Installing and Configuring BellHawk on a Windows Server

Introduction

This document is intended for the IT (Information Technology) staff of clients and partners to assist them in configuring and installing a BellHawk system. It assumes that readers are already familiar with setting up and supporting Microsoft Windows Servers and Workstations as well as related network components.

This manual covers the installation of a standard BellHawk operations tracking and management system as well as options that are run through its web-browser interface without requiring the installation of additional software. The installation of these options, such as the TAG barcode labeling option, weighing scale interfaces, DEX and MilramX interfaces are covered in separate configuration and installation guides, as is using SSRS reporting with BellHawk.

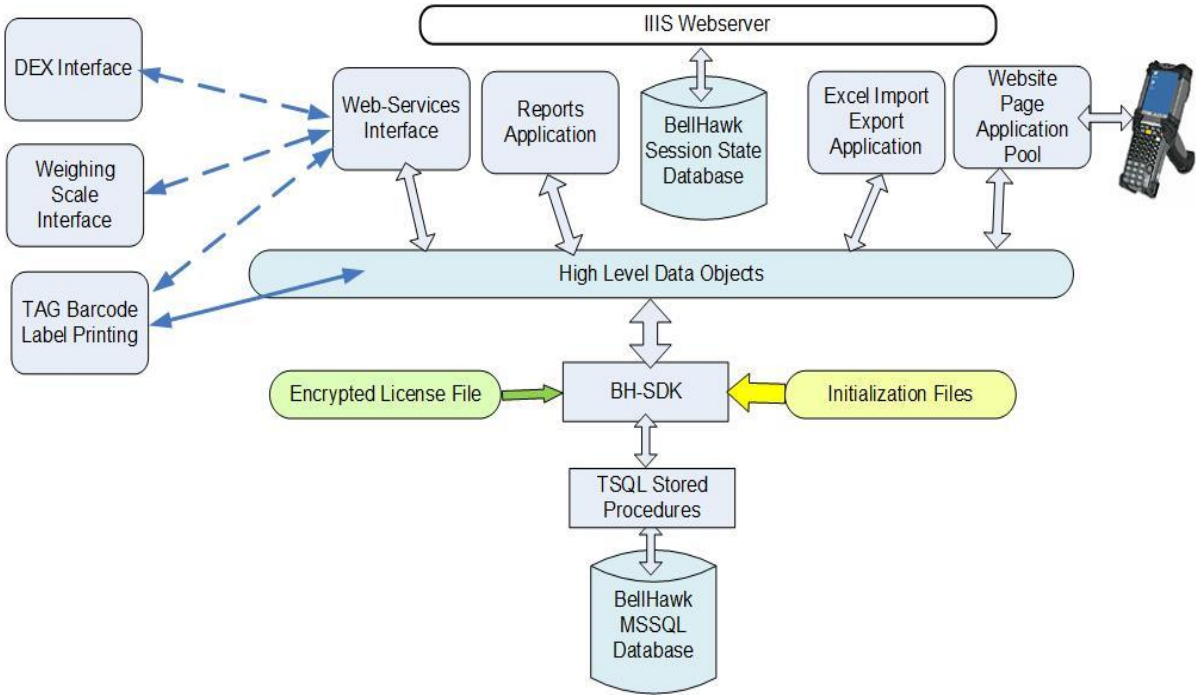
This guide is for installing BellHawk on Windows Server 2019 using IIS version 10 and SQL Server 2016. The process may differ in detail for other versions.

BellHawk Systems Architecture

BellHawk runs on a Windows Server computer. Its primary components are:

1. A BellHawk website through which users interact using web-browser based devices which are equipped with barcode scanners for data collection purposes.
2. A SQL Server database which holds all of the setup, tracking, and history data for BellHawk.

3. A set of interfaces through which external programs can interact with BellHawk.
4. Software that runs in Windows Workstation based computers or IIOT (industrial Internet of Things) appliances to provide local interfaces to devices such as barcode label printers, weighing scales, and RFID scanning stations.



BellHawk runs as an ASP.Net website under the control of the Microsoft IIS Webserver software on a Windows Server. This website consists of ASP.Net code in conjunction with VB.Net code libraries. It uses two SQL Server databases.

There is a main website plus two application websites, one for displaying reports on the user's screen, so that they can be printed, and one for doing more complex Excel imports and exports, such as for Work Orders. Please note that simple Excel imports to setup data, such as locations, are handled by the main BellHawk website.

BellHawk has a SOAP/XML web-services interface which is used by external interface code, such as that for Barcode Label printing and weighing scale data capture in a local plant that may be thousands of miles from the data center in which BellHawk is running.

For external interfaces, developers may directly communicate with the open architecture BellHawk database through its stored procedures, use its BHSDK.Net interface, use the web-services interface, or use the DEX interface, which remotely "mirrors" tables in the BellHawk database.

BellHawk uses HLDO's (High Level Data Objects) as a way to communicate with external systems, as well as for its own internal communications, through its BH-SDK interface. Please see data sheet on HLDOs, available from www.BellHawk.com, for more details.

BH-SDK is a DLL that contains the Tau-Adaptor expert system, which manages data exchange within BellHawk. BHSDK is setup using .ini initialization files. It also uses HLDO metadata files, and controls what features of BellHawk can be used based on the contents of an encrypted license-file.

Users access the website by entering the URL of the BellHawk website into a web browser running on a PC, wireless mobile computer, tablet or other similar device. Thereafter all interactions from that web-browser session on the device are relayed by IIS to the BellHawk website code.

In BellHawk, staff users have their own logins. Operators use shared devices, such as mobile computers, which have their own logins. In this case each individual operator is identified by scanning a barcode on their employee badge whenever they enter data.

BellHawk uses two SQL Server databases. One is the standard BellHawk database where all the materials tracking and traceability data is retained and the other is a special session state database that is used to retain the session state for each user between browser interactions with the website.

As data is entered into the BellHawk software through the form on the web-browser screen, the web-browser is sending each data entry field entered back to the website where it is verified for correctness against the contents of the main BellHawk database. This enables users to be warned as soon as they make a mistake. It also enables the data entry form to be modified dynamically as data is entered, so as to simplify the data entry task for the user.

When devices or users log-in, their user names and passwords are verified against device or user names and encrypted passwords in the main BellHawk database. A session state is then created for each device or user. This login verification also applies to any external system or device (including the TAG barcode printing module and weighing scales) that communicate with BellHawk through its Web Services interface.

As data is entered, it is remembered in the session state database, so when the form is finally submitted, it can be checked again for correctness before being stored in the main BellHawk database. BellHawk also uses a tree structure for its data entry where the user descends and ascends through a hierarchy of screens. The current position in the hierarchy is also remembered in the session state database.

The numbers of device and user logins, as well as the optional modules, which are licensed, are read from an encrypted license file whenever a user or device logs in. This license file also controls the termination date of subscriptions and free trials. This mechanism enables BellHawk Systems to update the available system configuration by issuing an updated license file.

The license file is specific to a named server as well as a named company and plant plus the state and country of use.

BellHawk will run on a physical or virtual Windows Server. But please bear in mind that each mobile device and PC may well interact with the website every few seconds. As a result, it is important that adequate processing cycles and memory be allocated to the BellHawk website otherwise response to user inputs may suffer.

It is desirable for both BellHawk databases to be co-located with the BellHawk website code so as to minimize latency of access from the website pages to the database. Please bear in mind that (1) each device will typically be running its own web-page(s) in parallel on the server and (2) that each web-page will have one or more parallel SQL Server threads active at any one time.

We find that a low-end physical 64 Bit Windows Server with a quad core processor and 8 GBytes of memory can support a dozen or more devices and an equal number of staff users without degradation in performance. Virtual servers also work well provided that the equivalent resources are available to the virtual server and that there is not a long delay between each virtual server getting a slice of processor time due to compute bound jobs being run on the other virtual machines that share the physical processor and memory.

As the data packets exchanged between the devices and the web-server are limited to a few Kbytes for each data exchange, the average network bandwidth required is relatively low unless there are a large number of devices and users using the system. We find that a 10 Mbps network connection is more than adequate for the LAN, wireless LAN, WAN and Internet connectivity between the devices and the server. While low bandwidth, less than 1 Mbps connections will work, user interaction times may be less than desirable.

We recommend stress testing the configuration with the maximum expected number of devices and users before operational deployment.

The TAG barcode label printing module as well as the weighing scale interface module make use of the web-services (SOAP/XML) interface provided by BellHawk. This can also be used by external systems to exchange data with BellHawk. This interface is written in VB.Net using the Windows Communications Foundation (WCF) libraries provided by Microsoft.

External software applications, communicating with BellHawk through its web-services interface, send SOAP/XML packets to the IIS webserver, which then relays them to the WCF based web-services interface code that is part of the BellHawk software. This interface enables external systems to interrogate data objects stored in the BellHawk database and to send new data objects to be stored in the BellHawk database. This interaction can take place over the client's Intranet or over the Internet.

If the public Internet is used for device, user, or web-services interaction with BellHawk, and the BellHawk data needs to be kept private, it is recommended that encrypted secure socket layer (https) communications be used. Please note that, without these precautions, user names and passwords are sent "in-the-clear" over the Internet by web-browsers and could be intercepted.

System Setup

Setting up IIS

BellHawk runs as a website under the control of the IIS webserver, which is provided as part of a Windows Server operating system.

The first step in setting up a BellHawk system is to set up the IIS webserver, if this has not already been done on the Windows Server you are using. By default IIS is not turned on as a security measure by Microsoft.

In a Windows Server operating system, use the Server Manager/Server Roles/Web Server to install IIS and required features.

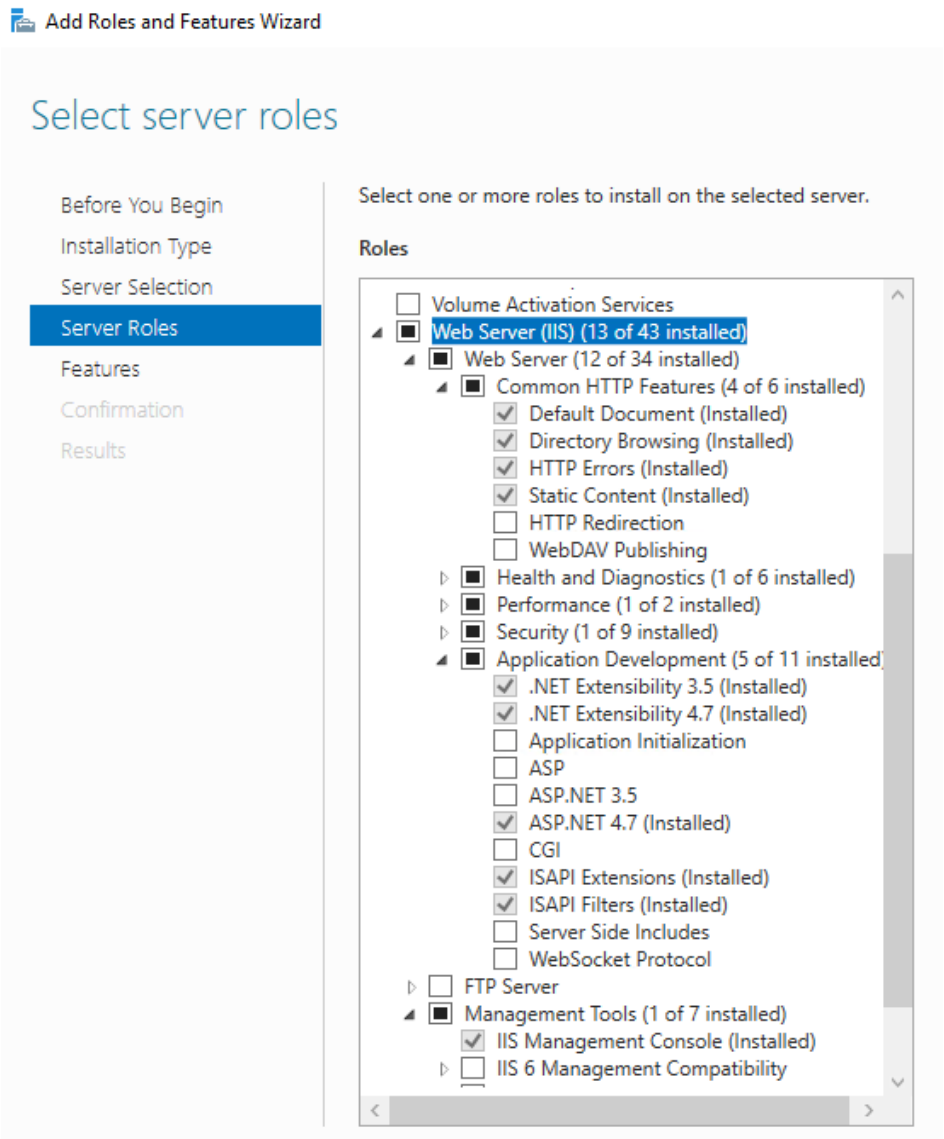
Make sure that you have the following set:

1. Under Application Development Features. Check the box beside .NET Extensibility 3.5. Check the box beside ASP.NET 4.0 (or greater). (Because of dependencies, certain other features will get checked automatically.)

2. Under Common HTTP Features. The boxes beside Default Document, Directory Browsing, HTTP Errors, and Static Content should be checked.

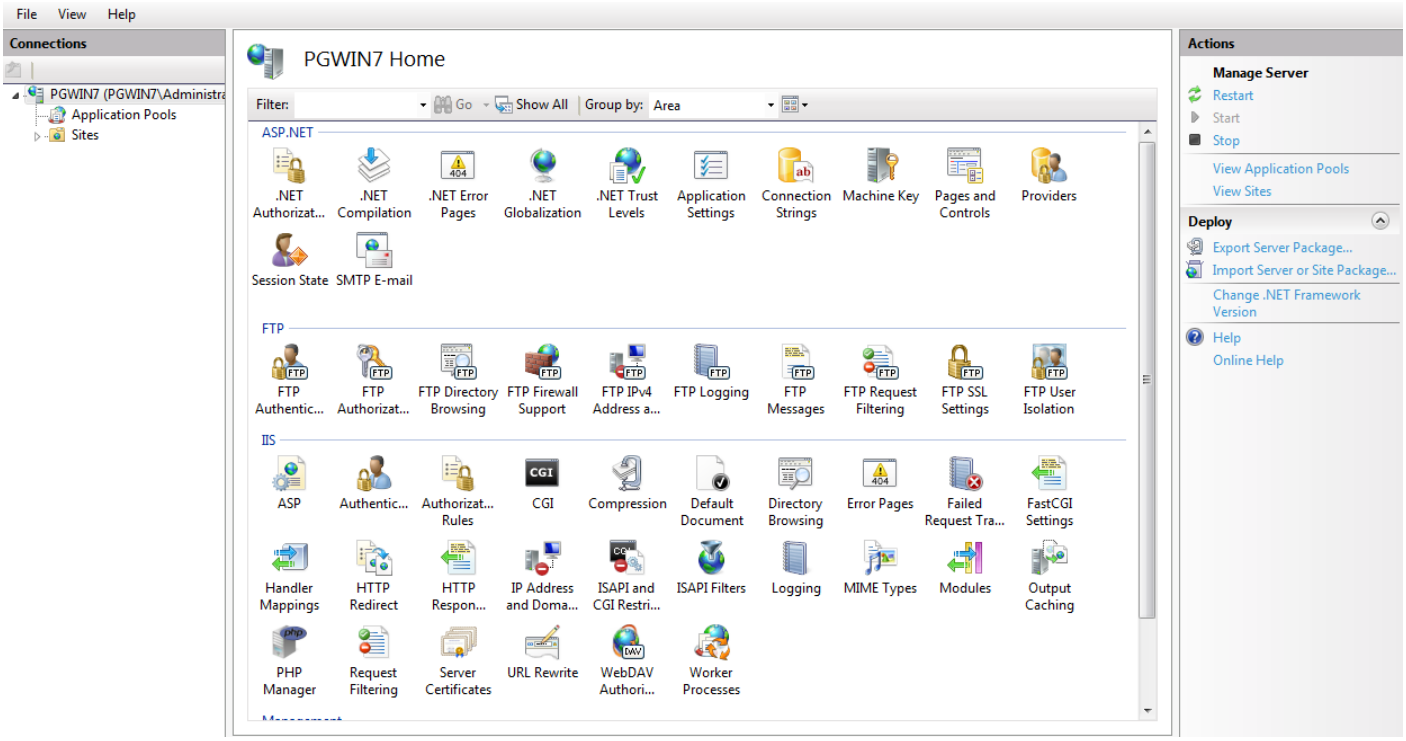
3. Under Management Tools. Check the box beside IIS Management Console.

If IIS is already running make sure that the above features are set up.



Accessing the IIS Management Console (IIS Manager)

This can be accessed on the *Tools* menu of the Systems Management console (Server Manager)



Setting up Firewall Ports

If you are running a firewall on your server then you will need to enable access for web browsers and external software using the web services to the BellHawk software running under IIS.

Normally BellHawk is set up to use port 80 for receiving HTTP and WCF messages, which by default are routed to IIS by the operating system. But other port numbers may be used if set up as part of the website setup in IIS.

If your Windows Server is protected with a software or hardware firewall then requests on port 80 (or other port to be used for the website) from the Intranet need to be allowed through this firewall.

If the BellHawk database is on another computer and that computer has a firewall, make sure that ports 1443 and 1444 are open for TCP transmission to allow remote access to the database.

Setting Up URL Routing for BellHawk

In its most simplistic form the URL can be something like 192.168.0.105, where this is an example of the local network IP address of the server on which the BellHawk website runs. By default this routes to the default website on the server, which can be set up as the BellHawk website.

You can also set up the URL to get to the website as something like 192.168.0.105:801 and then map port 801 to the BellHawk website, when you set up the IIS website parameters for BellHawk.

If you have a DNS name server on your network, you can set it up to route requests for BellHawk.ServerName.com to your BellHawk host server (such as 192.168.0.105 in our example). This will make it easier for users as they will not have to remember the numerical IP address. If you are also running the Bell-Connector software then you will need to set up a separate website URL for this website, such as Bell-Connector.ServerName.com, so using DNS routing makes it easier.

You may also have to put the routing information into the /etc/localhosts/ file on each computer if you are running a Workgroup network that is not under Domain control.

If BellHawk is to be accessed over the Internet then you can register an external URL name for your webserver (such as ServerName.com) with an ICANN registrar and then have a URL such as BellHawk.ServerName.com routed to the BellHawk website from anywhere in the world. This can be used to give access to the BellHawk website for mobile workers from anywhere they have an Internet connection, including through mobile phones and tablets on a cellular mobile data network.

If you allow external access then you need to set up your Internet firewall to route the external HTTP (or HTTPS) and SOAP/XML packets directly to the BellHawk server in a simple network setup or to your DNS server in a more complex network setup.

Set up a BellHawk User Account

An *application pool* defines a group of one or more worker processes that IIS spawns to serve requests to a website. In general, the built-in user *ApplicationPoolIdentity* is adequate for the task. However, ApplicationPoolIdentity does not have the privileges to launch an external process (a result of the standard website "Sandbox" security privilege setup).

In order to use the TAG LabelPrintS option or another option that requires launching of a background process external to the website, it is necessary to create a distinct user that is part of the local or domain Users group. You can choose any name, but here we will refer to this user as "BellHawk". User “BellHawk” should be made a member of the Users group and the IIS_IUSRS group. An application pool running as “BellHawk” will then have access to the installed website files as well as the ability to launch an external process when needed.

Please note that, if you are not using options that require the launching of external processes by the website, then the BellHawk website and its application pool can be operated using the default ApplicationPoolIdentity.

Installing the BellHawk Software

Distribution

A BellHawk system comes as a set of zipped files:

1. A backup of the BellHawk database
2. A set of files and directory folders to be copied into the folder that you will be using to hold the website code.
3. License files for your operational and test servers.
4. Optional files to support the installation of the TAG labeling option.
5. Optional files to support the installation of Weighing Scale interfaces.

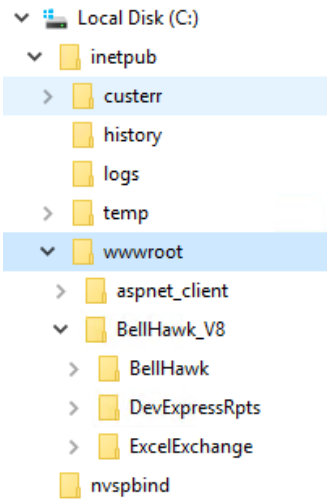
These optional files are described in the separate installation guides for these options.

Installing the Website Files

To do this you will need to be logged in as a Windows Server local systems administrator.

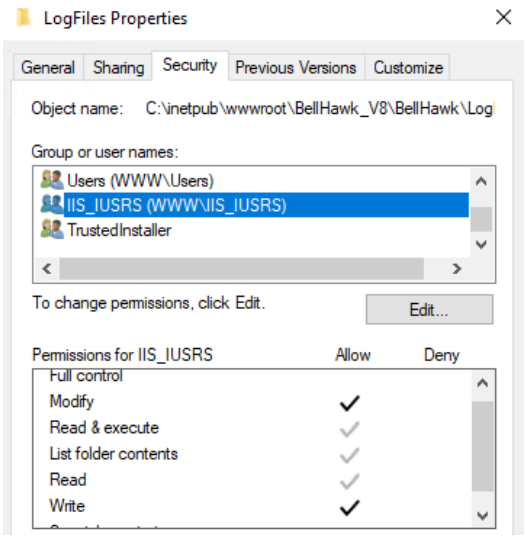
The BellHawk website software can be installed in any folder on the website server that is accessible to a member of the IIS_IUSRS user group or to the BellHawk user, if one was created. It is recommended that you install the software into C:\inetpub\wwwroot\BellHawk (or some other suitable name) as the wwwroot folder already has appropriate privileges set up for the IIS_IUSRS group and is the standard location for installing websites on a Windows Server.

After creating the BellHawk folder, then proceed as follows:



1. Use Windows Explorer to open the .zip file containing the BellHawk V8 software. Copy the folders *BellHawk*, *DevExpressRpts*, *ExcelExchange* (optional), and *SSRSrpts* (optional) from the zipped file into folder C:\inetpub\wwwroot\BellHawk_V8 (or whatever folder you are using for the website).
2. Verify that the BellHawk website (IIS_IUSRS) user group or your BellHawk user account has *read* and *execute* privileges to the newly created folder in which the website is installed. If the folder is installed in the wwwroot folder, this will normally have IIS_IUSRS group read and execute privileges. Edit the security settings for the folder and add the BellHawk user account if needed to the list of user names and allow read and execute to the permissions for this account.

3. Add *write* and *modify* privileges for the IIS_IUSRS group or the BellHawk account to the folders BellHawk\LogFiles, BellHawk\App_Data and ExcelExchange\App_Data.



4. If you received the license file BH.lex separately, copy it into the folder C:\inetpub\wwwroot\BellHawk_V8\BellHawk\App_Data.
5. Copy the web.config and the BHSDK.ini files from the subfolder BellHawk\Supporting to the BellHawk folder.
6. Copy the files DevExpressRpts.ini and web.config from the subfolder DevExpressRpts\Supporting to the DevExpressRpts folder.
7. Copy the files ExcelExchange.ini and web.config from the subfolder ExcelExchange\Supporting to the ExcelExchange folder.
8. Copy the files SSRSrpts.ini and web.config from the subfolder SSRSrpts\Supporting to the SSRSrpts folder.
9. If using the SSRSrpts application, edit the Reporting Services role security assignments to grant the built-in IIS_IUSRS group *Browser* access to folders containing reports that will be viewable through the application.

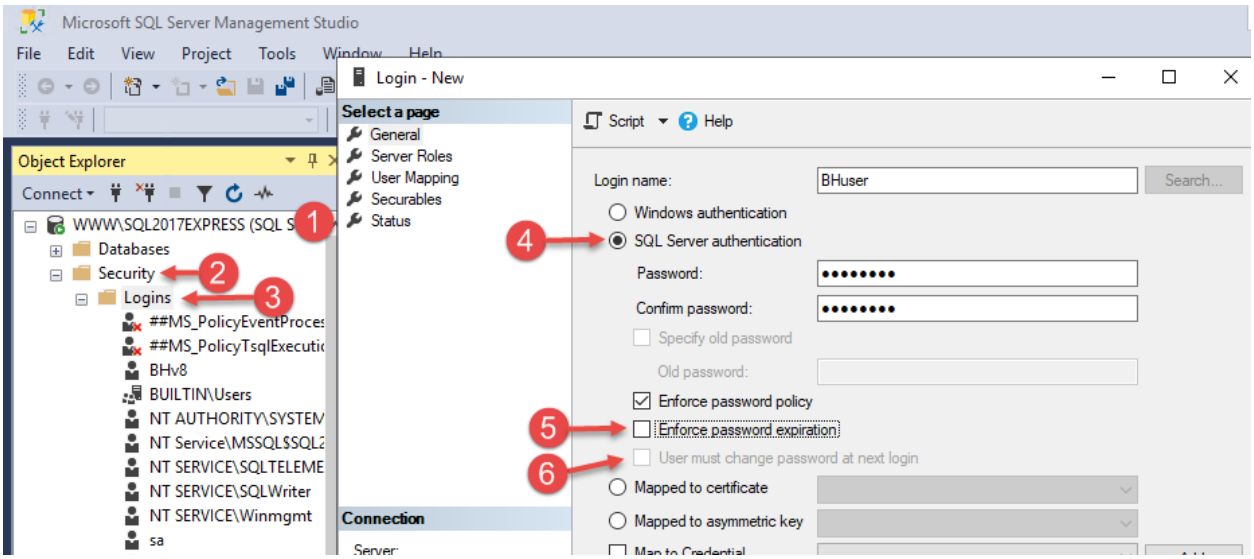
Setting up SQL Server

BellHawk will work with SQL Server Standard and Express editions but the standard edition is only needed if the size of the database is expected to exceed 10 GBytes or SSRS is to be used for reporting. For reference, the size of the BellHawk database at time of installation is under 50Mbytes.

The SQL Server instance you will use for the BellHawk database needs to be configured for Mixed Mode Authentication, i.e. both Windows Authentication and SQL Server Authentication. If you do not have an instance that will work in mixed mode, please create a new instance that will work in mixed mode.

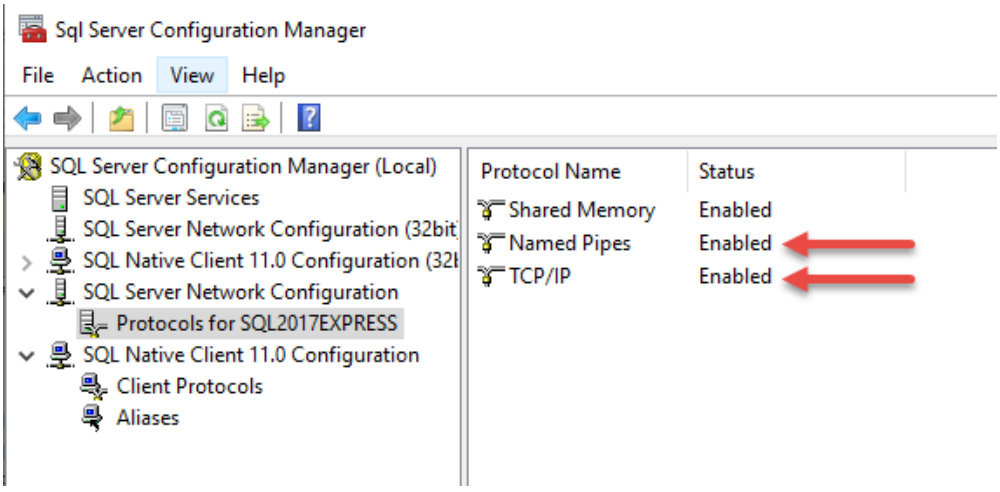
You will need to create a new SQL Server user to be granted full access (read, write, execute) to the BellHawk databases. This is because the use of the "sa" (system administrator) user for user

logins to the database is precluded. In order to add a login for the user, in SQL Server Management Studio expand the target server instance (1), Security (2) and Logins (3) to see a list of existing logins for the that instance. Right click on *Logins* and select *New Login*. Select SQL Server Authentication (4) and fill in the *Login Name* and *Password*.



We usually recommend turning off the Enforce password expiration (5) and User Must Change password (6). Because this login will probably not be used regularly by an interactive (human) user, warnings about upcoming password expiration will likely not be noticed until the BellHawk system suddenly becomes inoperable because it cannot access its own database.

Use *SQL Server Configuration Manager* to be enable **TCP/IP Protocol** and **Named Pipes Protocol** for the server instance being used for the BellHawk database.



If the SQL Server instance that you will be using for the BellHawk database is not on the same Windows Server as the BellHawk website, then:

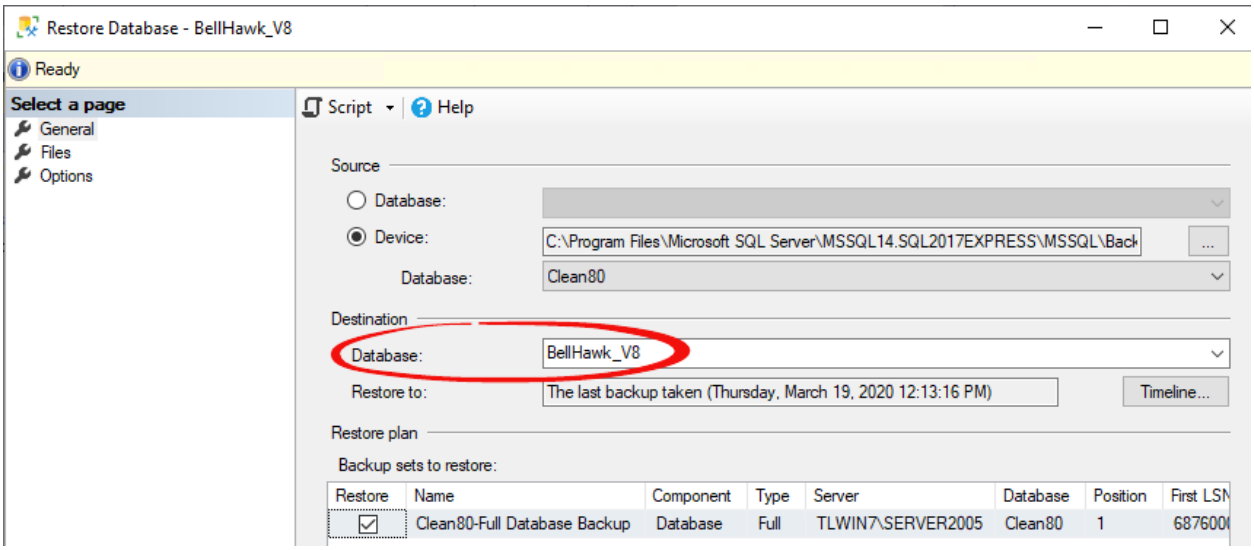
1. The SQL Server needs to be configure to allow **Remote Server Connections**

2. If there is a firewall between the SQL Server Computer and the IIS Windows Server, then ports 1433 and 1434 need to be open to enable Read/Write access from the BellHawk website to the BellHawk database.

Setting up Databases

Installing the BellHawk database

You will need to use SQL Server Management Studio to install a new BellHawk database. Extract the backup (.bak file) from the .zip archive provided, then use the Management Console expand *Databases* on your server instance. Right click *Databases* then select *Restore Database*. Select *Device* as the source and browse to the aforementioned .bak file. Assign the destination database a meaningful name as shown below. We recommend that you go to the *Files* page/tab to verify that the database filenames are being set to something appropriate for your system; some versions of SQL Server Management Studio try to preserve the original database filenames, which can be confusing.



Then you must make this database accessible to the SQL Server user login created previously.

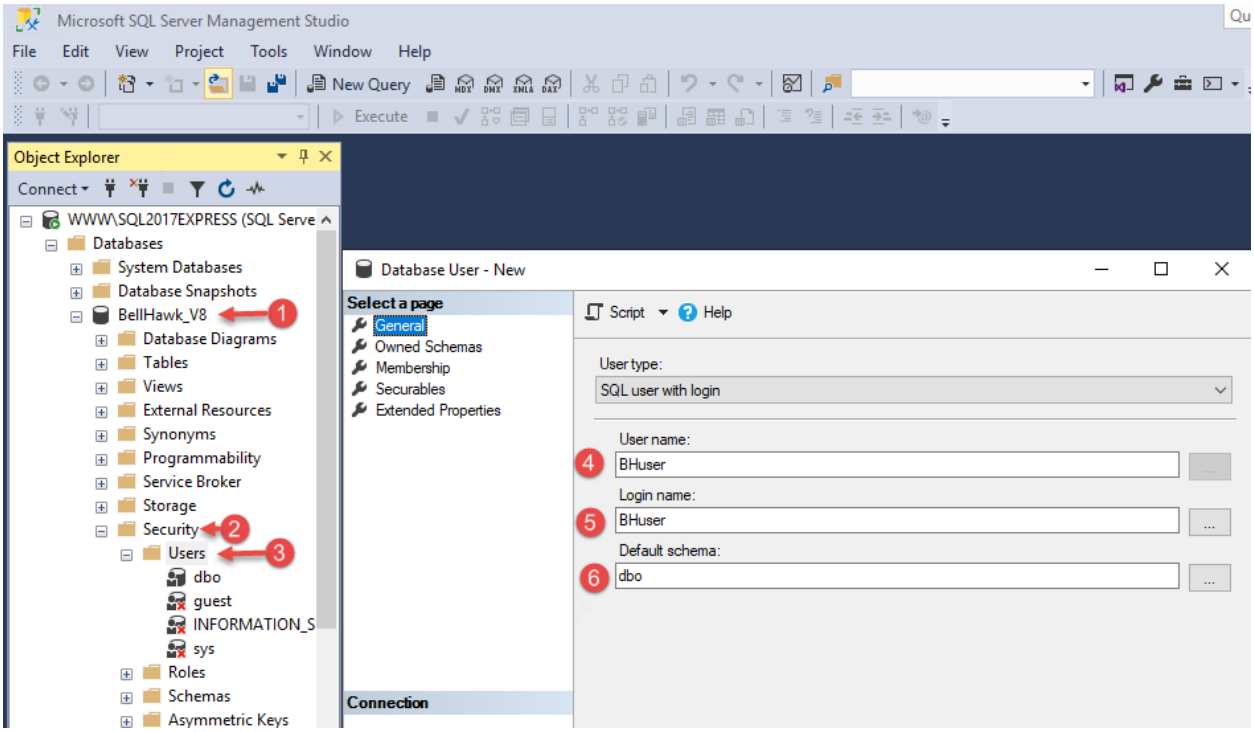
To do this:

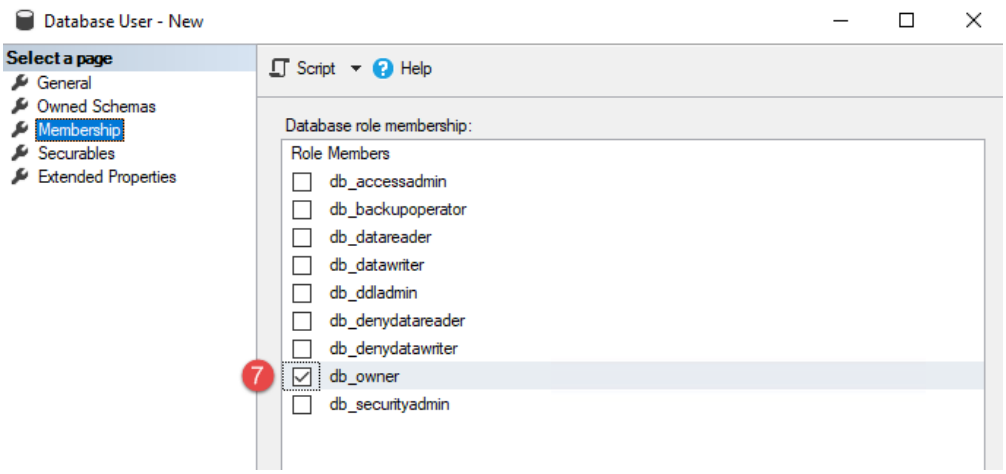
1. Select your BellHawk database
2. Expand *Security* (2)
3. Right Click *Users* (3) and select *New User*. The *General* tab dialog opens.
4. Enter *User name* (4) by which you want to identify this user for this database. For simplicity, we typically match the User name to the Login name set up for BellHawk on the SQL Server instance.

5. Enter the *Login name* (5) that was set up for the BellHawk user for this SQL Server instance.

6. Enter “dbo” as the *Default schema*

7. Select the *Membership* tab and check **db_owner** role membership (7) to grant full access to database tables, stored procedures and functions.





BellHawk requires that Common Language Runtime (CLR) execution be enabled on the SQL Server. Confirm this is enabled with the following SQL Server command:

```
EXEC sp_configure 'clr enabled';
```

If the "config_value" returned is 0, set the option value to 1 with the following SQL Server command sequence:

```
EXEC sp_configure 'clr enabled' , '1';
RECONFIGURE;
```

Recent security enhancements in Windows may require extra steps to label as "trustworthy" the assembly containing the custom CLR code in the BellHawk database. Assuming your BellHawk database is name "BellHawk_V8" execute the following query:

```
SELECT [BellHawk_V8].dbo.bh_JEX('','')
```

If the result is a Microsoft .NET error that includes the message "Could not load file or assembly 'bh_jex'", execute the following SQL Server command sequence:

```
USE [BellHawk_V8]
GO
EXEC sp_changedbowner 'sa'
GO
ALTER DATABASE [BellHawk_V8] SET TRUSTWORTHY ON;
GO
```

The query from the previous step should now execute without error.

Connecting the BellHawk database

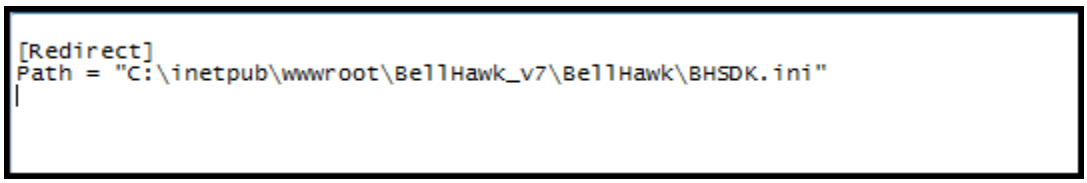
In the BHSDK.ini file in the website folder edit the user name and password for both the [BellHawk] database (1) and the [Control] database (2) to be the same as the User Login that you created above, using the name and password you set up for the BellHawk user for the SQL Server instance, as shown below:

```
[BellHawk]
DBType = SQLSERVER
XMLFile = BHMeta_v6.40.XML
Type = ODBC
Server = PGWIN7\SQLSERVER2008
Database = V66
UserID = V66DB
Password = V66db45

[Control]
; Replace the values on the next 4 lines with your BellHawk BHCTL SQL server database/login settings
Server = PGWIN7\SQLSERVER2008
Database = V66
UserID = V66DB
Password = V66db45

[Debug]
DebugLevel = 0
```

Please note that the [Control] section (2) should point directly to the BellHawk database and not to a separate ODBC Control interface. This provides the direct references to tables in BellHawk,

which are used for quick read access to the control data, whereas all writes to the BellHawk database are done through the BHSDK ODBC mechanism, specified in the [BellHawk] section (1) to ensure a high level of data checking for bad data and also to perform logging for Part 11 compliance tracking.

The DevExpressRpts, ExcelExchange and SSRSrpts applications can gather the necessary connection information from the BHSDK.ini file. Edit the file DevExpressRpts\DevExpressRpts.ini so that the Path parameter is set to reflect the full path to BHSDK.ini for your installation. Make the same change to the files ExcelExchange\ExcelExchange.ini and SSRSrpts/SSRSRpts.ini, if those applications are to be employed.



If using the SSRSrpts application, add an instruction indicating the URL of your local report server to the [Control] database (2) section of BHSDK.ini in the following format:

```
ReportServerURL = http://server2012r2/ReportServer_SQLSERVER2014
```

Session State Database

BellHawk makes use of an ASP.NET SQL session state database as part of the IIS infrastructure. If such a database does not yet exist, it needs to be created.

Although the program aspnet_regsql.exe has a wizard capable of creating a SQL Server database that supports a variety of ASP.NET applications, to install and configure the database to support session state requires that the aspnet_regsql.exe tool be run at the command line. The program can be found in the C:\Windows\Microsoft.NET\Framework64\v4.0.30319 folder. It must be run as a system administrator to work correctly.

Supply the following information in the command line:

1. The name of the SQL Server instance, using the -S option.
2. The logon credentials for an account that has permission to create a database on SQL Server. Use the -E option to use the currently logged-on user, or use the -U option to specify a user ID along with the -P option to specify a password.
3. The -ssadd command-line option to add the session state database.
4. The –sstype c option to specify a custom session state database.
5. The name BellHawkASPState, using the –d option

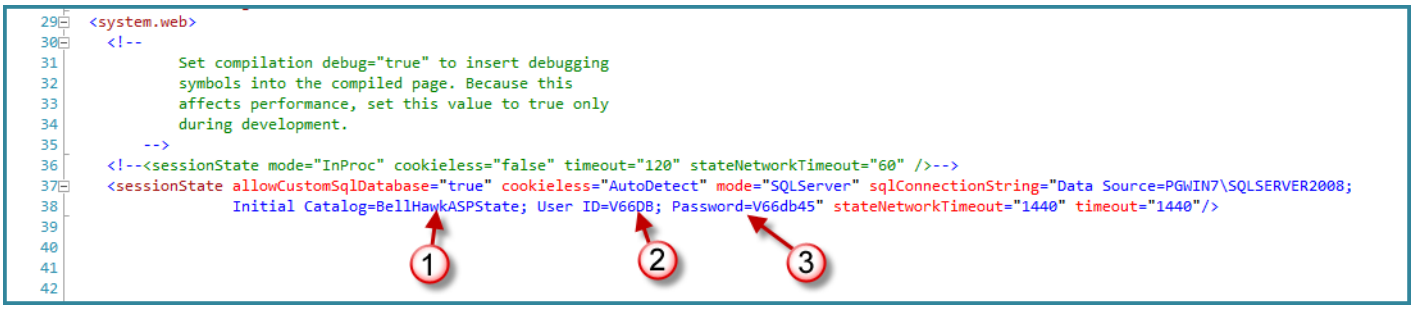
Example:

```
C:\Windows\Microsoft.NET\ Framework64\v4.0.30319\Aspnet_regsql.exe -S
WWW\SQL2017EXPRESS -E -ssadd -sstype c -d BellHawkASPState
```


The success of this step can be confirmed by using SQL Server Management studio. There should be a new database named BellHawkASPState composed of the two tables dbo.ASPStateTempApplications and dbo.ASPStateTempSessions.

Then go to this *Security* features of this new database and add the BellHawk user login, granting **db_owner** membership exactly the same as previously described for the BellHawk database.

Then edit Web.Config using Visual Studio (or a simple text editor):




```
<system.web>
  <!--
          Set compilation debug="true" to insert debugging
          symbols into the compiled page. Because this
          affects performance, set this value to true only
          during development.
      -->
  <!--<sessionState mode="InProc" cookieless="false" timeout="120" stateNetworkTimeout="60" />-->
  <sessionState allowCustomSqlDatabase="true" cookieless="AutoDetect" mode="SQLServer" sqlConnectionString="Data Source=PGWIN7\SQLSERVER2008;
                Initial Catalog=BellHawkASPState; User ID=V66DB; Password=V66db45" stateNetworkTimeout="1440" timeout="1440"/>
```

Here, we see our Session State database name (1) that matches the session state database we created previously along with our BellHawk user name (2) and login (3) that we set up for the user of for this database.

Remote Access to SQL Server

If you have your BellHawk database running on a different server from your BellHawk website, then you need to open up TCP ports 1433 and 1434 through the remote database server's firewall.

You will also need to install the SQL Server client tools on the server used for the website. These will normally include a copy of the SQL Server Management Studio, which can be used to test the connection to the remote database.

If you do not install the SQL Server Management Studio on the web server computer then use Access or some similar program to link to the remote database as a test.

Creating the BellHawk Website

Creating a Web Server Application Pool for BellHawk

An application pool is the dynamic set of web pages in use by a website. You should create a special application pool for your BellHawk website. To do this, proceed as follows:

1. Start IIS Manager. In the *Connections* frame, expand the entry that shares your computer name. Right-click on *Application Pools* and select *Add Application Pool*.

2. Enter BellHawk as the name of the new application pool, set its .NET Framework version to v4.0, and click OK to accept defaults for the other settings.

3. Select the new BellHawk entry in the application pool list, and in the *Actions* frame click *Advanced Settings*. *Enable 32-Bit Application*, *Start Automatically*, and *Load User Profile* should all be set to True.

4. If you are using an externally launched option such as TAG, change the *Identity* property of the BellHawk Application Pool to the "BellHawk" user account you created or the BellHawk website. In IIS7 Manager this is found in *Advanced Settings*, under *Process Model*. This is not necessary if TAG, or some other option automatically launched by BellHawk, is not being used. If this is not changed then the Application Pool processes are run with the default ApplicationPoolIdentity identity, which by definition is a member of IIS_IUSRS.

Creating the BellHawk Website

In the IIS Manager, right-click on Sites and select Add Web Site.

After assigning a site name (such as BellHawk), be sure to select the application pool BellHawk.

For Physical path, use the ellipsis button to browse to the folder where the site content is stored. Use C:\inetpub\wwwroot\BellHawk_V8\BellHawk\ or wherever the alternate folder is located.

Set the connection identity. If using TAG, in the IIS7 Manager click Connect as, select Specific user, and enter the credentials for the BellHawk account you created for this website. Click Test Settings... to confirm that the account and physical path are compatible. This can be left set to the default ApplicationPoolIdentity if TAG is not being used.

Set Binding Type, IP address and Port according to your requirements. Port 80 is the default port for HTTP requests, and the firewall should already be configured to accept connections on it. If the port number is set to 80, the port number can be left off the URL entered by users (example http://server_name or http://192.168.1.3)

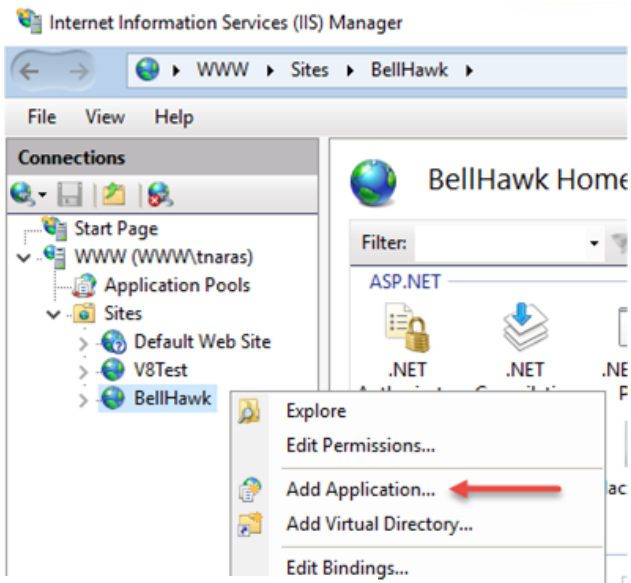
Assign a Host name if you want to assign more than one host name to one computer that uses a single IP address. If you specify a host name, clients must use the host name instead of the IP address and optionally a port to access the Web site.

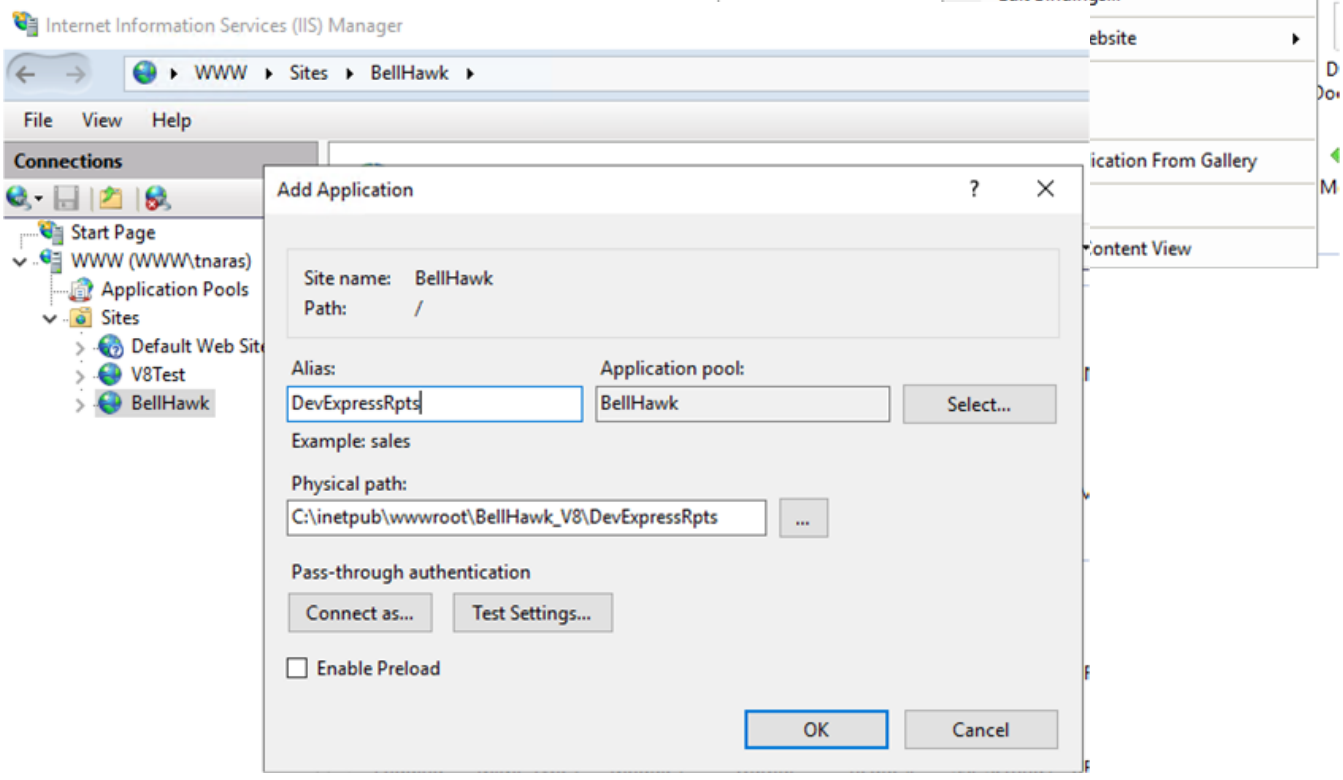
In the IIS Features group double-click the Authentication icon to configure the authentication settings for the site. Both Anonymous Authentication and Forms Authentication must be enabled. The other authentication types are not used by BellHawk and may be disabled for this site.

Adding Reports Application

To enable the DevExpress report viewing feature, right-click on your BellHawk website and select Add Application. Assign "DevExpressRpts" as the Alias, and keep the default application pool (same as the website) for the application.

For Physical path, use the ellipsis button to browse to the folder C:\inetpub\wwwroot\BellHawk_V8\DevExpressRpts \ or wherever the alternate folder is located.





Adding Excel Application

This follows a similar process to the Reports Application.

To actvate the ExcelExchange data exchange application, right-click on your BellHawk website and select Add Application. Assign "ExcelExchange" as the Alias, and keep the default application pool (same as the website) for the application.

For Physical path, use the ellipsis button to browse to the folder C:\inetpub\wwwroot\BellHawk_V8\ExcelExchange\ or wherever the alternate folder is located. Make sure that the IIS_IUSRS group or the BellHawk account has read & execute, write and modify privileges to the ExcelExchange\App_Data folder.

Setting up WCF

Both BellHawk V8 and the Label Printing Web-Services Interface employ the Windows Communication Foundation (WCF) part of the .NET Framework in handling SOAP/XML requests. To register WCF, Perform the following steps:

1. Start the command window as administrator: from the taskbar click Start -> All Programs -> Accessories, right-click Command Prompt and select Run as administrator.
2. Navigate to "C:\windows\Microsoft.NET\Framework64\v3.0\Windows Communication Foundation\".
3. Run the following command: `ServiceModelReg.exe /i /x`
4. Exit the command window.

Please note that this is only necessary of you are using TAG, the weighing scale interface, or have an external program that is interacting with BellHawk using its web services (SOAP/XML) interface.

Testing the Website

Use a PC and enter the URL that you set up for the BellHawk website. You should see the pre-login BellHawk splash screen. If not then you will probably get an error message from IIS that it cannot find your BellHawk website.

Alternately you may be informed of a Runtime Error, and that "another exception occurred while executing the custom error page for the first exception." In this case, a more useful error message can be accessed by making a temporary copy of the *Web.config* in the BellHawk site folder in order to change the configuration setting

```
<customErrors mode="On" defaultRedirect="AppErrorPage.aspx" ></customErrors>
```

to

```
<customErrors mode="Off" />
```


A message indicating that the system was unable to load some Http module is not uncommon, and usually reflects a problem with how Microsoft's installation and activation software set up multiple versions of .NET and left IIS configured with conflicting modules and handlers.

It can usually be corrected by re-registering ASP.NET 4.0, although details of the fix depend on the specific versions of the OS and IIS running on your server. A copy/paste of the error into an internet search engine should yield a number of more detailed explanations of the problem and a solution for your particular platform.

Commentary

Setting up a BellHawk website is a complex process in which it is easy to make a mistake. Also many of the above instructions apply to a simple single server configuration using Windows 2019.

You may need to make adjustments to these instructions based on your specific network and server configuration..

It is a recommended:

1. To set up the IIS server with a default website that has a "Hello World" test HTML page. This can then be used to make sure that IIS is working correctly.
2. To test the BellHawk website using a numeric URL and possibly alternate port number before proceeding to test DNS routing.
3. To test access to the BellHawk website over the LAN before proceeding to test access from the Internet.
4. To test network routings to the server using PING to make sure it is reachable from each device and that you have allowed access through all intervening firewalls.

