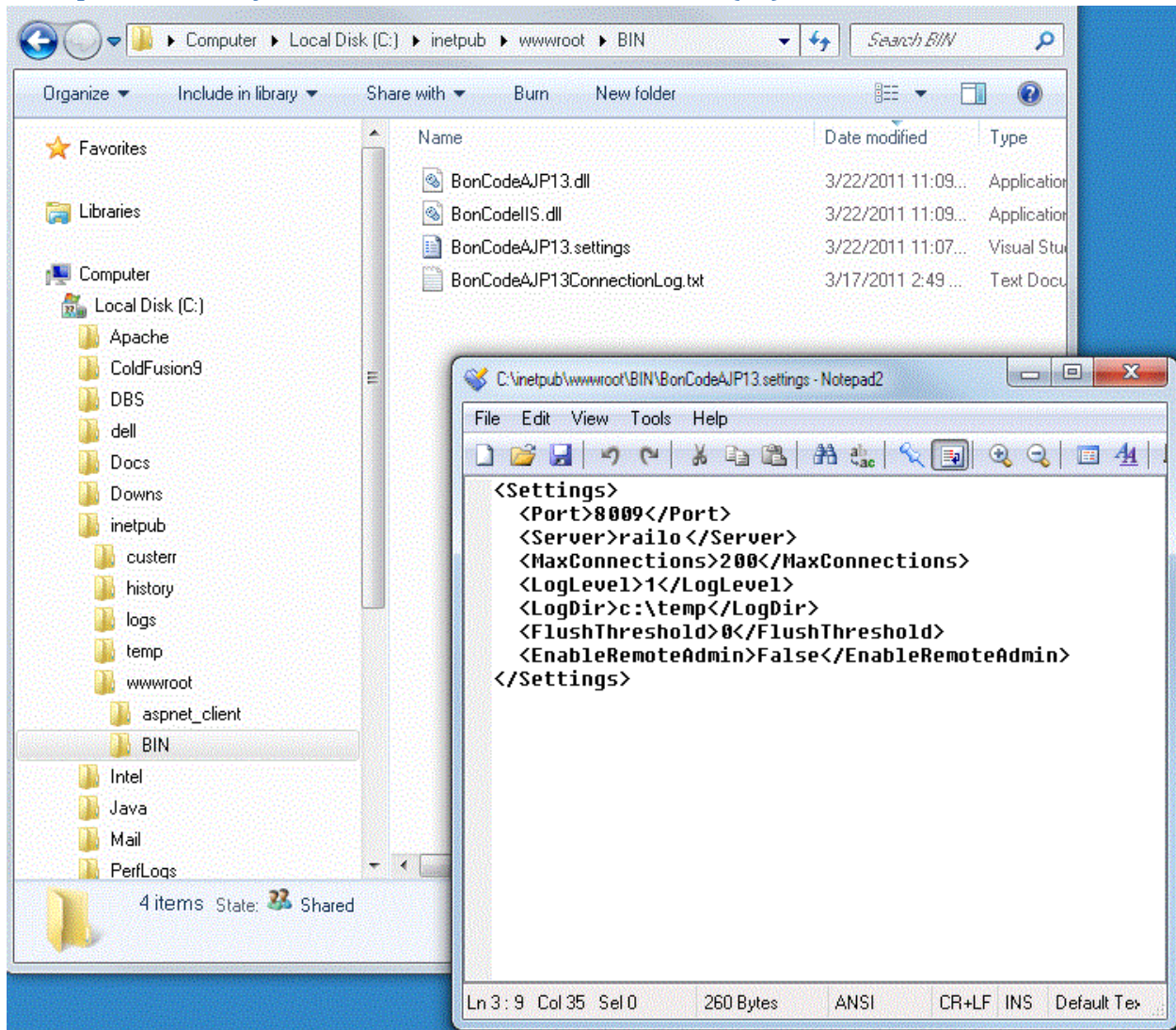
```
<add verb="*" path="*" type="BonCodeIIS.BonCodeCallHandler"/>
```


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 (all):



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

OPTIONAL STEPS

SKIP THIS STEP IF YOU JUST INSTALLED APACHE TOMCAT. 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.

Apache Tomcat configuration (normally this is already configured and nothing needs to change):

- a) Configure worker.properties for tomcat to add the AJP 1.3 protocol. This will tell Apache Tomcat that it can accept and process request via this protocol. Sample worker.properties file is below. See this reference for further options : <http://tomcat.apache.org/connectors-doc/reference/workers.html>
- b) Configure the uriworkermap.properties file for Apache Tomcat. This tells Apache Tomcat which extension it can process through which protocol. A sample file is below.

Sample Tomcat worker.properties file:

THIS IS OPTIONAL. THE DEFAULT APACHE INSTALL IS ALREADY CORRECTLY CONFIGURED.

Apache Tomcat 6 and 7 have jk listener preconfigured by default. If you need to make changes please see Apache Tomcat Documentation.

This file configures the tomcat application server to listen on port 8009 for JServ 1.3 protocol messages:

```
worker.list=ajp13,jkstatus
# Defining a worker named ajp13w and of type ajp13
# Note that the name and the type do not have to match.
worker.ajp13w.type=ajp13
worker.ajp13w.host=localhost
worker.ajp13w.port=8009
# Defining a load balancer
worker.ajp13.type=lb
worker.ajp13.balance_workers=ajp13w
```

Sample Tomcat uriworkermap.properties file:

THIS IS OPTIONAL. THE DEFAULT APACHE INSTALL IS ALREADY CORRECTLY CONFIGURED.

This file contains mappings that direct Apache Tomcat to accept the listed extensions via the ajp13 protocol. The default Apache Tomcat configuration does not need to be changed:

```
/*.cfm=ajp13
/*.cfc=ajp13
/*.cfml=ajp13
```

```
/* .cfres=ajp13
/* .cfchart=ajp13
/* .cfm/*=ajp13
/* .cfml/*=ajp13
/* .cfc/*=ajp13
/* .jsp=ajp13
/* .do=ajp13
/=ajp13
```

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. 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 level. 0=none, 1=Basic (startup/shutdown/error), 2=Headers, 3=Debug and packet Contents. If you set logging to 2 or 3 the log file can get large quickly; please monitor connector carefully. Also ensure that IIS has write permission in the designated Log Directory.
LogDir	Default: installation directory of connector. 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.
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 files not transferring it is recommended to turn this feature off).
EnableHeaderDataSupport	This is for dynamic host configuration on Tomcat side working with the special mode_cfml_valve open source project by Jordan Michaels. 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. 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 tomcat AJP protocol. The connector will send path info in this header instead. If set to blank this will be disabled. Default xajp-path-info

Example Settings file

```

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

```

General Recommendations

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. Thus, 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 recommended that IIS and Apache Tomcat point to the same document directories. Thus for each additional website that is added on IIS a similar entry needs 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.

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 the 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 look most likely 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.

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
```

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 sites will have the connector installed.
acceptLicense	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.
--	--

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 Port on which Tomcat is listening to AJP traffic
configureServerXml	Default:0 Options: 0 or 1 (0=No, 1=Yes) 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. 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.

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) 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>You are not able to retrieve mapped extensions from IIS7.</p> <p>➔</p> <p>You have not installed the ".net extensibility" for IIS7. See above for instructions or use the Server Manager GUI to install this IIS7 feature.</p>
<p>Error Message:</p> <p><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>➔</p> <p>Double check your webroot and ensure that you have the BIN subdirectory.</p> <p>Double check that library files (BonCodeAJP13.dll and BonCodeIIS.dll) are in your BIN subdirectory.</p>
<p>Error Message:</p> <p>Security Exception</p> <p>Description: The application attempted to perform an operation not allowed by the security policy. To grant this application the required permission please contact your</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:</p>

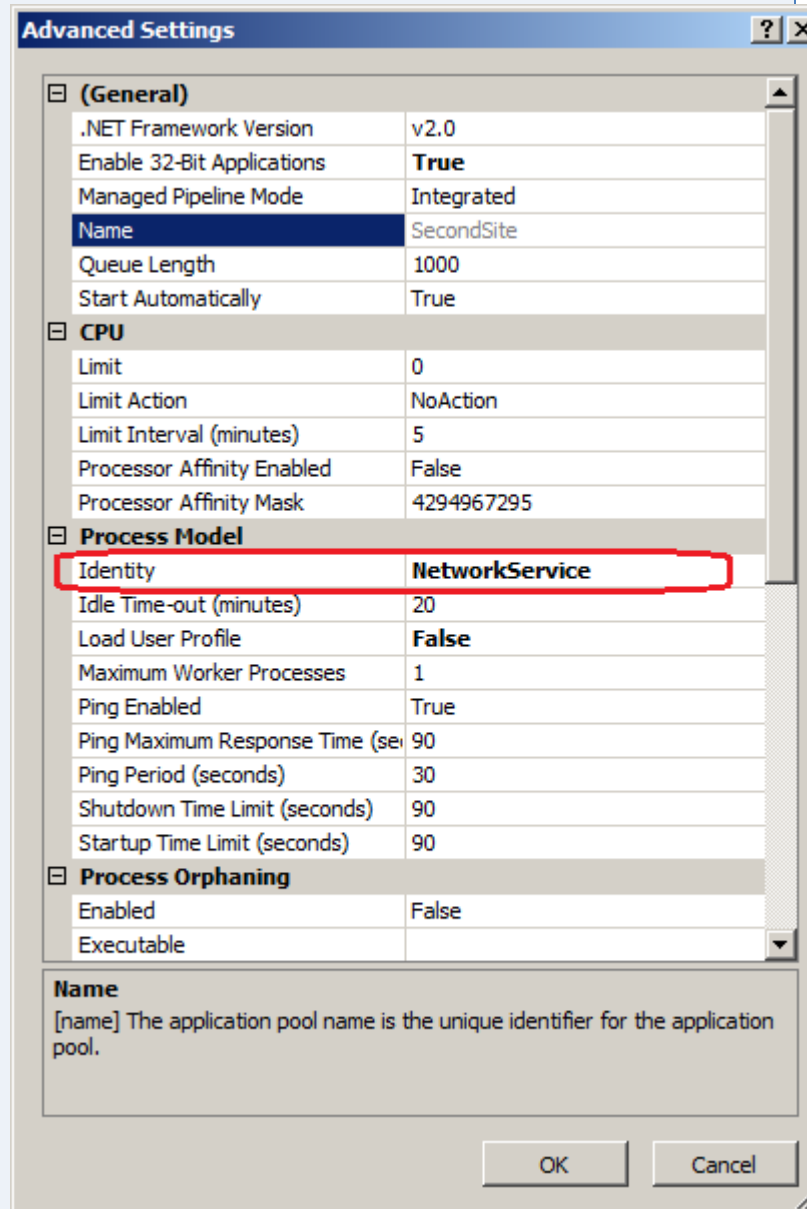
system administrator or change the application's trust level in the configuration file.

Exception Details:
System.Security.SecurityException:
Request for the permission of type
'System.Web.AspNetHostingPermissi
on, System, Version=2.0.0.0,
Culture=neutral,
PublicKeyToken=b77a5c561934e089'
failed.

<http://blogs.msdn.com/b/friis/archive/2010/06/09/system-security-securityexception-request-for-the-permission-of-type-system-web-aspnethostingpermission-failed.aspx>

Simply browse to the files in your webfolder, right-click, then, click the “Unblock” button on the General-Tab.

Or, you can change the Application Pool Identity back to “NetworkService” which was used in IIS7.



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}");
```

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