Using Get / Set IIS Property with IIS7

Contents

Intro

A recent release of InstallAware has allowed the use of Get / Set IIS Property commands with IIS 7 (and later). Not only that it is now possible to access direct properties and attributes in the IIS7 configuration system through a special syntax, but InstallAware also automatically maps IIS6 metabase properties to IIS7 configuration paths, saving you time and trouble with respect to backward compatibility to IIS6.

Configuration Access Syntax

In order to allow access to IIS7 configuration properties, InstallAware has expanded the meaning of the Property Name field inside the Get / Set IIS Property commands. Now, this field can either contain IIS6 metabase property names (which will automatically be mapped into IIS7 configuration paths - see Automatically Mapping IIS6 Properties to IIS7 Configuration Paths below for a list of automatically mapped properties), or IIS7 configuration access expressions.

An IIS7 configuration access expression is comprised of two parts. Its first part is the IIS section name of the configuration path (for example, system.webServer/asp). Its second part is the actual path inside the configuration section that InstallAware is to retrieve. The latter can contain the following commands:

1) "[]" – retrieve the entire collection
2) "()" – retrieve all child elements
3) "[elementname]" – access the specific element inside the collection named elementname
4) "[elementname:property=value]" – access a specific element inside the collection named elementname that **contains** a property named property that has the value value. This might seem confusing at first, but it will come in handy as you read through this manual.
5) "(childname)" – access the specific child element named childname
6) "(childname:property=value)" – access a specific child element named childname that **contains** a property named property that has the value value
7) "->" - access a specific attribute inside the element

Here is an example of an IIS7 configuration access expression:

system.webServer/security/access->sslFlags

This will point InstallAware to the section system.webServer/security/access. InstallAware will then access that configuration path, and retrieve the attribute named "sslFlags". Please note that the above commands are "stackable". Here is another example:

some.configuration/section(informationSegments)[segment:id=1]()

In this case, the section name would be “some.configuration/section”. InstallAware will access that specific section. It will then move on to the “informationSegments” child element. From there, InstallAware will go to the element inside the “informationSegments” collection that is named “segment” and that contains a property named “id” that has a value “1”. Finally, InstallAware will enumerate and return a representation of all the child elements of the “segment” element.

APPHOST / web.config

IIS7 allows for a feature called Configuration Delegation. This essentially means that the configuration system is organized in a hierarchy, and that it is possible to define configuration elements at different levels of the hierarchy. The level of the hierarchy in which a property is set or retrieved is determined by the Web Site and Virtual Folder field of the Get / Set IIS Property commands.

When the Get / Set IIS Property commands are called without a site index nor a virtual folder, the configuration is set / retrieved on a global level (MACHINE/WEBROOT/APPHOST/).

When Get / Set IIS Property commands are called with a site index but without a virtual folder, configuration is set / retrieved on the site’s level (MACHINE/WEBROOT/APPHOST/SiteName/).

When Get / Set IIS Property commands are called with a site index and a virtual folder, configuration is set / retrieved on the virtual folder’s level (MACHINE/WEBROOT/APPHOST/SiteName/Folder).

Using the IIS Schema to determine the Configuration Access Expression

As you know, IIS7 pre-defines the structure of its configuration system in the IIS_schema.xml file, located in the folder %windir%\system32\inetsrv\config\schema. Using this file, we are able to determine how to construct the configuration access expression.

For example, let’s say we want to retrieve the Default Web Site’s TCP binding. Let’s examine the IIS_schema file and find the section that describes websites:

```
<sectionSchema name="system.applicationHost/sites">
  <collection addElement="site" defaultElement="siteDefaults" >
    <attribute name="name" type="string" required="true" isUniqueKey="true" validationType="siteName" />
    <attribute name="id" type="uint" required="true" isUniqueKey="true" />
    <attribute name="serverAutoStart" type="bool" defaultValue="true" />
    <element name="bindings">
      <collection addElement="binding" clearElement="clear">
        <attribute name="protocol" type="string" required="true" isCombinedKey="true" validationType="nonEmptyString" />
        <attribute name="bindingInformation" type="string" required="true" isCombinedKey="true" validationType="nonEmptySt
      </collection>
    </element>
```

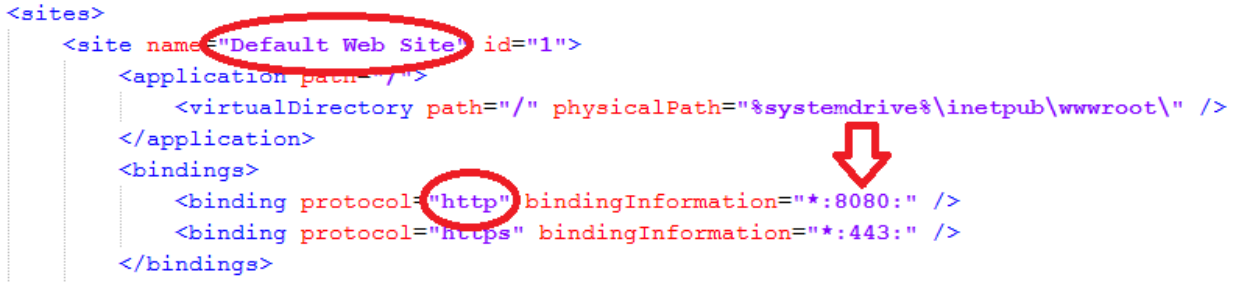
As you can see, the sites are described inside a section named “system.applicationHost/sites”. That means that the first part of our configuration access expression will be “system.applicationHost/sites”. Now let’s look at the way that IIS saves the configuration inside the “sites” section. We can see that the “sites” section contains a collection (line 2) of “site” elements. “site” elements have 3 attributes – “name”, “id”, and “serverAutoStart” (lines 3, 4, 5). “site” elements also contain child elements (line 6)

called “bindings”. Let’s translate what we have so far to an expression string. As we said before, the first part will be “system.applicationHost/sites”. This will put us at the “site” collection. Then, we need to access our specific site’s (“Default Web Site”) “site” element. We will do so by using a collection directive with a constraint. The constraint will be an element named “site” that has a property named ”name” that has the value “Default Web Site”. Using the configuration access expression syntax, we can create this constraint by adding the following – “[site:name=Default Web Site]”. Now, our configuration access exception has the value “system.applicationHost/sites[site:name=Default Web Site]”. This puts as at the “site” element that has the property “name=Default Web Site”. Now, let’s access that site’s “bindings” child element (noted by the “element” declaration in the IIS_schema), which is a collection of “binding” elements (noted by the “collection” declaration in the IIS_schema). As you recall, accessing collections is done by using the (collection_name) directive. So let’s add (bindings) to our expression. As we can see, the “binding” elements inside the “bindings” collection have a “protocol” attributes. To get to the TCP binding, we need to get to the “binding” element that has a “protocol” attribute with the value “http”. To do so, we will use a constraint again. As you figured, the syntax to get to that “binding” will be “[binding:protocol=https]”. Now, our expression equals “system.applicationHost/sites[site:name=Default Web Site](bindings)[binding:protocol=https]”. This puts us at the right “binding” element, but that’s not what we want – we want the actual TCP binding. To do so, let’s look at the “binding” element as it is described in the IIS_schema. As you can see, the “binding” element has an attribute named “bindingInformation”. This attribute holds the actual TCP binding of Default Web Site. To get to its value, we will use the attribute directive (->). Our final configuration access expression is:

system.applicationHost/sites[site:name=Default Web Site](bindings)[binding:protocol=http]->bindingInformation

Please note that sites are described on the APPHOST global configuration path (MACHINE/WEBROOT/APPHOST/). This means, if you recall from the “APPHOST / web.config” section in this manual, that we need to not specify a specific web site or virtual folder in the Get IIS Property command dialog.

If you look at the %windir%\system32\inetsrv\config\applicationHost.config file, which holds most of your APPHOST global configuration, you could see the collections, child elements and attributes for yourself:



Automatically Mapping IIS6 Properties to IIS7 Configuration Paths

Although IIS7 deprecates the use of metabase properties that are supported on IIS6, InstallAware automatically maps your IIS6 metabase properties to IIS7 configuration paths. For example, the AccessSSLFlags metabase property is automatically mapped (in accordance with Microsoft's documentation) to system.webServer/security/access->sslFlags. Here is a list of the IIS6 metabase properties that are mapped automatically by InstallAware:

IIS6 Metabase Property	IIS7 Configuration Path
AccessFlags	system.webServer/handlers->accessPolicy
AccessSSLFlags	system.webServer/security/access->sslFlags
AllowKeepAlive	system.webServer/httpProtocol->allowKeepAlive
AnonymousUserName	system.webServer/security/authentication/anonymousAuthentication->username
AnonymousUserPass	system.webServer/security/authentication/anonymousAuthentication->password
AppAllowClientDebug	system.webServer/asp->appAllowClientDebug
AppAllowDebugging	system.webServer/asp->appAllowDebugging
AspAllowSessionState	system.webServer/asp(session)->allowSessionState
AspAppServiceFlags	system.webServer/asp(comPlus)->appServiceFlags
AspAppServiceFlags.AspEnableTracker	system.webServer/asp(comPlus)->appServiceFlags
AspAppServiceFlags.AspEnableSxs	system.webServer/asp(comPlus)->appServiceFlags
AspAppServiceFlags.AspUsePartition	system.webServer/asp(comPlus)->appServiceFlags
AspBufferingLimit	system.webServer/asp(limits)->bufferingLimit
AspBufferingOn	system.webServer/asp->calcLineNumber
AspCodePage	system.webServer/asp->codePage

AspDiskTemplateCacheDirectory	system.webServer/asp(cache)->diskTemplateCacheDirectory
AspEnableApplicationRestart	system.webServer/asp->enableApplicationRestart
AspEnableAspHtmlFallback	system.webServer/asp->enableAspHtmlFallback
AspEnableChunkedEncoding	system.webServer/asp->enableChunkedEncoding
AspEnableParentPaths	system.webServer/asp->enableParentPaths
AspEnableTypelibCache	system.webServer/asp(cache)->enableTypelibCache
AspErrorsToNTLog	system.webServer/asp->errorsToNTLog
AspExceptionCatchEnable	system.webServer/asp->exceptionCatchEnable
AspExecuteInMTA	system.webServer/asp(comPlus)->executeInMta
AspKeepSessionIdSecure	system.webServer/asp(session)->keepSessionIdSecure
AspLCID	system.webServer/asp->lcid
AspLogErrorRequests	system.webServer/asp->logErrorRequests
AspMaxDiskTemplateCacheFiles	system.webServer/asp(cache)->maxDiskTemplateCacheFiles
AspMaxRequestEntityAllowed	system.webServer/asp(limits)->maxRequestEntityAllowed
AspPartitionID	system.webServer/asp(comPlus)->partitionId
AspProcessorThreadMax	system.webServer/asp(limits)->processorThreadMax
AspQueueConnectionTestTime	system.webServer/asp(limits)->queueConnectionTestTime
AspQueueTimeout	system.webServer/asp(limits)->queueTimeout
AspRequestQueueMax	system.webServer/asp(limits)->AspRequestQueueMax
AspRunEndOnEndAnonymously	system.webServer/asp->runEndOnEndAnonymously
AspScriptEngineCacheMax	system.webServer/asp(cache)->scriptEngineCacheMax
AspScriptErrorMessage	system.webServer/asp->scriptErrorMessage
AspScriptErrorSentToBrowser	system.webServer/asp->scriptErrorSentToBrowser
AspScriptFileCacheSize	system.webServer/asp(cache)->scriptFileCacheSize

AspScriptLanguage	system.webServer/asp->scriptLanguage
AspScriptTimeout	system.webServer/asp(limits)->scriptTimeout
AspSessionMax	system.webServer/asp(session)->max
AspSessionTimeout	system.webServer/asp(session)->timeout
AspSxsName	system.webServer/asp(comPlus)->sxsName
AspTrackThreadingModel	system.webServer/asp(comPlus)->trackThreadingModel
AuthFlags.AuthAnonymous	system.webServer/security/authentication/anonymousAuthentication->enabled
AuthFlags.AuthBasic	system.webServer/security/authentication/basicAuthentication->enabled
AuthFlags.AuthMD5	system.webServer/security/authentication/digestAuthentication->enabled
AuthFlags.AuthNTLM	system.webServer/security/authentication/windowsAuthentication->enabled
AuthPersistence	system.webServer/security/authentication/windowsAuthentication->authPersistSingleRequest
CacheControlCustom	system.webServer/staticContent(clientCache)->cacheControlCustom
CacheControlMaxAge	system.webServer/staticContent(clientCache)->cacheControlMaxAge
CacheControlNoCache	system.webServer/staticContent(clientCache)->cacheControlMode
CentralBinaryLoggingEnabled	system.applicationHost/log(centralBinaryLogFile)->enabled
CentralW3CLoggingEnabled	system.applicationHost/log(centralW3CLogFile)->enabled
CGIRestrictionList	system.webServer/security/isapiCgiRestriction
CGITimeout	system.webServer/cgi->timeout
ConnectionTimeout	system.applicationHost/sites[site:id=$SITEID$](limits)->connectionTimeout
CreateCGIWithNewConsole	system.webServer/cgi->createCGIWithNewConsole
CreateProcessAsUser	system.webServer/cgi->createProcessAsUser

CGITimeout	system.webServer/cgi->timeout
DefaultDoc	system.webServer/defaultDocument()
DefaultDocFooter	system.webServer/staticContent->defaultDocFooter
DefaultLogonDomain	system.webServer/security/authentication/basicAuthentication->defaultLogonDomain
DemandStartThreshold	system.applicationHost/webLimits->demandStartThreshold
DirBrowseFlags.DirBrowsexxx	system.webServer/directoryBrowse->showFlags
DirBrowseFlags.EnableDefaultDoc	system.webServer/defaultDocument->enabled
DontLog	system.webServer/directoryBrowse->dontLog
EnableDocFooter	system.webServer/staticContent->enableDocFooter
EnableReverseDns	system.webServer/security/ipSecurity->enableReverseDns
HcCacheControlHeader	system.webServer/httpCompression->cacheControlHeader
HcCompressionDirectory	system.webServer/httpCompression->directory
HcDoDiskSpaceLimiting	system.webServer/httpCompression->doDiskSpaceLimiting
HcExpiresHeader	system.webServer/httpCompression->expiresHeader
HcMaxDiskSpaceUsage	system.webServer/httpCompression->maxDiskSpaceUsage
HcMinFileSizeForComp	system.webServer/httpCompression->minFileSizeForComp
HcNoCompressionForHttp10	system.webServer/httpCompression->noCompressionForHttp10
HcNoCompressionForProxies	system.webServer/httpCompression->noCompressionForProxies
HcNoCompressionForRange	system.webServer/httpCompression->noCompressionForRange
HcSendCacheHeaders	system.webServer/httpCompression->sendCacheHeaders
HcSendCacheHeaders	system.applicationHost/webLimits->headerWaitTimeout
HttpCustomHeaders	system.webServer/httpProtocol(customHeaders)[]
HttpErrors	system.webServer/httpErrors[]

HttpRedirect	system.webServer/httpProtocol(redirectHeaders)[]
IPSecurity	system.webServer/security/ipSecurity->allowUnlisted
LogExtFileFlags	system.applicationHost/log(centralW3CLogFile)->logExtFileFlags
LogFileDirectory	system.applicationHost/log(centralBinaryLogFile)->directory
LogFileLocalTimeRollover	system.applicationHost/log(centralBinaryLogFile)->localTimeRollover
LogFilePeriod	system.applicationHost/log(centralBinaryLogFile)->period
LogFileTruncateSize	system.applicationHost/log(centralBinaryLogFile)->truncateSize
LoginUTF8	system.applicationHost/log->logInUTF8
LogPluginClsid	system.applicationHost/sites[site:id=$SITEID$](logFile)->customLogPluginClsid
MaxBandwidth	system.applicationHost/sites[site:id=$SITEID$](limits)->maxBandwidth
MaxConnections	system.applicationHost/sites[site:id=$SITEID$](limits)->maxConnections
MaxGlobalBandwidth	system.applicationHost/webLimits->maxGlobalBandwidth
MimeMap	system.webServer/staticContent->mimeMap
MinFileBytesPerSec	system.applicationHost/webLimits->minBytesPerSecond
Path	system.applicationHost/sites[site:id=$SITEID$][application][virtualDirectory]->physicalPath
Realm	system.webServer/security/authentication/basicAuthentication->realm
RedirectHeaders	system.webServer/httpProtocol(redirectHeaders)[]
ScriptMaps.Extension	system.webServer/handlers->path
ScriptMaps.Flags	system.webServer/handlers->requireAccess
ScriptMaps.Check_Path_Info	system.webServer/handlers->resourceType
ScriptMaps.ScriptProcessor	system.webServer/handlers->scriptProcessor

ScriptMaps.Verbs	system.webServer/handlers->verb
ScriptMaps.Flags	system.webServer/handlers->requireAccess
SecureBindings	system.applicationHost/sites[site:id=$SITEID$](bindings)[binding:protocol=https]->bindingInformation
ServerAutostart	system.applicationHost/sites[site:id=$SITEID$]->serverAutoStart
ServerBindings	system.applicationHost/sites[site:id=$SITEID$](bindings)[]
SSIExecDisable	system.webServer/serverSideInclude->ssiExecDisable
UploadReadAheadSize	system.webServer/serverRuntime->uploadReadAheadSize