Windsor Solutions



IT Administration Guide

Updated for Version 4.17

Revision Date: September 28, 2023.

THIS PAGE INTENTIONALLY LEFT BLANK

**Contents**

[Overview 1](#_Toc64451581)

[Architecture 2](#_Toc64451582)

[nFORM Build and Deploy Process 2](#_Toc64451583)

[Implementation Steps 4](#_Toc64451584)

[1. Establish Prerequisites 4](#_Toc64451585)

[2. Server Configuration and Verification 4](#_Toc64451586)

[3. Deploy the Database 8](#_Toc64451587)

[4. Configure the System 9](#_Toc64451588)

[5. Copy the Source Files 9](#_Toc64451589)

[6. Test the System 9](#_Toc64451590)

[Configuration Options 11](#_Toc64451591)

[Site Branding 11](#_Toc64451592)

[Lookups 11](#_Toc64451593)

[Email and Message Templates 11](#_Toc64451594)

[Configure System to Support Anonymous/Guest Submissions 12](#_Toc64451595)

[System Configuration 12](#_Toc64451596)

[Other Configuration Options 19](#_Toc64451597)

[System Administrative Features 20](#_Toc64451598)

[Accessibility 22](#_Toc64451599)

[Frequently Asked Questions 23](#_Toc64451600)

[Appendices 25](#_Toc64451601)

[Appendix A - Application Pool Configuration 25](#_Toc64451602)

THIS PAGE INTENTIONALLY LEFT BLANK

**Version Control**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Author | Date | Notes |
| 1.0 | Windsor Solutions | 4/22/2013 | Initial version. |
| 2.1.1b | Windsor Solutions | 12/12/2013 | Adjusted implementation steps based on most recent feature set. |
| 2.2.6 | Windsor Solutions | 3/12/2014 | Adjusted browser compatibility. |
| 2.2.8 | Windsor Solutions | 4/8/2014 | Minor cleanup. |
| 2.2.11 | Windsor Solutions | 5/28/2014 | Added App Pool Autostart configuration details. |
| 3.4 | Windsor Solutions | 12/23/2015 | Adjustments for release 3.4. |
| 4.1 | Windsor Solutions | 2/22/2017 | Adjusted for Release 4.1 |
| 4.2 | Windsor Solutions | 5/22/2017 | Adjusted for Release 4.2 |
| 4.2/4.3 | Windsor Solutions | 12/20/2017 | Further adjustments for release 4.24.3. |
| 4.4 | Windsor Solutions | 1/12/2018 | Updates to reflect v4.4 |
| 4.4.1 | Windsor Solutions | 4/24/2018 | Updated to include additional attributes available in the email templates. |
| 4.5 | Windsor Solutions | 6/28/2018 | Updated to reflect v4.5. |
| 4.6 | Windsor Solutions | 3/6/2019 | Updated to reflect v4.6. |
| 4.6 | Windsor Solutions | 5/10/2019 | Added Anonymous/Guest User config options. |
| 4.7 | Windsor Solutions | 5/30/2019 | Updated to reflect v4.7. |
| 4.8 | Windsor Solutions | 2/7/2020 | Updated to reflect v4.8. |
| 4.9 | Windsor Solutions | 7/27/2020 | Updated to reflect v4.9. |
| 4.9 | Windsor Solutions | 9/4/2020 | Updated to add secondary payment address. |
| 4.10 | Windsor Solutions | 9/4/2020 | Updated to reflect v4.10. |
| 4.10 | Windsor Solutions | 12/21/2020 | Added clarification with regards to use of app.secondaryPayment attributes. |
| 4.11 | Windsor Solutions | 2/17/2021 | Updated to reflect v4.11 |
| 4.12 | Windsor Solutions | 9/20/2021 | Updated to reflect v4.12. |
| 4.13 | Windsor Solutions | 5/2/2022 | Updated for 4.13 |
| 4.14 | Windsor Solutions | 8/31/2022 | Updated to reflect release 4.14. |
| 4.15 | Windsor Solutions | 12/19/2022 | Updated to reflect 4.15. |
| 4.16 | Windsor Solutions | 5/11/2023 | Updated to reflect 4.16. |
| 4.17 | Windsor Solutions | 9/28/2023 | Updated to reflect 4.17 |

THIS PAGE INTENTIONALLY LEFT BLANK

# Overview

This document is intended for IT administrative staff of the nFORM system, as a reference and guide when administering the system from a technical perspective (e.g., making configuration changes or while planning a move to a new hosting environment). Each installation of the nFORM system will have its own specific configuration files and templates that will be utilized to customize the application. Changes applied to one environment must also be made in all other environments for that client in order for them to take effect.

# Architecture

Please see the nFORM Technical Architecture document for details on the nFORM technical architecture.

# nFORM Build and Deploy Process

To facilitate the downloading, compiling, and deployment of the source code, Windsor uses the Jenkins Continuous Integration Server.

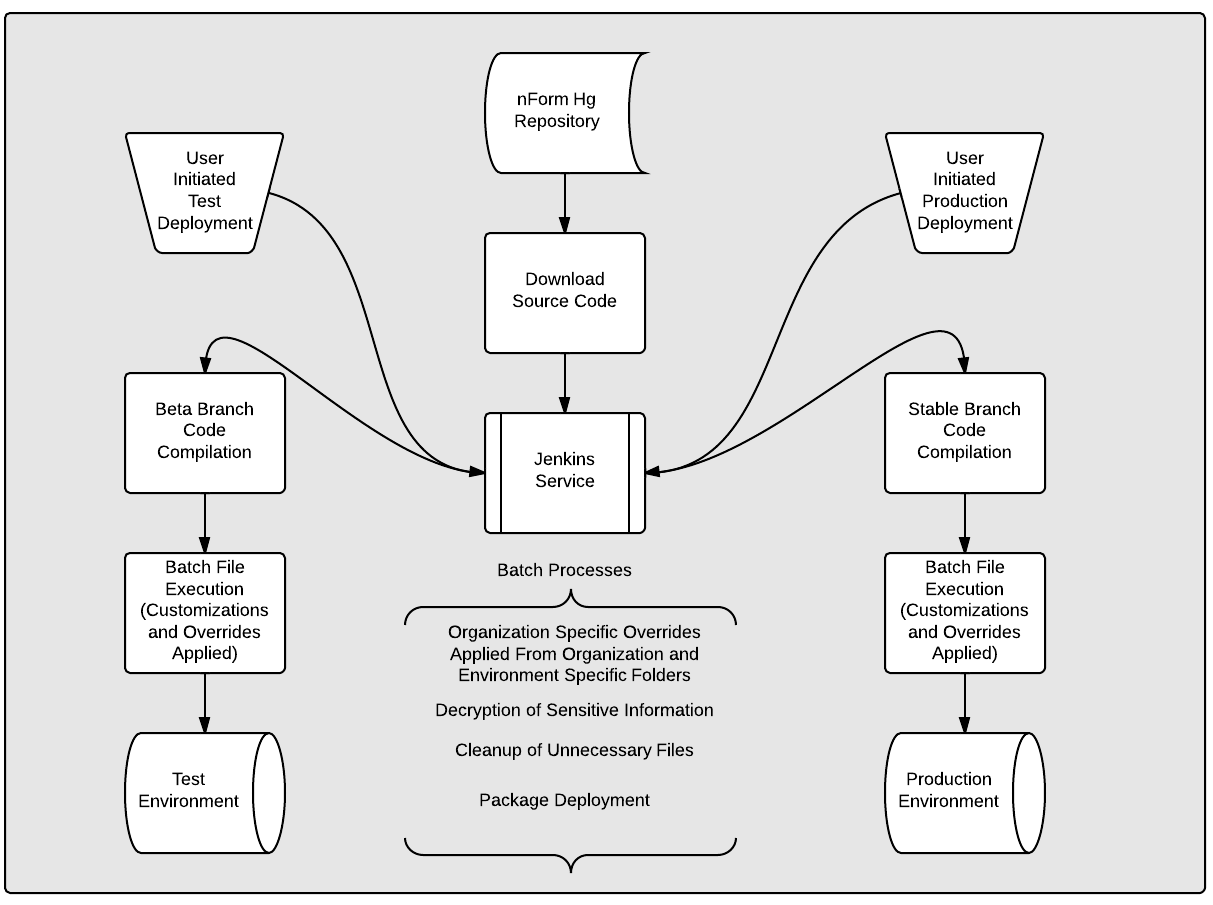


Figure 1: Build and Deploy Process Diagram

### GitLab Source Code

The test and production systems subscribe to Windsor GitLab server to retrieve deployed builds from Windsor’s code repository. The test server is subscribed to the beta branch of code which will contain new features to nFORM that have been made available for beta testing. The stable branch is the latest production version of nFORM.

### Jenkins Continuous Integration Server

nFORM subscribers who choose to use the continuous integration capabilities for publishing code to their test and production environments. Jenkins functions as a gateway for downloading and compiling code from the beta and stable code branches, compiling the code, and deploying the code to the test and production environments. Although Jenkins can be scheduled to build and deploy automatically when changes are detected, Windsor recommends that the build and deploy process only be started by manually initiating it through the Jenkins interface when you are ready to upgrade. There are separate jobs for both the test and production environments in a properly configured Jenkins setup and each can have their own set of parameters to pass into the batch processes for correctly configuring your application.

**Note:** *Windsor maintains a set of configuration files that have been set to match your configuration for internal testing. By default, this file is deployed with the application by Jenkins, and you will need to send your modified and tested configuration files to Windsor. It is highly recommended to maintain backup[[1]](#footnote-1) sets of configuration files.*

### Batch Processes

Windsor has developed a batch file that will be triggered when Jenkins is deploying to either environment. In addition to cleaning out the files that are not required for the operation of the system, parameters can be set that will

* determine the publishing location,
* determine any configuration files to override, and
* decrypt any sensitive information.

### Account used by Jenkins

A local IIS User account is required to allow Jenkins to publish code to the different environments. This account is used by Jenkins to deploy files to the web directory used by the system. If an existing account is moved, then the new directory will require deployment configuration to be established using IIS.

When establishing this deployment account within IIS, the minimum permissions necessary to publish to the Website in IIS and restart the application pool must be created and assigned. Jenkins uses a batch file to obtain the secure credentials necessary for executing the deployment process. The bat file is used to gather credentials stored in the file: "c:/ci/credentials/[servername].txt".    This will allow web deploy the access need to set up the ACL’s.

# Implementation Steps

The nFORM system is designed to run in a wide variety of Microsoft-based environments and the implementation is relatively simple once the prerequisites are met. The following steps will be used to bring nFORM online with a new Windows Server:

1. Establish Prerequisites
2. Server Configuration and Verification
3. Deploy the Database
4. Configure the System
5. Copy the Source Files
6. Test the System

The sections below will describe each of these steps in detail.

## Establish Prerequisites

nFORM is designed for any system meeting the minimum requirements described in the nFORM System Requirements document.

## Server Configuration and Verification

### IIS Configuration

#### Required IIS Components

The following components will need to be activated/installed with the IIS service:

* Dynamic Compression

#### Application Pool Settings

It is recommended that a new application pool specifically be allocated for the nFORM application. This application pool must be configured with the following properties:

* .Net Framework 4.8.2
* Integrated mode
* Idle-Timeout value must be set to 0
* Recycling Specific Times value must be set to the specific time the application will be reset on a daily basis, typically in the middle of the night (e.g., 04:30:00). See Application Pool Recycle Scheduling in the appendix for further details.
* See Autostart Configuration section below for additional details on establishing process to maintain Application Pool readiness.

#### Virtual Directory Creation

The following will need to be established to prior to application publishing:

* Create a target folder on the Web Server where the application files will reside.
* Create a virtual directory in IIS, mapped to the file system folder containing the application files.
* Map the virtual directory to the established Application Pool.

#### Autostart Configuration (Optional)

For optimal application performance, the nFORM application should be autostarted daily to prevent the first user who uses the system from having to wait for the application to wake up and perform any pre-caching that is needed to run the application. The section below provides details for configuring Autostart for the differing versions of IIS.

##### IIS 7.0 and prior

IIS 7.0 does not provide built in capabilities to “autostart” an application so a process will need to be established to maintain Application Pool readiness. ARDEQ built a custom utility to perform this wake-up process which can be leveraged for older versions of IIS.

##### IIS 7.5 - Windows Server 2008 R2

IIS7.5 provides preload functionality through a special provider module. When configured this module calls a custom preload function in the template. The preload function simply calls the website using the localhost address (ie, <http://localhost/nform>). The provider must be registered at the server level before it can be used on individual sites.

All references to applicationhost.config refer to **C:\Windows\System32\inetsrv\config\applicationhost.config.**

###### 1. Configure the application in IIS

* Ensure nFORM has a dedicated pool set to Start Automatically=True
* Include the manually specified pool recycle time as well, taking into account the server time zone offset.
* The root website must be configured to resolve as localhost. As such only one root level website per machine can use the auto start feature under IIS 7.5.  The web site bindings must contain a port 80 binding entry for either **localhost** or **\***.

###### 2. Configure the Windsor Preload service provider

* The Windsor.Commons.AspNet.dll included in the /lib directory of nFORM provides a function used to warm the cache in the event of an application pool recycle.
* If this is the first nFORM install on a server, add the definition of the WindsorPreload service into the config file. This is only necessary ONCE.
* Insert the following code inside the **applicationhost.config** <system.applicationHost> section:

<serviceAutoStartProviders>

<add name="WindsorPreloader" type="Windsor.Commons.AspNet.Preloader, Windsor.Commons.AspNet" />

</serviceAutoStartProviders>

###### 3. Configure the application pool for auto start

* Open applicationhost.config and search for your application pool in the <applicationPools> section
* Modify the pool to include autoStart=”true” and startMode=”AlwaysRunning”:

<add name="nFORM" managedRuntimeVersion="v4.0**" startMode="AlwaysRunning" autoStart="true"** >

###### 4. Modify the IIS application to use the WindsorPreload service

* Open applicationhost.config and find the application in the <sites> section
* Modify the application to reference the WindsorPreloader library with **serviceAutoStartEnabled="true"** and **serviceAutoStartProvider="WindsorPreloader"**
* Alternatively, you can enter this value directly in IIS. Select the root web site and click the Configuration icon from the main panel. In the drop down at the top, expand system.applicationHost and

<application path="/nform" applicationPool="nFORM**" serviceAutoStartEnabled="true" serviceAutoStartProvider="WindsorPreloader"**>

##### IIS8 - Windows Server 2012

IIS8 is very easy to configure for preload using the new **Application Initialization** module. When enabled, this module will hit the site when the pool is started, ensuring any initialization functions are run BEFORE a user hits the site. No special code is necessary.

<http://blogs.iis.net/wadeh/archive/2012/04/16/re-introducing-application-initialization.aspx>

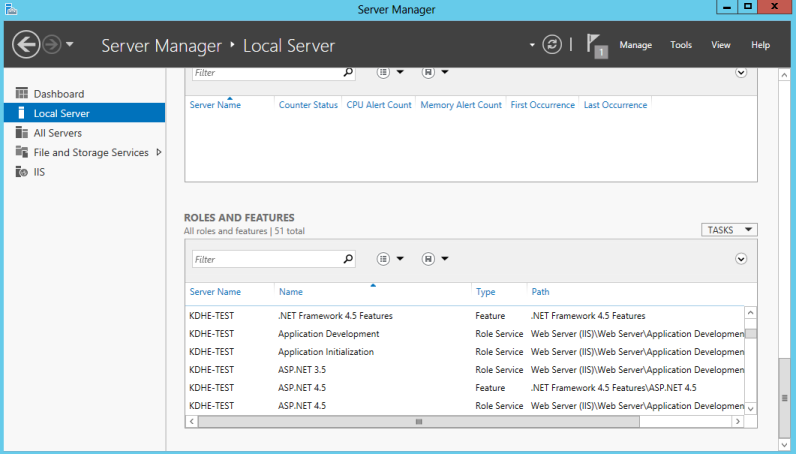
<http://blogs.iis.net/wadeh/archive/2012/05/01/application-initialization-part-2.aspx>

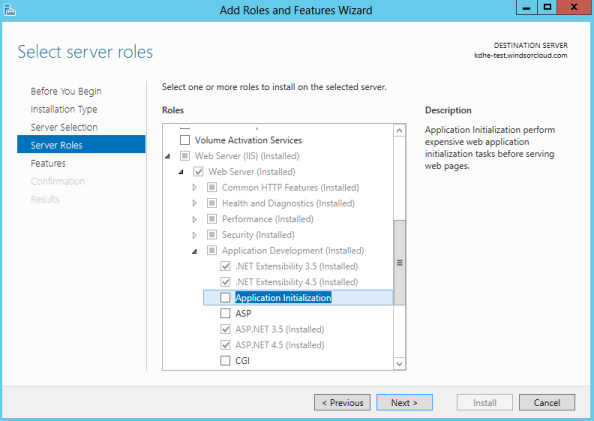
<http://www.iis.net/learn/get-started/whats-new-in-iis-8/iis-80-application-initialization>

###### 1. Install required IIS components

The default Server 2012 install includes .Net 4 but no Web Server. You must ensure the required roles and features are installed on the server prior to application deployment.

* Run the Server Manager application
* Select Local Server from the left menu
* Scroll to the bottom to the Roles and Features section
* Verify the Web Server Role is selected
* Under the Web Server Role Features, in addition to default selections ensure the following are also selected
  + Performance
    - Dynamic Content Compression
  + Application Development
    - .NET Extensibility 3.5
    - .NET Extensibility 4.5
    - Application Initialization
    - ASP.NET 3.5
    - ASP.NET 4.5
    - ISAPI Extensions
    - ISAPI Filters

****

****

###### 2. Configure the application pool

* Ensure nFORM has a dedicated pool set to .Net 4.0 Integrated mode.
* In the pool Advanced Settings set **Start Automatically=True** and **Start Mode = AlwaysRunning**
* Include the manually specified pool recycle time as well, taking into account the server time zone offset.

###### 3. Modify the IIS application for Preloading

* Select the nFORM virtual application and click Advanced Settings
* Set **Preload Enabled = True**

### Document Template Hosting (Optional)

In nFORM, designers have the ability to establish document templates (Excel, Word, PDF, etc.) which can be referenced in their forms and downloaded by submitters. As an option, a virtual directory can be established to store and host these ancillary documents. If this is needed, a publically accessible file hosting location needs to be established and made available to the public via a URL. Documents will be published to this location as needed.

## Deploy the Database

The database can be established either by utilizing a database script followed by seeding the database with any necessary initial data, or a baseline database can be restored to the environment providing the necessary database structure and seeded data. The following section describes the processes for deploying the system in either scenario.

### Scripted Database

If establishing the client database for the first time and a backup is not available for use then the nFORM SQL Database creation scripts can be run on a new database by an administrator. This will create the necessary objects and seed data. A database login will need to be established with datareader and datawriter. And the custom privileges db\_executor which enables the execution of stored procedures, for this database.

If the database already exists and is being updated, a custom database upgrade script will be established to facilitate the upgrade process. Windsor utilizes the Flyway tool to manage database versioning and scripting. Flyway will be run on the database to compare the deployed version of the database to the desired version to be deployed and will establish the appropriate script to upgrade the database.

### Restored Baseline Database

If available, the easiest way to deploy the database is to restore the Windsor-provided database. A database can typically be provided based on the clients test version which has the additional benefit of allowing established forms to be included. Test data would typically be removed. The restore can be performed directly onto the production server. After restoring the database, a database login will need to be created or reconnected to the database user using the sp\_change\_users\_login system stored procedure. Note: The database user used by the application must have db\_executor, datareader and datawriter privileges.

### FileStreaming

If FileStreaming is activated for the first time, a database script will need to be run to migrate files from the core database and to the file system. If interested in performing this process, please contact Windsor for the appropriate script.

## Configure the System

All configurations for the system are within files managed in a central directory, [Root Directory]\conf\. It is imperative that these files are configured appropriately. Key parameters such as database connection information are necessary for successful operation of the system. Incorrectly configured key parameters will prevent the application from running at all until they are corrected for the environment. Integration configuration properties (e.g., queue names and security credentials, etc.) are necessary to support integration points. Please refer to the Configuration specific section of this Administration Guide for additional details.

## Copy the Source Files

There are two alternatives for copying the source files for the application in support of the deployment process; direct file copy and continuous integration. The sections below describe each of these methods.

### Direct File Copy Approach

A direct file copy deployment approach is useful when continuous integration is not an option or where clients are not comfortable with an automated deployment process.

To perform the direct file copy method, copy all source files from the deployment package into the destination directory on the web server. After the project is built with the “**[Root Directory]\build\Deploy.bat /p**” BAT file from a machine with .NET 4 installed, the appropriate files (stripped by the process of unnecessary source code) will be located in **[Root Directory]\src\obj\Release\Package\PackageTmp\** folder.

Alternatively, the files in **[Root Directory]\src\** folder can be copied directly from the application directory of another web server that runs the application.

### Continuous Integration

Continuous integration is a great option for automating parts of the deployment process. This approach allows for automation and consistency in the process as well as performance of unit tests, as appropriate. It also allows for control of identifying particular versions of the application to deploy.

In a continuous integration environment, the deployment staff will simply need to initiate the deployment process on the continuous integration server and verify the deployment was performed correctly (no errors were encountered).

## Test the System

The application can now be tested prior to official launch. Key tests to run include the following:

1. Register a user and assign roles to them.

* Log in with the newly created user and verify that you receive the first time user message and access reflects the roles assigned.

1. Create a submission:
   * Verify that screens come up and the finder works successfully (if not, the application likely does not have database connectivity, or the jobs configuration may not be turned on)
   * Create and submit a form and verify that you received an email to the email account of the user that submitted the form.
2. Download the pdf of the submission.
3. Make a payment (if applicable).

# Configuration Options

The nFORM system is composed of the code files that reside in the IIS application directory and the nFORM database. IIS application settings and the configuration files must be deployed and configured correctly with the database setup and the SQL logins configured before the system will run correctly.

All code files needed by the application are located in the root directory of the application. Most files are used by the system and should never be modified. Some files may need to be modified if you wish to change your configuration options or implement new features, or modify the colors or style sheet properties. The following aspects of the nFORM system can be customized for your needs:

* Site Branding
* Email and Message Templates
* Application Configuration
* Other Configuration Options

Instructions on how to modify the configurable options will be detailed in the Configuration section below.

## Site Branding

The colors and some styles used by the application can be modified by a system administrator. Any system administrator performing this task should be comfortable using CSS. Windsor utilizes .less files to inject styles into the system. The .less files consist of CSS code and variables that can be set to override the base color scheme. The following directories and files those used for customizing the site styles.

* **[Root Directory]/Content/CustomSite.less**: Any organization specific style customizations can be defined in this file.
* **[Root Directory]/Content/Colors.less:** Base colors are controlled in this file.
* **[Root Directory]/Content/Custom/:** Any organization specific header graphics are stored in this directory.

## Lookups

Lookups are lists that can be stored in the database tables LOOKUP and LOOKUP\_VALUE. The LOOKUP table contains the metadata that describes the lookup, while the LOOKUP\_VALUE table uses the Name column to populate a list of items. This is ideal for lists that are commonly used, or long lists in which typing in the name makes finding the one you want much easier. Lookups can be used in a short text control with the autocomplete feature turned on, or in the single select or multi select controls. The name column will be both the value and the selected index of the dropdown array.

## Email and Message Templates

The system delivers system messages, including emails and pop-up, using templates that are configurable per installation. The templates can be modified to accommodate the structure and content a client wishes to display, in html format. This section describes how to modify the messaging templates.

See the *nFORM Email Template Guide* for additional details.

## Configure System to Support Anonymous/Guest Submissions

It is possible to configure a form for external users to fill out and submit a form without creating an account or logging in to the system, a common use case for complaint forms.

To enable a form designer to make a form available for anonymous/guest user support, the security.guestAccountCanLogin setting in the varsOverrides.config must be set to true.

## System Configuration

The system configuration is used to designate templates, configure communications with other servers, or turn on / off features or system options. XML formatted configuration files are used to set the majority of options in the nFORM system. These files include a number of name value pairs which define the desired configuration. Files are easily modified by changing the value portion of the configuration setting, however some values will be limited to a specific range of permitted values or may be limited due a client specific infrastructure. Certain values are simply true or false value while others are text.

*Note: all configuration files can be found in the* ***[Root Directory]/conf/*** *folder of the system.*

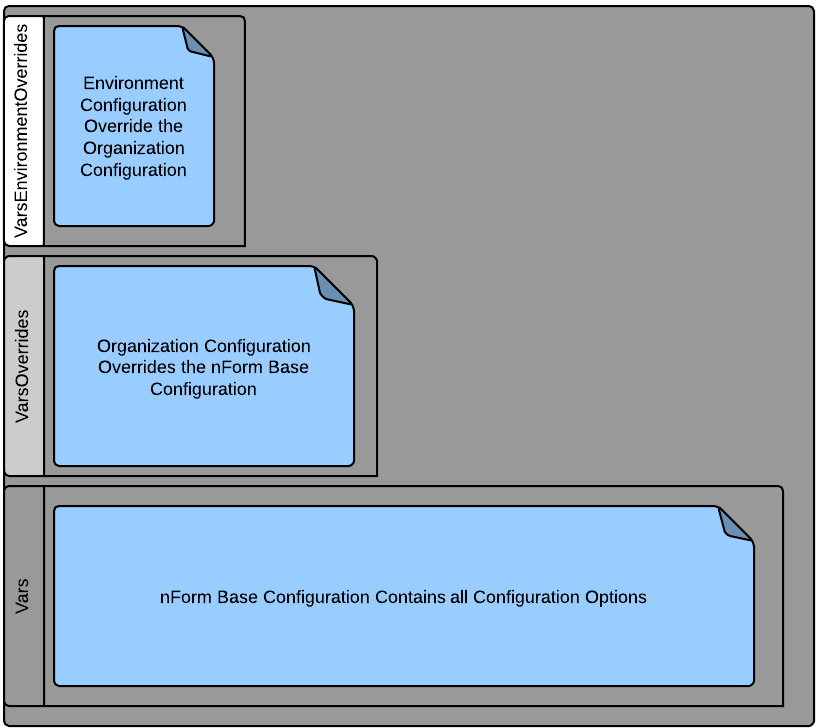


Figure 4: Configuration Hierarchy

### Relevant files

Below is an overview of configuration files and how they can be modified by system administrators to customize the nFORM application to meet their needs

* **[Root Directory]/Conf/vars.config:** All default system configuration variables are located in this file. This file sets the default configuration used by nFORM and it should NOT be modified but can be utilized as a reference for identifying (and copying) configuration properties to add to your organization specific override files explained further below. This file contains the default configuration for the application, and contains every configuration value available.
* **[Root Directory]/Conf/VarsOverrides.config:** This file allows an organization to set specific configuration settings. Any configuration values included in this file will modify (or override) the default configuration identified in the **vars.config** file.
* **[Root Directory]/Conf/VarsEnvironmentOverrides.config:** This file provides a means for an organization to identify environment specific configuration settings. Any configuration values included in this file will modify (or override) the organization specific configuration identified in the VarsOverrides.config.
* **[Root Directory]/Conf/ varsOverridesSensitive.config:** Analogous to the corresponding varsOverrides.config, but is provided for documenting sensitive settings (e.g. encryption passwords) which need to be further secured. For agencies subscribing to Windsor’s Build and Deploy process, sensitive information can be encrypted by the agency and provided to Windsor. The build and deploy process (at the agency) will have access to the decryption key for deployment, but Windsor will not have access to this configuration data.
* **[Root Directory]/Conf/ varsEnvironmentalOverridesSensitive.config:** Analogous to the corresponding VarsEnvironmentOverrides.config, but is provided for documenting sensitive settings. (e.g. encryption passwords) which need to be further secured. For agencies subscribing to Windsor’s Build and Deploy process, sensitive information can be encrypted by the agency and provided to Windsor. The build and deploy process (at the agency) will have access to the decryption key for deployment, but Windsor will not have access to this configuration data.
* **[Root Directory]/Conf/Web/connectionStrings.config:** The connection strings utilized by the system environment are defined in this file.
* **[Root Directory]/Conf/Web/authentication.config:** The authentication method and settings for the system environment are defined in this file.

*Please note that any changes to the configuration will require that the application pool is reset for the changes to fully take effect.*

### Configuration Values

Collections of like configuration items have been organized into like categories. A high-level synopsis of the application configuration category and its related items are provided below. In addition, a definition of each configuration value is provided.

### Organization Information

Configuration relevant to a specific organization is possible. This includes configuration options such as support /administrative email addresses or groups email accounts, mailing address, homepage links, the application title for search engines as well as the official and shortened name of the organization.

|  |  |
| --- | --- |
| Setting | Definition |
| app.name | Name of the system (e.g.: ePermitting, nFORM, etc.). |
| app.title | Title of the Application |
| app.headerImageLocation | Path to the header image used by the system. |
| app.home.url | URL target for the home button. |
| app.termsOfUseLink | Allows you to specify a page to serve as your terms and conditions and privacy policy. |
| app.footerClientAddressHtml | Physical address shown on the footer. |
| app.organization | Organization or Agency name. |
| app.organization.offical | Full name of the organization. |
| app.organization.site.url | URL target for the Home button. |
| app.organization.site.name | Name for the Home button |
| app.date.format | Format to use for date displays. |
| app.timeZone | The time zone basis if different than the default server time zone.  Options include:   * Hawaiian Standard Time * Pacific Standard Time * Mountain Standard Time * Central Standard Time * Eastern Standard Time   Arizona and Indiana have different time zones.  Time-zones are generic and will use localization to determine exact time.  The full list of valid values can be generated by TimeZoneInfo.GetSystemTimeZones(). |
| app.allowedNetworks | Limiting Access to certain networks can be set here by entering the IP address of the allowed networks. |
| app.excludeRobots | Enable search engine robot exclusion. |
| app.secondaryPayment.addressLabel | The label for a secondary payment address (e.g., “Overnight Mail”), if a secondary payment address is desired. If all app.secondaryPayment properties are empty, no secondary address will be displayed. |
| app.secondaryPayment.address | The mailing address for a secondary payment address, if a secondary payment address is desired. If all app.secondaryPayment properties are empty, no secondary address will be displayed. |

### Database Settings

The Database settings stored in the configuration files are essential to operation of the system and must be properly configured for the system to run. The two components are the connection string and the configuration value that designates the correct connection string to utilize.

|  |  |
| --- | --- |
| Setting | Definition |
| app.connectionstringkey | Reference to the name of the connection string to utilize, from the connectionStrings.config file where all available connection strings are found. |
| app.blankStringsToNulls | Determines whether the application will convert all blank strings to null prior to committing the data in the database. |

### Email and Message Settings

The nFORM system sends the majority of its messages via a SMTP mail server. In this configuration section the mail server address, any required values to send and receive emails, and the recipients of the internal emails can be specified.

By default, the base email templates will be utilized. If the intention is to create custom email content, the location of the organization specific email templates will need to be specified in the **email.templateOverridePath** variable. All system emails are set to use the same designated email address and sender name for the system. This server will need to be made accessible and configured for access prior to operation of the system, and the designated recipients of the internal emails will need to be established.

|  |  |
| --- | --- |
| Setting | Definition |
| email.isSendEnabled | Master switch" for email - turn it on or off by toggling this value. |
| email.support.address | Support Email Address – listed as the email contact on the site. This can be an individual or email group. |
| email.admin.address | All security and system emails will be sent to this address. |
| email.fromAddress | Address used to send notifications to users. |
| email.fromAddressDisplayName | Display name show on emails to users. |
| email.templateBasePath | Path to the base template files. |
| email.templateOverridePath | Path to the active set of email templates. |
| email.smtpServerHost | IP address or DNS of the SMTP server. |
| email.smtpServerPort | Port used by SMPT mail. Default is 25. |
| email.useSsl | Determines whether SSL is used to encrypt emails. Your SMTP server must support this feature. |
| email.smtpServerUsername | Username for sending email if required. Leave blank if it is not going to be used. |
| email.smtpServerPassword | Password for sending email if required. Leave blank if it is not going to be used. |
| email.genericName | Generic Name for addressing emails if user name is not available. |

In addition to specifying the settings used for identifying and authenticating sending messages with the email server, any emails sent out from by the system can be configured to have a custom email subject by overriding the default value.

### Configuring Workflow Actions

Workflow configuration can now include sending documents, changing status of a submission, or completion of a step in addition to sending out email for milestones. Each step can be set up to perform these actions based on a an event. Events include the completion of a step, the activation of a step (it becomes the next step in order to be completed), if its past its target date, or approaching the target date. There is also an event of “Payment Completed” which will activate if a payment is moved from “Due” to “Paid” or “Waived”.

### Application Defaults & Control Features

The system allows defaulting of some values in the system when forms are to be distributed to the regulated community in a specific location and it would be exceptional for it to be changed. Defaulted values are system wide, and can be overwritten by the person submitting the form when they are in the data entry step.

The values that can be defaulted include the state and country code for the contact control and the default latitude or longitude of the location control.

|  |  |
| --- | --- |
| Setting | Definition |
| location.latitude | Default Latitude for the starting point for the location control. |
| location.longitude | Default Longitude for the starting point for the location control. |
| contact.state | State can be pre-populated for the contact control. Leave blank if you do not want to utilize this feature. |
| contact.country | Country can be pre-populated for the contact control. Leave blank if you do not want to utilize this feature. |
| attachments.allowConfidential | Determines if you want to allow confidential attachments. |
| attachments.largeFileMessage | A message displayed with the attachment control, intended to represent the recommended maximum file size allowed. |

### Inline Help, Error Messages, Confirmation Messages

Inline help, error messages and confirmation message text is available on the pages of the application and can be modified by application administrators. This text is generally brief in nature and intended to instruct the user on the functionality of the area and interpreting a successful or failed action. These messages can also be customized by overriding the base values with values placed in the override files specific for the agency. These messages can be refined by adjusting the configuration values which are prefixed with “**message.**” or “**error.**”.

### Challenge Questions

As part of the electronic submission ability, users are required to provide answers to 5 security questions of their choice. The challenge questions are used to ensure identity at the time of submission for an electronic signatory certification required form.

The questions that are presented for the electronic signatory verification can be modified through the configuration values set for this group of questions. Additional questions can also be added by adding addition entries with the same naming convention, or they can be removed. These challenge questions can be refined and augmented by adjusting the configuration values which are prefixed with “**message.user.question*NN*.text***”*, where *NN* is the number of the question.

The challenge question system has been expanded to obtain questions from external systems. Use of an external system requires a base integration with the client security system. Once established, the viability of using said system for challenge questions can be investigated.

### Scheduled Jobs

nFORM has a series of administrative tasks that must be run. The following jobs will need to be scheduled for your system.

* Pending deletion of a draft submission notification
* Deletion of a draft submission
* Processing step approaching target date notifications
* Processing step past due notifications
* Rebuilding the search index for importing form data
* Population of staging data for exporting form data
* Autocomplete company list building

The system uses the CRON scheduler to manage all job schedules. Standard CRON scheduling syntax can be used to adjust these schedules.

The CRON format “SE MI HR DY MN DW”, where SE is the seconds (0-59), MI is the minutes (0-59), HR is the hour (0-23), DY is day (1-31), MN is the month (1-12), DW is the day of week (0-6). A star can be used to represent any possible value and a question mark can be used to represent any day of the week.

*Please note that after tasks that are used to index the site have been run, the application will need to be reset. Windsor recommends that you schedule the application pool to reset 15 – 30 minutes after the last scheduled job.*

|  |  |
| --- | --- |
| Setting | Definition |
| jobs.isEnabled | Global switch to enable/disable all jobs. |
| jobs.approachingTargetDate.schedule | Schedule of when to notify submission owners that their submission is APPROACHING the target completion date. |
| jobs.pastTargetDate.schedule | Schedule of when to notify submission owners that their submission is PAST the target completion date. |
| jobs.pendingDelete.schedule | Schedule of when to notify submission owners that their unchanged submission will soon be deleted. |
| jobs.deleteOldDrafts.schedule | Schedule of when to delete abandoned/unchanged draft submissions. |
| jobs.rebuildSearchIndex.schedule | Schedule of when to rebuild index of data that is used to populate new data templates. |
| jobs.exportControlValuesService.schedule | Schedule of when to export any new/updated submission control data to the export table. |
| jobs.buildCompanyLookup.shedule | Schedule of when to collect company information from the database to populate the autocomplete company name on registration. |

### Payment Processing

nFORM supports the use of 3rd Party Payment Processors to collect payments for form submissions, if desired.

Detailed Payment Processor setup is out of scope for this document, as they will be specific to the Agency’s needs. Please see the Integration Guide for further details.

|  |  |
| --- | --- |
| Setting | Definition |
| app.secondaryPayment.addressLabel | The label, if provided, for the secondary payment address. |
| app.secondaryPayment.address | The address, if provided, for the secondary payment address. |
| app.paymentInstruction.message | Displayed on the payment screen for both online and offline payments. |
| app.paymentVoucherInstruction.message | Payment instructions on the mail in voucher (offline only). |

### Document Management

The system can archive files on a Document Management server (e.g., S3 for cloud applications, File System of SQL FileStream for on premises and nVIRO document management systems for nVIRO solutions) that is accessible by the system itself. When Document Management Integration is active, files will be published at the time of submission to the Document Management server, and an email will be sent to the system administrator email address if there was problem publishing. This also allows the user to publish an existing form by enabling a button on the submission details screen.

Document Management integration is available with the system and can be used to archive submissions and their attachments in a document repository.

Please see the Integration Guide for further details.

### nTEGRATE

nTEGRATE can be used as a tool to facilitate an internet-based data integration.

Please see the Integration Guide for further details.

### User Defined Alternate Identifier Field

nFORM provides the ability to collect an unique identifier.

If you need to collect information about the submission for integration purposes with an existing system, you can define a field that is collected during the processing information step of the wizard that allows you to define the label of a simple text field that is 100 characters in length.

|  |  |
| --- | --- |
| Setting | Definition |
| app.submission.alternateidentifier.show | Determine whether an alternative identifier value is enabled. |
| app.submission.alternateidentifier.label | The label/name displayed with the alternative identifier field. |

### Notes

Notes are used by the internal users and the system to convey information to other internal users or submitters about a submission. Notes used to communicate with the application are considered external notes, while notes from the system or internal user generated notes are only viewable by users with internal access.

Note configuration properties allow you to define the label that appears on notes as well as the colors that define internal, external and completed notes.

|  |  |
| --- | --- |
| Setting | Definition |
| note.label.internal | Label for internal notes that the submitter cannot see. |
| note.label.external | Label for external notes that will be viewed by the submitter. |
| note.color.internal | Background color for internal notes. |
| note.color.external | Background color for external notes. |
| note.color.complete | The color of a note marked complete. |

## Other Configuration Options

### Internal Data

Internal Data is metadata about the submission that is stored outside of the complexity of the submission structure itself, and is used to provide easy access details about individual submissions properties. Internal data can be used to facilitate data integrations or document routing using data collected from or about the submission.

Internal data components can be created by form designers (in which case it will appear with all fields present when a submission is created) or after submission by a processor (in which case the field will only exist on the submission itself). The fields of data can be mapped to individual controls so that the data is captured directly from the submission, or the values can be designed to be entered by the processor of the form as milestones are passed. The fields can also be locked after they have been filled so that they cannot be changed.

### Database

The database is essential for all aspects of the nFORM system, and the system cannot run without a properly set up database. Prior to accessing the application for the first time, the database must be configured correctly with a designated database user that the application will use to retrieve and store system data. After the database is properly configured and running, you can then generate the database connection strings that identify the server instance and credentials necessary to connect to the database and place the text in the **[Root Directory]/Conf/Web/ConnectionString.xml** directory from the root of the application where all connection strings are stored. To determine which of the stored connection strings is used, you will designate it by name in the database configuration section below.

### Form Document Repository

Form controls can utilize any publicly available resource to provide a certain context or format for data to be returned in. For example, the table control requires that an excel spreadsheet is provided to download as part of a submission. Instruction controls can also contain links to files or instructions, lists etc.

Form controls that contain references to files such as a spreadsheet or graphic image, map, etc. do not store the files themselves and only use hyperlinking (e.g., the file is only referenced). While any internet accessible file is valid, if it is moved or renamed, the URL saved within the control will not work. While reference files can use any freely accessible internet URL, Windsor recommends creating a virtual directory for permit designers to store these files as a subdirectory of the application itself, and then mapping the location as a virtual drive for easy access.

# System Administrative Features

### Application Logging

The application provides a system for reviewing and storing errors generated by the system. When experiencing a problem, these can be very helpful to diagnose the problem that is appearing. The application log is maintained in the **[Root Directory]\App\_Data\Logs** folder. The current and active log file will be the only one without a date; all others are archives and can be deleted if required. The system will archive the current log file each day.

Important system events are also stored in the database in audit tables. For example, signature collection, security modifications, etc. are logged in both a file based audit log as well as in the database.

In addition to these error logs, nFORM also offers the ability to display a web based error logging details utilizing a system known as ELMAH. ELMAH error details are displayed when viewing the application from the server via localhost on the production site and are always enabled on the test server. The archive of logs can be accessed by going to the root URL of the web application and adding the following to the URL address, “/Elmah.axd”.

### Search Indexes

The system uses a text based search indexing tool named Lucene to manage the Application Finder tool and provide the search capabilities to import submission data. The system also uses text files to collect system data in the form of application log files. Administrators should become familiar with these files as they may need to be cleaned when moving a system, or reviewed in the event of problems with the application.

Log files and Search indexes can be found in the folder **[Root Directory]\App\_Data** Folder. If you are establishing a new environment or making modifications to the data through the database, it is recommended that you delete these files and allow the system to rebuild the indexes anew. Typical installation of nFORM will have the following folders used for storing the associated files:

* **Finder Index**: The form Finder utilizes Lucene index files to provide quick and intelligent search results. The folder housing these index files is named **[Root Directory]\App\_Data\FormItemSearchIndexes.**
* **Data Import Search**: This is related to the job named Search Index referred to in the Jobs configuration section in this document.The system allows users to search for and import information from other systems into their user profiles for use in prepopulating their forms. The searching function which supports this function utilizes Lucene index files to provide quick and intelligent search results. The folder housing these index files is named are in the folder named **[Root Directory]\App\_Data\SearchIndexes.**

### Identification of Administrative User Accounts

Users of the system can be considered an Administrative user. These Administrative users have access to perform any task for any organization in the system. A user can also be assigned to other roles but administrator status will override all other roles. Only existing Administrative users can promote another user to be an Administrator. To add a new Administrator, select the user you want to promote from the user list, and click the administrator checkbox to add the new user.

### Help/User Guides

The system provides page level help to support the user experience and provides steps to perform most actions in the system. The help is composed of two sets of help information; administrative help and public help. Administrative help is designed for the users with internal roles for administering information such as organization content, form content, or processing form submissions. Public help is designed for external users who use the system to submit and monitor their own form submissions. Within the system each web page is considered “Public” or “Administration” based on the functions that the page performs and the roles of the user logged in. Each help page can consist of several related topics. A hard copy of the help is available as the Public or Administrative Guides.

Additionally, two system User Guides are available with similar content. One is focused on topics relevant for “Public” users while the other focuses on topics most relevant to “Administrative” users.

See the nFORM Online Help/User Guides Resources topic in the Windsor Wiki for more details.

# Accessibility

In designing and developing the nFORM tool, Windsor has worked to establish a system which follows industry coding, usability and ADA (Americans with Disabilities Act of 1990) compliance standards in accordance with the WC3 Accessibility Guidelines, Section 508 Standards and WCAG (Web Content Accessibility Guidelines). The system has been tested and verified as WCAG 2.0 Level A compliant.

# Frequently Asked Questions

**How do we migrate this system or recreate it on another server?**

Please refer to the Implementation Steps section of this document for the steps for creating a new environment.

**Where are the system files located?**

The files can be found in the root directory of the application and can be located from within IIS by clicking on the application and selecting explore.

**What is the minimum Window Server Roles and features needing to be installed for this system?**

The default IIS User account can be used for the web portion of the application. Write access to **[Root Directory]/App\_Data** is required; however, this is standard in ASP.NET environments. For more information on App\_Data, see: <http://msdn.microsoft.com/en-us/library/06t2w7da%28v=vs.80%29.aspx>.

**What are the key settings in the vars file?**

Please refer to the "System Configuration" section of this document.

**Where is the connection string?**

This is stored in the file connectionstrings.xml file located in the **[Root Directory]/conf/Web/** directory.

**Are there other software applications that need to be installed to support this application?**

No, the nFORM application is a standalone system which includes all necessary libraries and files.

**If this application exchanges data with other systems, how can this be accomplished?**

External interfaces include PayPal, Sharepoint and data export/import services. Additional communication occurs with the application database. The configuration is described in the System Configuration area is additional detail.

All communication with PayPal is over HTTPS port 443 and communication. PayPal returns users to the application, at which point a single outbound port 443 request is issued to gather more details on the transaction.

The system has the ability to publish documents to Sharepoint and this is largely a one-way communication initiated from the application. Sharepoint utilizes HTTP Port 80. Some pull communication is performed, to simply test for existence of the library.

Data can also be imported and exported from other auxiliary systems. Please refer the Data Integration Document for further details on this process.

**Does the system store data outside of the database, if so where and how?**

Data Stored Outside of nFORM:

* Integrated System:
  + 3rd Party Payment Processor Information: Online payments may be facilitated via a 3rd Party Payment Processor (e.g., PayPal, Authorize.Net, etc.).
  + Document Management System Documents and Metadata: Submitted forms may be published to a 3rd Party Document Management system (e.g., Sharepoint).
  + Systems receiving nFORM data: Other agency system may receive data submitted on a form via custom integration process.
  + RoundHouse database configuration information: Database versioning information will be stored in RoundHouse to facilitate the deployment process.
  + Application/Server Log Files: Log files are generated and stored in APP\_DATA/Logs. Full text search indexes are stored in several folders inside **[Root Directory]/App\_Data**, but these indexes can be deleted and the application will recreate the data on the next start.
  + Unstructured documents: If FileStreaming is enabled, unstructured documents will be stored on the Database Server file system.
  + Static Documents: Internal users may store documents in a static resource directory, for use as a reference on Forms. Please refer to **Form Document Repository** section for additional details.

All other data is stored in to the system database

**How/where is table control spreadsheet template location – static file - managed?**

The template location is defined as a hyperlink within the table control. Any valid URL can be linked to retrieve the template. To use the static file location please refer to **Form Document Repository** section of this document.

**Where/how to change the email address for electronic signatory requests?**

The email that receives any electronic signatory requests is defined with the "smtp.system.email" configuration key.

**What changes require the application to be refreshed/restarted?**

Any changes to the configuration or the database will require the application pool to be reset.

**How can I add additional lookup lists or maintain existing lookup lists for use in the form lookup controls?**

These values are maintained in the LOOKUP table using name value pairs in a JSON format. Note: any changes to this table will require the application pool to be refreshed to take effect.

# Appendices

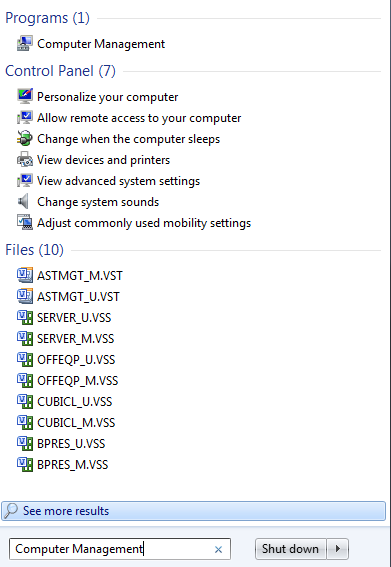
## Appendix A - Application Pool Configuration

nFORM uses data caching to allow speedy access to large objects. It is therefore essential to coordinate the timing of external services such that the correct indexing is produced for the system and that the cache is not reset automatically when idle.

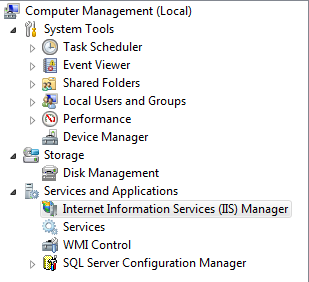
By default, the application pool recycles automatically based on a set time interval from the prior cache (essentially using a countdown timer). The application pool also performs a memory dump after 15 minutes (default setting) of inactivity, which forces the next user to wait while the cache is rebuilt.

These defaults can be modified to allow the application pool reset and / or the memory dump to occur at a specific time. The process for setting a default application recycle time and setting the idle recycling time is detailed below.

### Accessing the IIS Manager panel in Windows 7



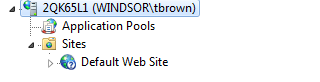
1. Use the Search programs and files option by clicking the start icon. Search for “Computer Management”.
2. Select the “Computer Management” selection that appears under programs.
3. This will open your computer management console.
4. Expand the option “Services and Applications” and select Internet Information Services (IIS) Manager.



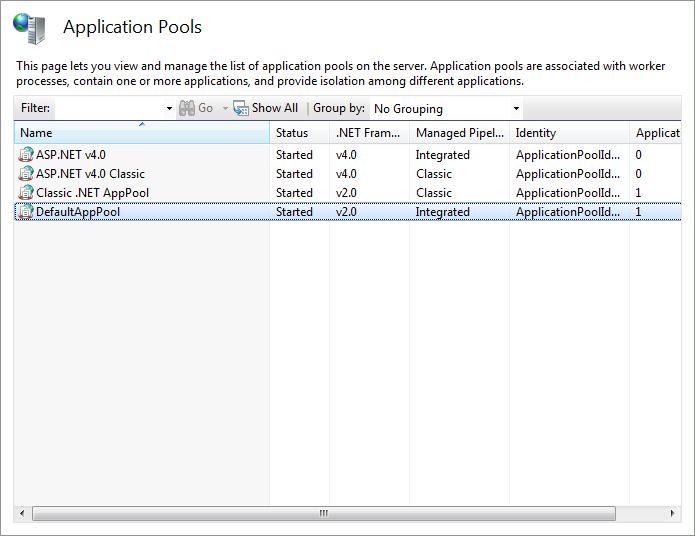
1. The IIS Manager Console will appear in the window.

### Setting the Application Pool Recycling Settings

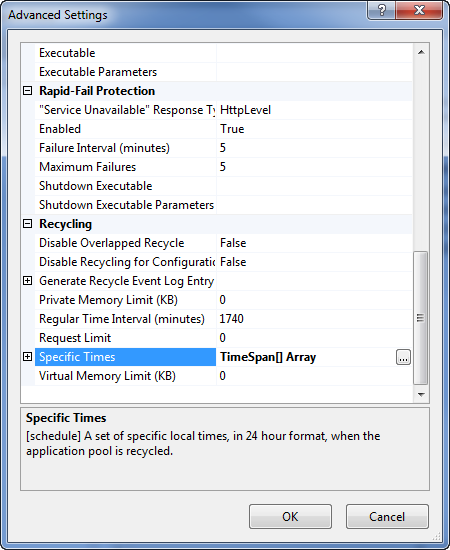
1. In the IIS Manager Console, expand the server that contains the application and select Application Pools.



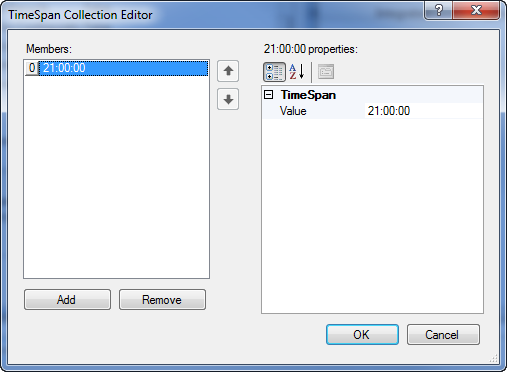
1. The list of application pools currently running on the system will be displayed. Find the application pool for the nFORM system.



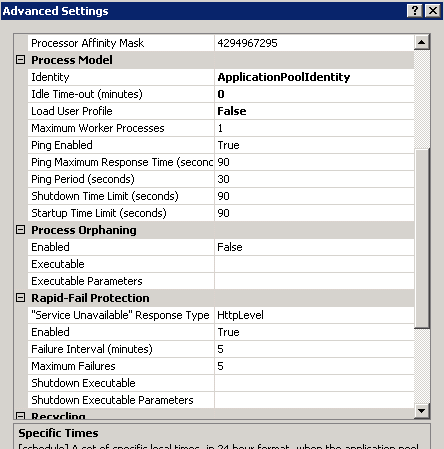
1. Right click on the application pool and click the Advanced Settings option.
2. This will bring up the Advanced Settings dialog, scroll to the bottom to the section labeled “Recycling”. Locate the entry named Specific Times and click the “+” button to extend the Specific Times properties.



1. Click the More (“…”) icon next to the Specific Times value to open the Time-Span[] Collection Editor dialog.



1. Click the “Add” button, and then type in the specific time value (i.e., in military time format) representing when you want the application pool to recycle. For example, “21:00:00”.
2. Click the “OK” button to close the dialog.
3. Scroll up to the Process Model heading in the Advanced Settings dialog.
4. Set the setting Idle Time-out (minutes) value to “0”.



1. Close the settings dialog.
2. Close the Management Console.

1. Windsor is not responsible for a client’s configuration files once deployed. Modifications to configuration files by client staff may result in application instability. [↑](#footnote-ref-1)