

Running SAWS Behind IIS

Overview

SAWS is Dyalog's Stand Alone Web Services framework for consuming and providing SOAP-based web services. "Stand Alone" means that SAWS does not require any other web server components to function. In some instances, it may be desirable to run SAWS behind a commercial web server.

- In a corporate environment, industry standard web servers like IIS and Apache may be required by the IT department.
- Other web servers may provide functionality like logging, redirection, etc., which are not present in the web server component within SAWS.

With this in mind, this document describes how to configure SAWS and IIS in order to run SAWS behind IIS.

Setting up a Single SAWS Instance

Step 1 – Get the Latest SAWS Software

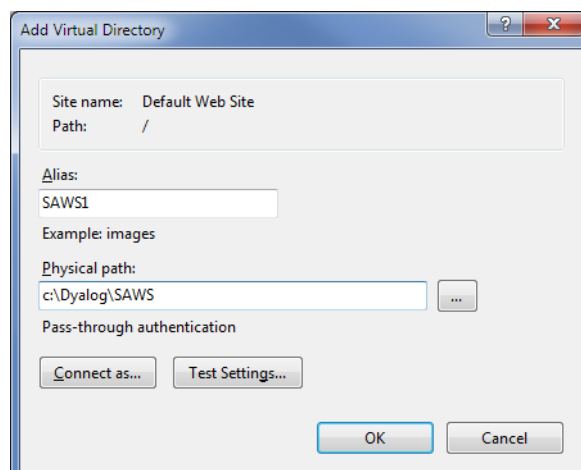
SAWS is located on GitHub in the repository linked by: <https://github.com/Dyalog/SAWS.git>

You can either clone the entire repository, or, at a minimum copy the contents from the SAWS/Distribution/ folder at: <https://github.com/Dyalog/SAWS/tree/master/Distribution>

Step 2 – Configure IIS

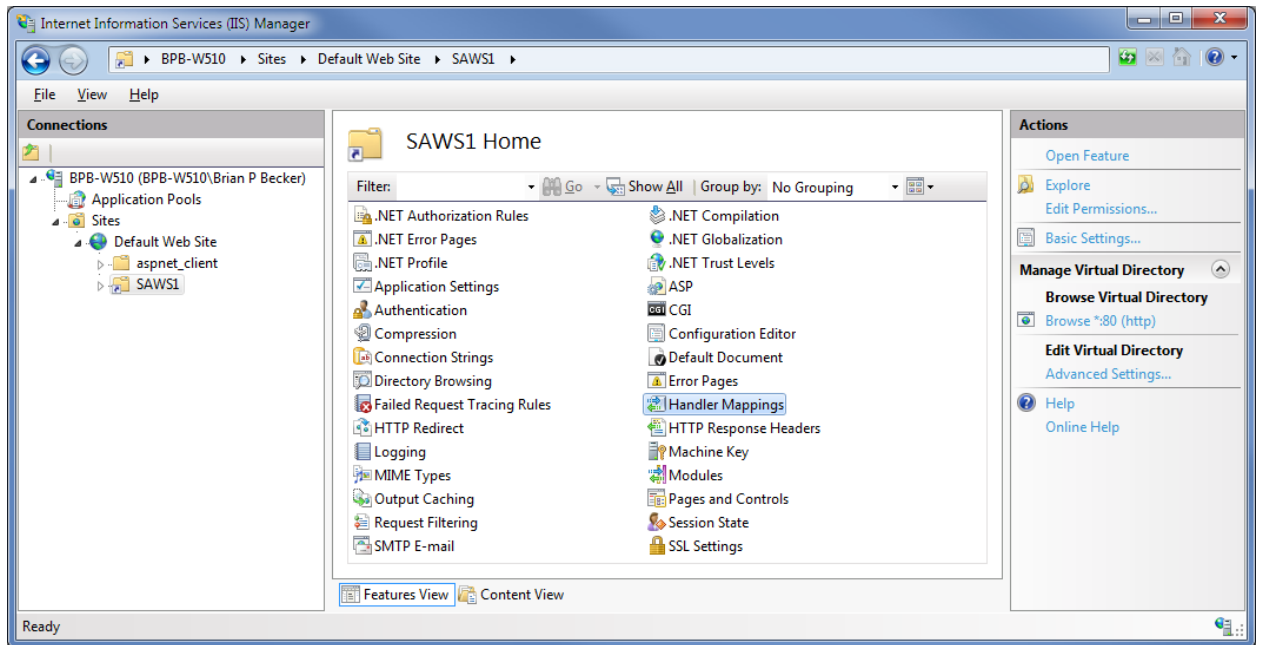
There are some options on how to do this step. The decision largely boils down to whether you want your IIS instance to host only SAWS or other web content as well. In my case, I chose the latter.

- 1) Open the Internet Information Services (IIS) Manager – there should be an option to do this from the Administrative Tools menu.
- 2) Right-click on Default Web Site entry in the Connections Panel on the left of the screen and select "Add Virtual Directory..."
 - a. Choose an alias for serving SAWS content – in the example, we use SAWS1, meaning that URLs with the format {web_address}/SAWS1/... will be directed at SAWS. Note: this name is completely arbitrary – you can use anything you like.
 - b. Enter the physical path where SAWS is installed

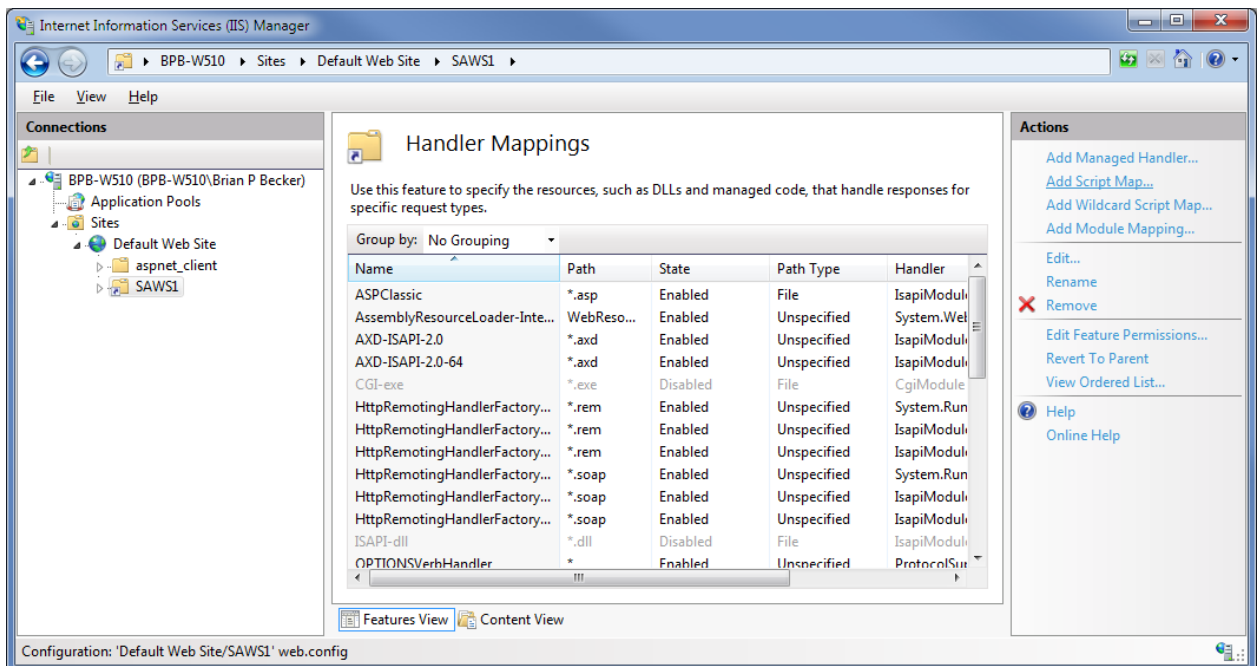


Running SAWS Behind IIS

- 3) With your newly added Virtual Directory selected, open the Handler Mappings configuration.



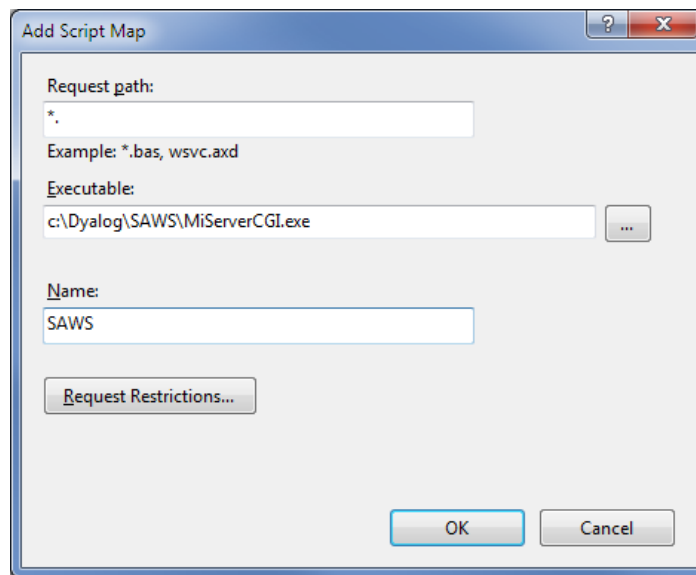
- 4) Under the Actions menu on the right of the screen, select "Add Script Map..."



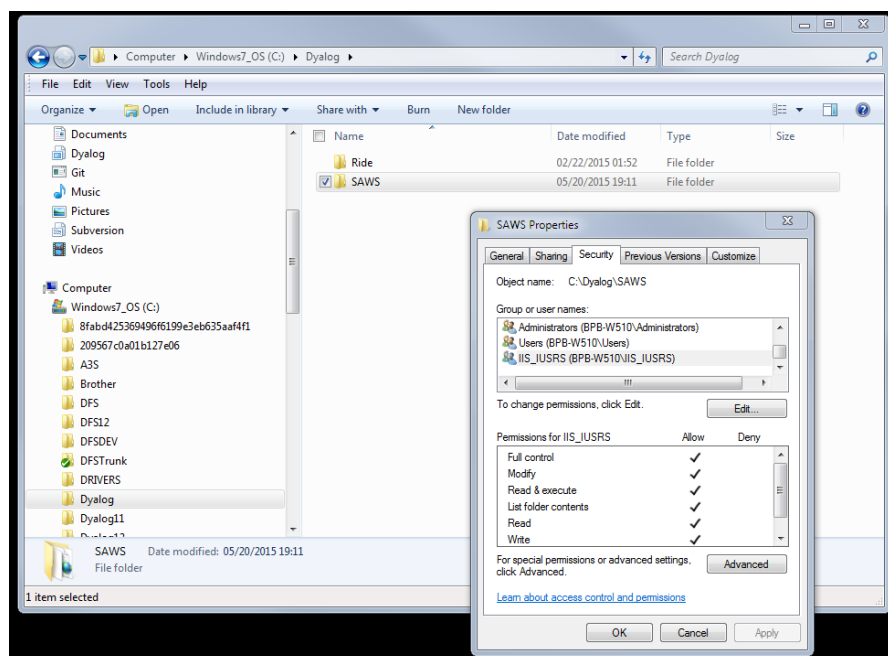
Running SAWS Behind IIS

5) Add the Script Map

- Use a request path of "*" – meaning all requests to this virtual directory will be handled by this handler.
- Enter the location of the MiServerCGI.exe that you downloaded from GitHub. By default, MiServerCGI expects SAWS to be running on port 8080 and on this machine 127.0.0.1 (localhost). Command line parameters are available to change these:
-p port-number
-a IP-address
- Give the script map a name – again this name is arbitrary.



- Depending on how your security is set up, you may need to give the IIS_IUSRS group permission on the folder where SAWS is installed.



Running SAWS Behind IIS

Step 3 – Test SAWS Behind IIS

- 1) Start SAWS on port 8080 (or the port you specified in the command line in the Script Map).

```
SAWS.Run 8080 1 ' ' ' ' '/SAWS1'
```

There is a new fifth parameter in the right argument to SAWS.Run – this is the Virtual Directory alias you specified in Step 2.2

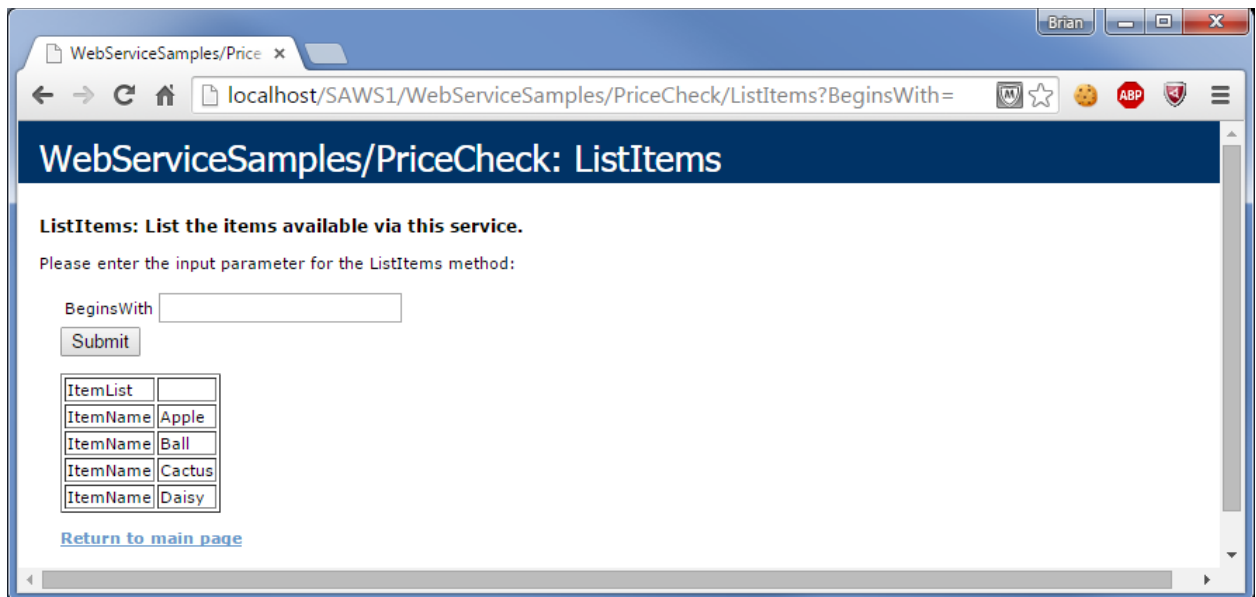
- 2) Test the installation. You can do this by either opening a browser or using SAWS.Call

```
SAWS.Run 8080 1 ' ' ' ' '/SAWS1'
```

Web server 'HTTPSRV' started on port 8080

Handling requests using ##.HandleRequest

```
'localhost' SAWS.Call '/SAWS1/WebServiceSamples/PriceCheck' 'ListItems' ('BeginsWith' '')
0 ListItems 1 ItemList      xmlns http://localhost//SAWS1/WebServiceSamples/PriceCheck/
            2 ItemName     Apple
            2 ItemName     Ball
            2 ItemName     Cactus
            2 ItemName     Daisy
```



Setting up Multiple SAWS Instances

This is being researched.