OpenSpan Server 5.2

Administration and User Guide

April 17, 2012

This manual focuses on use of OpenSpan Studio 5.2 and OpenSpan Runtime 5.2 in conjunction with OpenSpan Server 5.2 in a connected architecture.

Developers should see help.openspan.com for reference information on creating solutions with OpenSpan Studio. Select **Help for OpenSpan 5.2 Users** from the drop-down list box located with the Contents - Index - Search menu.

OpenSpan Inc.
Suite 140
4501 North Point Parkway
Alpharetta, Georgia 30022

Phone (International) +1 (678) 527-5400
Phone (US and Canada) (877) 733-1136
Sales: +1 (678) 527-5455
Phone, United Kingdom (EMEA): +44 (0) 207 043 3573
Email: sales@openspan.com

CONTENTS

Chapter 1 OPENSPAN SERVER 5.2 OVERVIEW

In this Chapter

Welcome to the Overview chapter of the *OpenSpan Server 5.2 Administration and User Guide*. This chapter:

- Describes the relationship between OpenSpan Studio 5.2, OpenSpan Server 5.2, and OpenSpan Runtime 5.2.
- Shows how OpenSpan Server 5.2 can be deployed and scaled to provide enterprise-wide efficiency in administering desktop runtime packages and processing Events information coming to the server from user desktops.
- Defines broad categories of people who use the three products, and provides a guided set of references to help find the parts of this manual that will be of the most initial use. You may want to read through all of the material, because like many enterprise software suites, it's useful to understand how features and functions relate to each other.

General

There are three main application components in a full OpenSpan 5.2 architecture: OpenSpan Studio, OpenSpan Server, and OpenSpan Runtime.

- **OpenSpan Studio** is the development environment, based on Microsoft Visual Studio, for developing, testing, and configuring OpenSpan solutions in the form of Runtime desktop packages. These packages can be deployed directly to user desktops along with manifests the keep versions current. When used with OpenSpan Server 5.2, OpenSpan Studio can upload these runtime solution packages directly to OpenSpan Server for modification and download to the user desktop. Additionally, OpenSpan Studio is used to enable runtime solutions to collect and transmit certain types of events information on specific processes as determined by business requirement and data parameters.

  Note: A wide assortment of core and elective training is available for OpenSpan Studio. Classes are both instructor led and non-resident, and a full library of lessons, solutions, and support materials is available. Contact OpenSpan Support for more information.

  OpenSpan publishes version-specific online help for OpenSpan Studio via the Internet. It is available from help.openspan.com.
- **OpenSpan Server** is the middle link between OpenSpan Studio, where solutions are developed, and OpenSpan Runtime, which executes packages on the user desktop. OpenSpan Server receives runtime solution packages directly from OpenSpan Studio, and can further modify them codelessly to assign default values and tool tips to the interrogated application controls in the solutions. OpenSpan Server performs package management to deliver both packages and modification feature sets to each desktop based on the user's login and administrative group assignment. OpenSpan Server also installs services and queues that collect application and process events from user desktops, writes them to the OS_Events database, and has a report package for displaying basic visualization of the data. The Events architecture is scalable and can be deployed in different configurations. Several diagrams follow in this chapter that illustrate some common solutions.
- **OpenSpan Runtime** is a small, licensed application that runs on the end-user's workstation. It executes OpenSpan solution runtime packages. The runtime packages can come from OpenSpan Server or any other

deployment method. Using OpenSpan Runtime in conjunction with OpenSpan Server has a number of advantages.

- First, OpenSpan Server sends a runtime solution package to each desktop based on the user's login credentials. Different users can access the same workstation at different times but receive different solution packages based on job tasks, role, or similar criteria.
- Second, OpenSpan Runtime communicates back to OpenSpan server with configurable events data, reporting on application and process metrics from the user desktop. The events data is useful to determine time spent in applications, errors, application churn, and process adherence, to name a few. OpenSpan Runtime can be enabled to send high-level events data even if no solution package is downloaded, and can run "silently" as a background process so as to not interfere with user performance.
- Additionally, each instance of OpenSpan Runtime can display (if enabled) a modal Process Toolbar that desktop users can use to select particular predefined processes, collect events data, and send that data to OpenSpan Server to help measure business process efficiency.

OpenSpan Server 5.2 Architecture

General

In OpenSpan Server 5.1, event collection logic was consolidated with the Site, User Management, and STS (authentication) services. However, for release 5.2, Event collection is a separate logic piece, because it is where OpenSpan Server usually incurs the highest bandwidth. Splitting the OpenSpan Server 5.2 Events architecture into components means that there are now a number of alternatives for scaling and load balancing.

OpenSpan Server 5.2 Events Features

In OpenSpan Server 5.2, the Event Collection service has been separated from the OSS service, which in past releases controlled event input and transfer to the database tables. The Site, OSS, User Management, and STS logic stay together and control the application server, but the event collection and processing can be deployed independently.

Event Collection Logic

The OpenSpan Server 5.2 event collection logic consists of:

- An Event Collection Web service
- An events queue running under MSMQ
- A Windows service called the Event Processor

Events Flow

1. The OpenSpan runtime client sends an event to the **Event Collection Web service**.
2. The Event Collection service does no processing or metadata validation. It simply puts the event in a **queue** and returns success or failure.
3. The **Event Processor**, a Windows service, has the mission of reading Events off the **queue**, validating the meta-data (matching the dimensions of incoming Events with dimensions defined in the database), and writes the event to the **Events tables** in the **OS_Events** database.

Two Databases: OS_Server and OS_Events

Additionally, the OS_Server and OS_Events databases have been separated. To eliminate read/write collisions within a single server database, in release 5.2 the raw event tables are now written to their own database. At specified intervals, events are replicated from the OS_Events database to the OS_Server database, where they are aggregated. This means that the two databases can be put on separate machines, eliminating contention between events processing and application overhead.

Deployment Examples

Basic

The illustration below shows an example of an OpenSpan 5.2 server. This architecture normally supports around 50 runtime users or less. The events processing services and message queue(s) are deployed on the same machine. This is the most fundamental installation variant. In this diagram:

- Client applications, OpenSpan Studio, and Runtime packages on user desktops interact with the Application and User Management services supported by the OS_Server database.
- Runtime packages send events data from user desktops to the Events Collection service, which puts the events in the message queue. The Event Processor picks the Events up from the message queue and sends them to the OS_Events database.

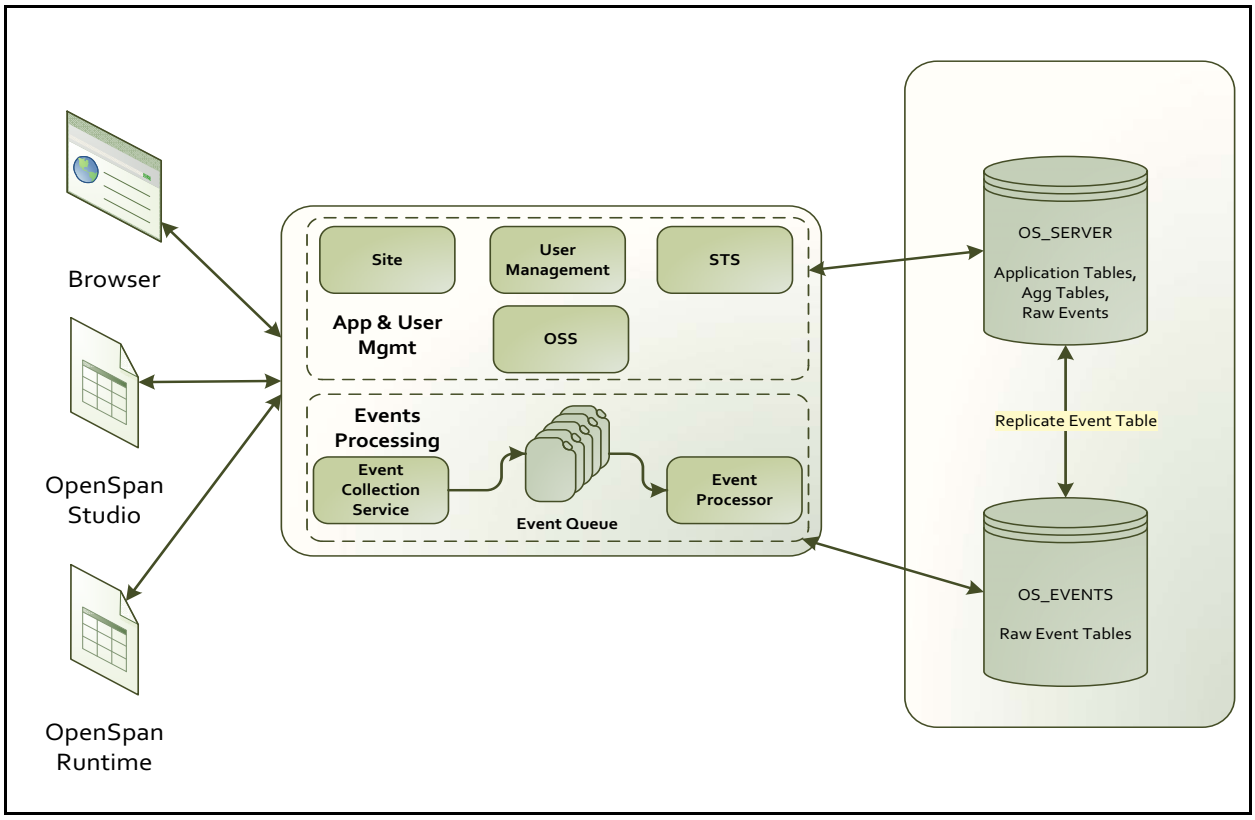


FIGURE 1. OpenSpan Server 5.2 Basic Architecture

Scaling Options

OpenSpan Server 5.2 allows you to increase event processing capacity in different ways:

- Run Parallel Event Service Logic
- Add Event Processor Services
- Add OS_Events Databases
- Deploy hybrids of these elements

Run Parallel Event Service Logic

A simple way to increase events throughput is to install multiples of the event processing logic chain.

- A load balancer can be deployed as a “traffic cop” to optimize flow to the Event Collection services.
- The Event Processor services can be configured to write to a single (shown here) or separate OS_Events databases.

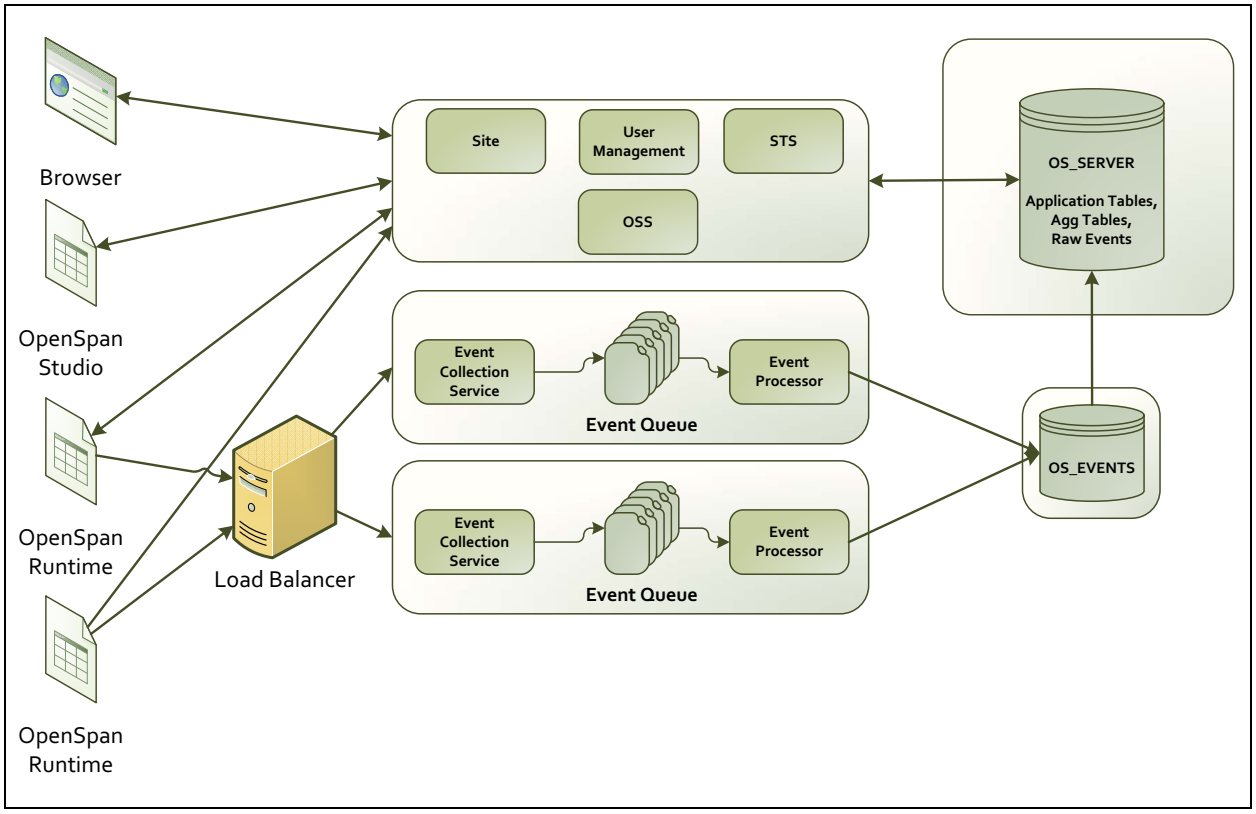


FIGURE 2. Add Event Services Chains

Add Event Processor Services

Typically the Event Processor Windows service incurs a heavier processing load than the Event Collection Web service and queue. Deploying Event Processors in a higher ratio to the number of Event Collection services is an option to distribute the work of validating metadata and moving events to the OS_Events database.

- Deploy one Event Processor for approximately 2000 runtime instances.
- Consider hosting the Event Processors on a server with multiple processor cores; more threads will be available and make the events flow even more efficient.

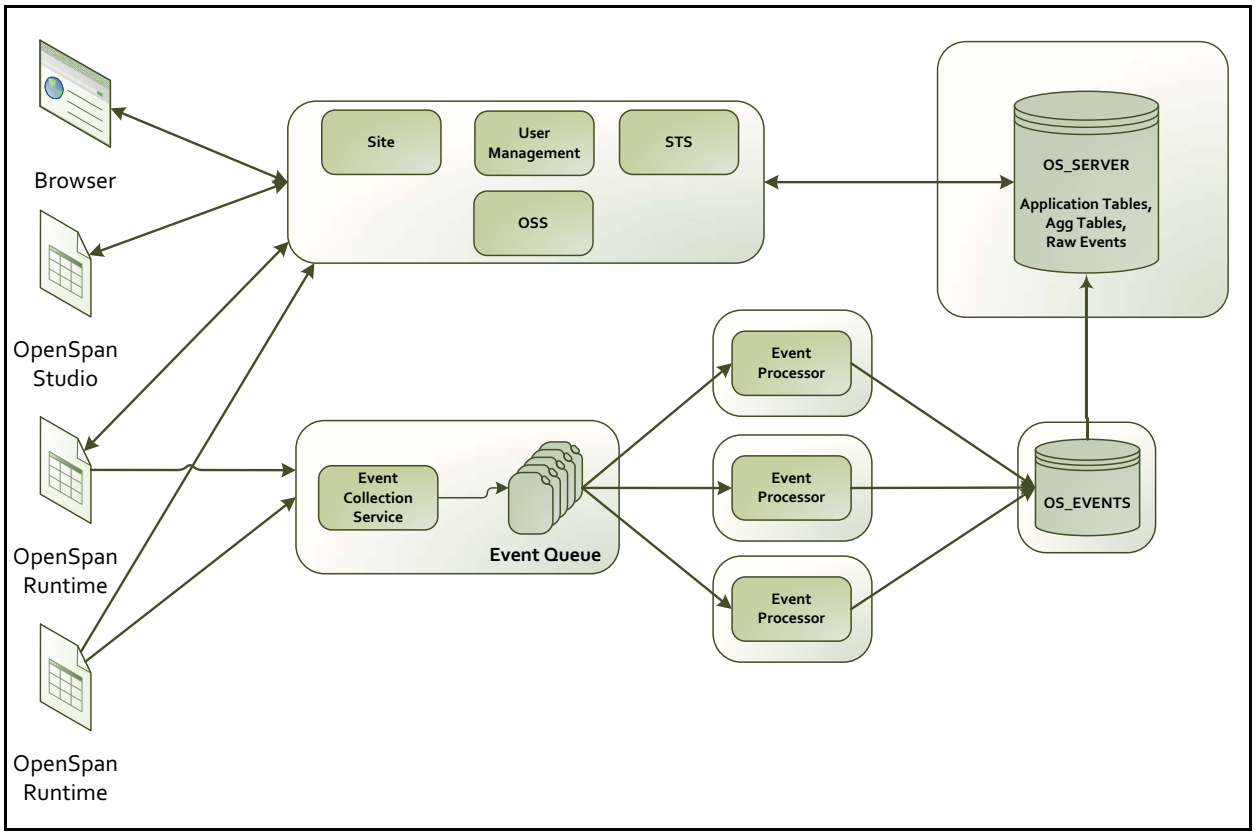


FIGURE 3. Add Event Processor Windows Services

Add OS_Events Databases

This diagram is the same as Figure 3, “Add Event Processor Windows Services,” on page 1-6 except that it shows another option: additional OS_Events database instances. These are commonly located on separate machines. In some situations, such as widely dispersed geography (an increasing consideration for global enterprises), dedicated databases for each enterprise location are highly recommended. Other requirements driving multiple OS_Events databases include segmentation within organizations, security, etc.

Note: Best practices show that it’s important to physically locate the Event Processors close to their associated databases so that network latency or other infrastructure factors don’t create issues with throughput.

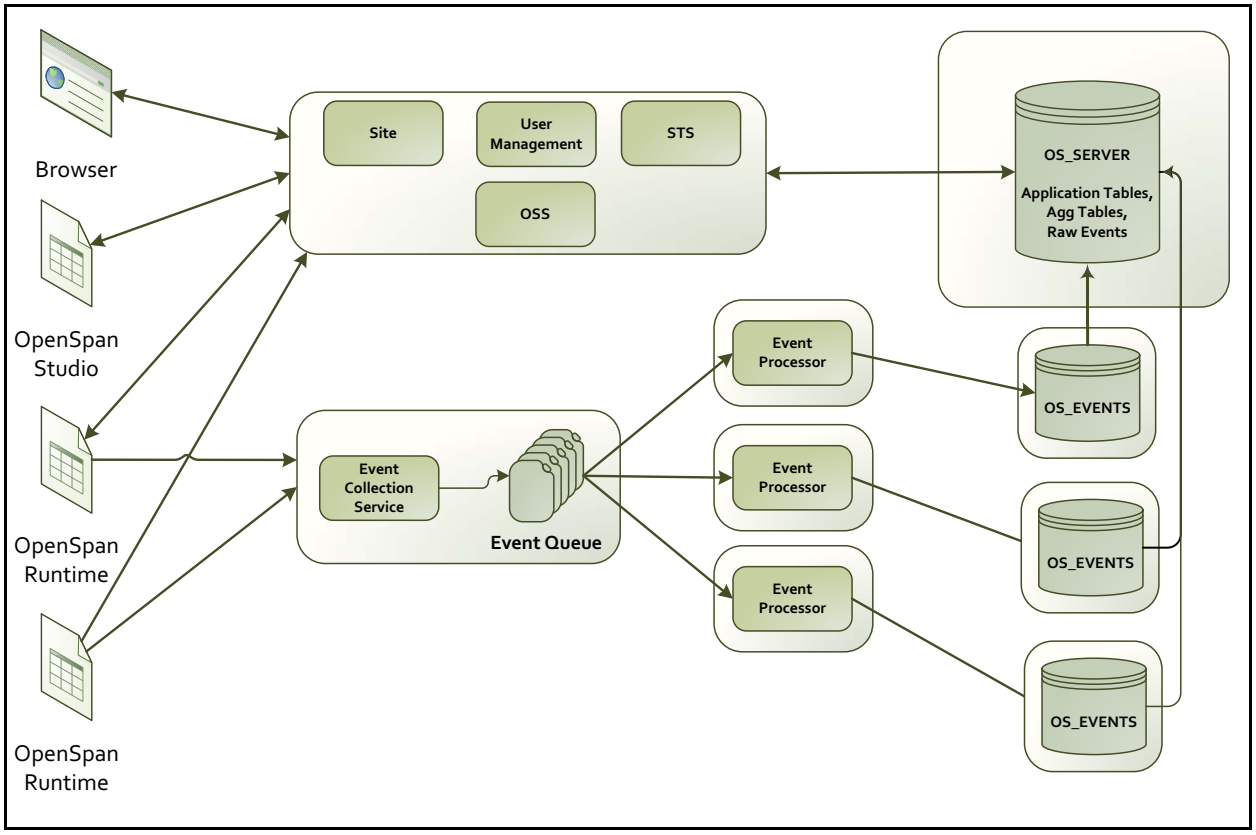


FIGURE 4. Add Event Databases

Hybrid Combinations

This final diagram shows a hypothetical combination of all the solutions offered previously. Use of Internet Information Services (IIS) and Windows Communication Foundation (WCF)-based components means that scaling can be done consistently and reliably.

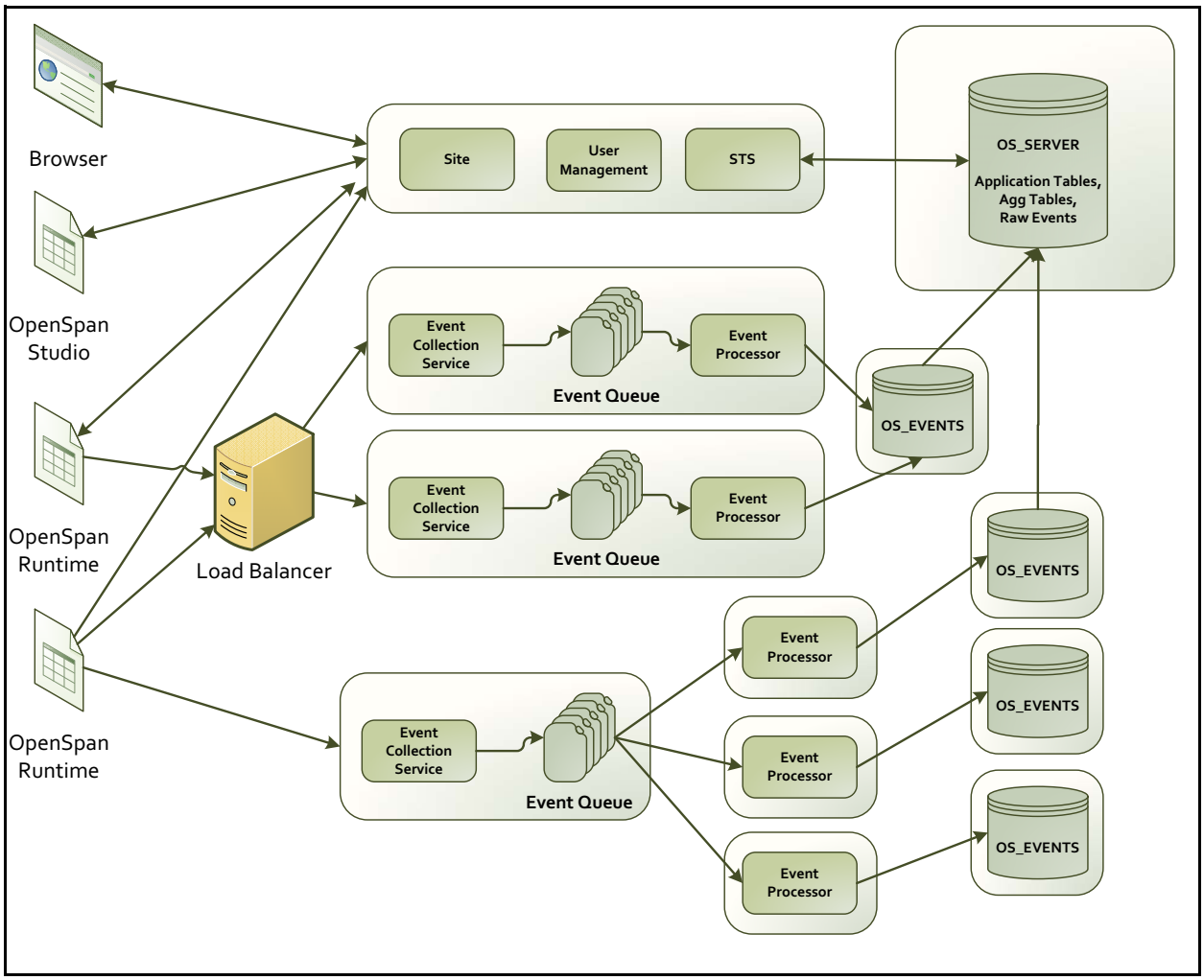


FIGURE 5. Combined Solution

Deployment and Scaling Summary

A large organization may have OpenSpan Runtime solutions deployed on thousands of desktops, active at the same time, sending a steady stream of events data. The components of OpenSpan Server 5.2 allow a number of ways to flexibly accommodate different volumes of traffic, including:

- Running parallel Event Collection services, message queues, and Event Processors, with load balancers managing input to the Event Collection services.
- Adding Event Processor services instances to validate event metadata and write multiple streams to the OS_Events database.
- Adding multiple OS_Events database instances.
- Combining these tactics into a cohesive optimization strategy dependent on the needs of the enterprise. OpenSpan Professional Services engineers are experienced in assessing customer infrastructure and business needs and making recommendations for optimal events processing.

Where Do I Start?

Here are some recommended paths through the manual, based on common categories of OpenSpan users. When reading along, it's useful to have access to the referenced products: OpenSpan Studio, Server, or Runtime. Having access to an environment with test data or other "sandbox" arrangements will also reinforce or make clear ways to take the basic facts and instructions in this manual and the OpenSpan online help and fit them to your organization's needs and tasks.

TABLE 1. Finding Information Based on Your Role

If you are...	Then check out...	And pay particular attention to...
An **Administrator** installing and configuring OpenSpan Server	**OpenSpan Server 5.2 Installation and Structure** on 2-1	**All parts**
	Site Settings on 4-44	
A **Software Developer** of solutions using OpenSpan Studio	**Configuring OpenSpan Studio for OpenSpan Server** on 3-2	
	Matching Configuration and Promotion Levels to Runtime Packages for Debugging on 3-6	
	Producing Process Automations on 3-9	
	OpenSpan Runtime 5.2 Configuration on 5-1	**RuntimeConfig.xml** on 5-2
A **Business Analyst** managing process improvements	**Groups and Users** on 4-6	• **Adding Groups** on 4-14 • **Adding Users** on 4-7
	Managing Deployment Packages on 4-20	• **Adding, Editing, Deleting Features** on 4-21 • **Assigning Feature Sets to Groups** on 4-29 • **Assigning Configurations to Groups** on 4-31 • **Assigning Feature Set Levels** on 4-33 • **Assigning Configurations to Groups** on 4-31 • **Assigning Groups to Packages** on 4-34
	Producing Process Automations on 3-9	
	Managing Group Runtime Settings on 4-39	
	OpenSpan Runtime 5.2 Configuration on 5-1	All parts
	Chapter 6, *OpenSpan Server 5.2 Reports and Graphs*	All parts

TABLE 1. Finding Information Based on Your Role (Continued)

If you are...	Then check out...	And pay particular attention to...
A **Runtime user** of desktop OpenSpan deployment packages	**Runtime Menu** on 5-4; **Process Toolbar** on 5-7.	

Chapter 2 OPENSPAN SERVER 5.2 INSTALLATION AND STRUCTURE

In this Chapter

Welcome to the Installation and Configuration Chapter of the OpenSpan Server 5.2 Administration and User Guide. After reading, you will be familiar with these topics:

Foundations

Each enterprise that implements OpenSpan projects, including OpenSpan Server, has a different set of requirements and host infrastructure. Successful outcomes have shown that it is essential that specialists from OpenSpan's Services organization partner with customer IT staffs to negotiate the myriad of server requirements, configuration options, and special requirements such as security.

Recommended System Configuration

OpenSpan Server is deployed with an Application Server component as well as a Database Server component.

Notes:

- GB-capacity network between application servers and database servers is recommended.
- We recommend deploying a Load Balancer in front of the Application Servers and deploying an Application Server for every 1000 users, with a minimum of 2 Application Servers.

Application Server — Minimum

TABLE 1. Minimum Application Server Requirements

Component	Details
Operating System	• Windows Server 2008 • Microsoft .NET Framework 3.5 SP1 • Microsoft Windows Installer version 3.0 or later • MSMQ feature installed
Processor	• 1 Quad-core 3+ GHz CPU
RAM	• 6 GB RAM or higher
Storage	• 2 LUNs (SCSI or SSD) Operating System MSMQ — 100 GB SSD
Networking	• GB NIC with TOE (TCP/IP Offload Engine)

Application Server — Recommended

TABLE 2. Recommended Application Server Requirements (1 of 2)

Component	Details
Operating System	• Windows Server 2008 R2 • Microsoft .NET Framework 3.5 SP1 • Microsoft Windows Installer version 3.0 or later • MSMQ feature installed
Processor	• 2 Quad-core 3+ GHz CPUs
RAM	• 16 GB RAM or higher

TABLE 2. Recommended Application Server Requirements (2 of 2)

Component	Details
Storage	• 2 LUNs (SCSI or SSD) Operating System MSMQ — 100 GB SSD
Networking	• GB NIC with TOE (TCP/IP Offload Engine)

Database Server — Minimum

TABLE 3. Minimum Database Server Requirements

Component	Details
Operating System	• Windows Server 2008 • Microsoft .NET Framework 3.5 SP1 • Microsoft Windows Installer version 3.0 or later • Microsoft SQL Server 2008 Enterprise Edition or higher
Processor	• 2 Quad-core 3+ GHz CPUs
RAM	• 16 GB RAM or higher
Storage	Database size varies based on number of users, events monitored and retention period. • 4 LUNs (SCSI or SSD) Operating System OpenSpan Server Database — 1 TB (if on SAN, this should be zoned) SQL Transaction Log Files — 256 GB SSD SQL Backup
Networking	• GB NIC with TOE (TCP/IP Offload Engine)
Miscellaneous	• No scheduled jobs during normal business hours other than OpenSpan specified • Virus protection On-Access scans disabled for MDF, LDF and NDF files

Database Server — Recommended

TABLE 4. Recommended Database Server Requirements

Component	Details
Operating System	• Windows Server 2008 R2 • Microsoft .NET Framework 3.5 SP1 • Microsoft Windows Installer version 3.0 or later • Microsoft SQL Server 2008 Enterprise Edition or higher
Processor	• 4 Quad-core 3+ GHz CPUs
RAM	• 32 GB RAM or higher
Storage	Database size varies based on number of users, events monitored and retention period. • 6 LUNs (SCSI or SSD) Operating System OpenSpan Server Database — 1 TB (if on SAN this should be zoned) SQL Transaction Log Files — 256 GB SSD SQL Backup TempDB - 100 GB SSD TempDB Transaction Log Files — 25 GB SSD
Networking	• GB NIC with TOE (TCP/IP Offload Engine)
Miscellaneous	• No scheduled jobs during normal business hours other than OpenSpan specified • Virus protection On-Access scans disabled for MDF, LDF and NDF files

OpenSpan Server 5.2 Installation Concepts

OpenSpan Server 5.2 runs under Microsoft Internet Information Services (IIS). Knowledge of IIS is key for understanding of how OpenSpan Server 5.2 functions.

In general, the OpenSpan Server 5.2 installation process automatically configures IIS for proper operation, and then creates necessary directories and support files.

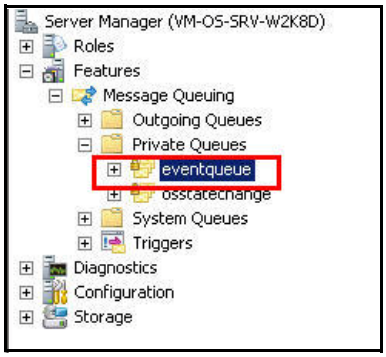
The following items are configured or installed in the following sequence. Configuration will vary according to requirements. OpenSpan Services representatives will execute the configuration and enablement.

1. Event Message Queue (below). Create in MSMQ.
2. OpenSpan Server (includes database creation scripts) (page 2-6). Get and run the installer.
3. Event Collection Service (page 2-7). Get and run the installer.
4. Event Processor (page 2-7). Get and run the installer.

Event Message Queue

The host server must be configured for Microsoft Message Queuing (MSMQ). Then create a private queue, called eventqueue. Ensure that the proper permissions are set, as needed, for access by:

- System
- Network Service
- IIS_IUSRS
- Others as required



Note: The EventPluginQueue, osstatechange, is created by the Event Collection service (page 2-7) installer.

OpenSpan Server

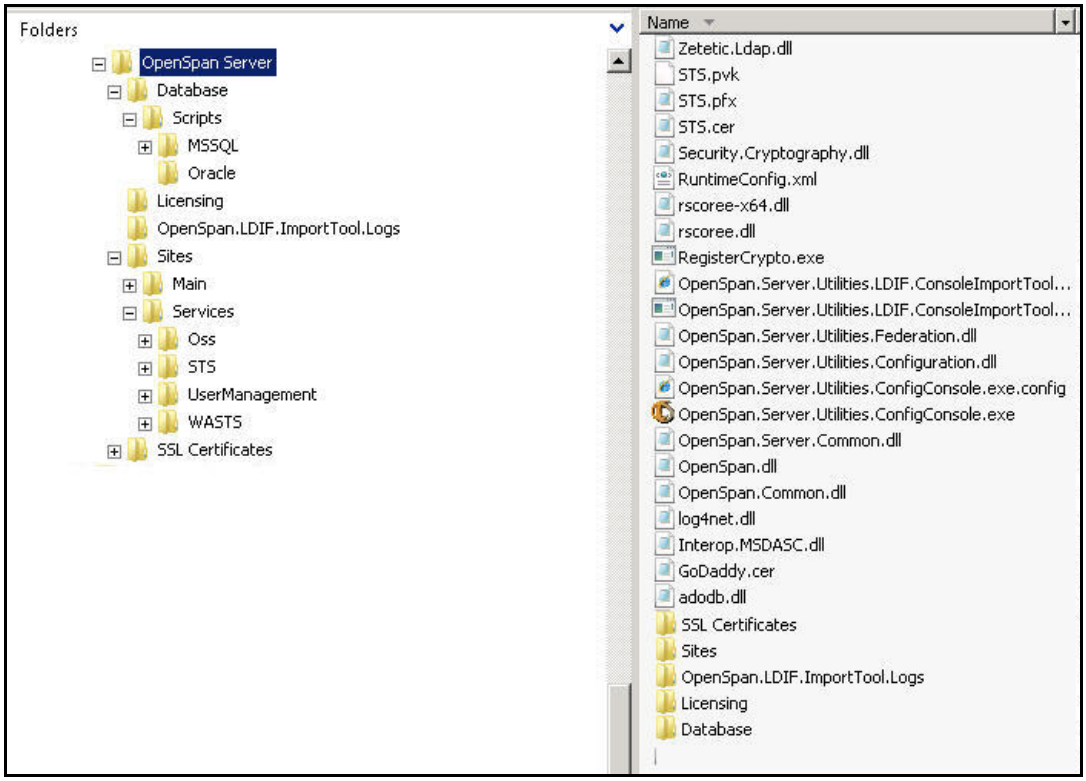
OpenSpan Server uses a single installer for single-site deployment of “app server” functions. The default installation directory is:

```
C:\Program Files\OpenSpan\OpenSpan Server
```

Note — Database Creation: After installation of OpenSpan Server 5.2, run the appropriate installation scripts. The scripts needed for OS_Events or OS_Server database creation or upgrades are installed in:

```
C:\Program Files\OpenSpan\OpenSpan Server\Database\Scripts\MSSQL\
```

This illustration shows a typical set of OpenSpan Server 5.2 directories and their contents.

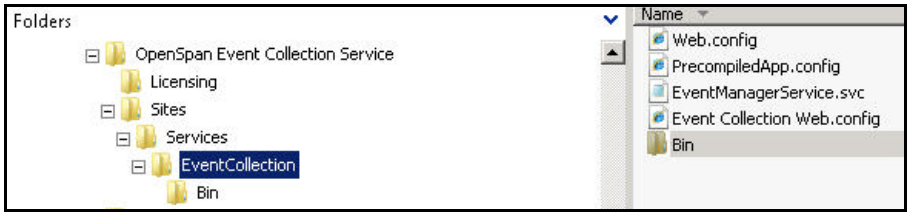


Based on requirements, some directories may be deployed elsewhere, in which case it’s necessary to use the configuration console to reference them. For more information, see “OpenSpan Server Configuration Console” on page 2-10. The chief directories are:

- OpenSpan Server — Root directory; configuration console, RuntimeConfig.xml, related DLLs
- Database — Contains required scripts for report generation and administrative functions
- Sites — The default installs the Main site, containing business logic and presentation layer support, as well as STS metadata
- Services — Required services for the server itself, STS, Windows Authentication STS, and User Management
- SSL Certificates — Provided as required for installation on the host server later to ensure SSL connections

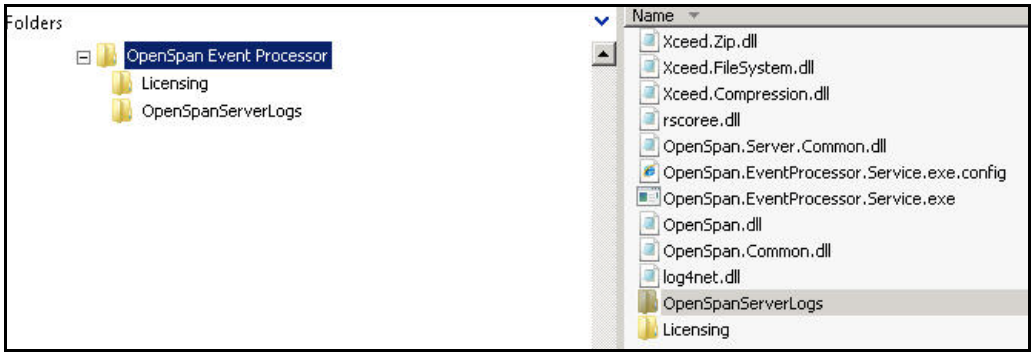
Event Collection Service

The Events Collection Web service runs under IIS. The service receives raw event data and puts it in the Events message queue. It creates its own application pool called OpenSpanEventCollectionAppPool.



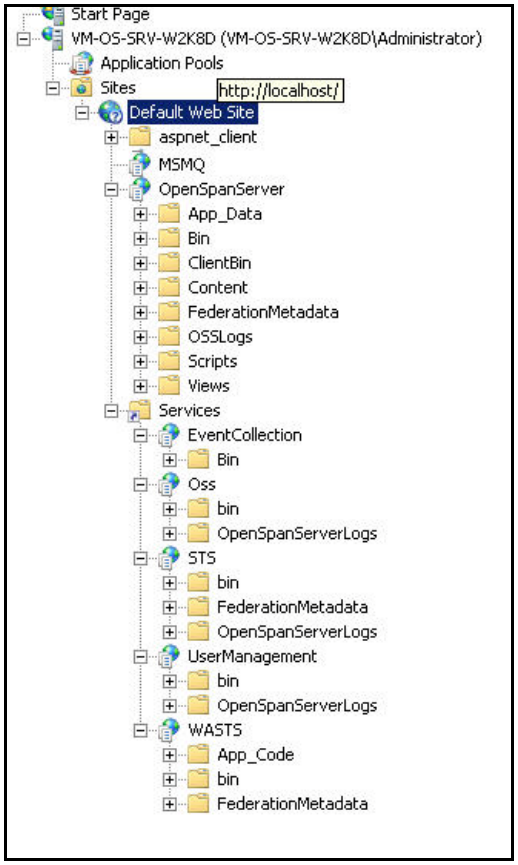
Event Processor

The Event Processor Windows service routes events from the message queue to the raw events tables in the OS_Events database.



IIS Rollup

When fully installed, the host server's IIS directory looks similar to the one below. Note that the Event Collection Web service (and the Event Processing Windows service) can be located on other machines to distribute Events data to separate OS_Events databases.



OpenSpan Server Diagnostics

The OpenSpan Server provides two basic means of obtaining diagnostic information:

Log4Net Logs

The Log4Net logs are a logging facility built into the OSS code similar to how the main OpenSpan diagnostics work, where both exceptions and key operational activities are logged to a file. Each of the four OSS sites (main, services/oss, services/sts, services/usermanagement) provide this type of logging. The logging can be controlled by editing the web.config file for a given site and modifying the information in the <log4net> section of the document. The factory shipped configuration for each site is that the logging will be done to a local folder to the site called "OpenSpanServerLogs\somefilename.txt" and will use an appending scheme where the file is appended to until it reaches 10MB in size. After that occurs the file will be renamed to include a sequence number on the end of the log file name. A maximum of 10 files is maintained, after which the 11th file will be deleted to conserve diskspace. These types of logs are in general very useful for exception tracing.

.NET Tracing

The other type of diagnostic information can be obtained via the user of the built-in .NET diagnostic tracing. .NET provides the ability to use configuration based diagnostics through the definition of listeners and sources in the <system.diagnostics> section of the web.config files. The factory-configured web.configs for the OSS sites have these diagnostics presently enabled however this will likely be revised to be disabled as built-in .NET diagnostics do not provide the ability to do log truncation or rolling logs without implementing your own logger assembly/class, which is not presently implemented. There are two types of information logged:

- Windows Identity Foundation activity (WIF) for security tracing. This is only enabled for the services/STS site. The source for this information is Microsoft.IdentityModel.
- WCF activity for service call diagnostics. This is enabled on all sites. The source for this information is System.ServiceModel.

To disable diagnostic logging, the <system.diagnostics> section of web.config can be commented by enclosing the <system.diagnostics> ... </system.diagnostics> section in XML comments.

OpenSpan Server Configuration Console

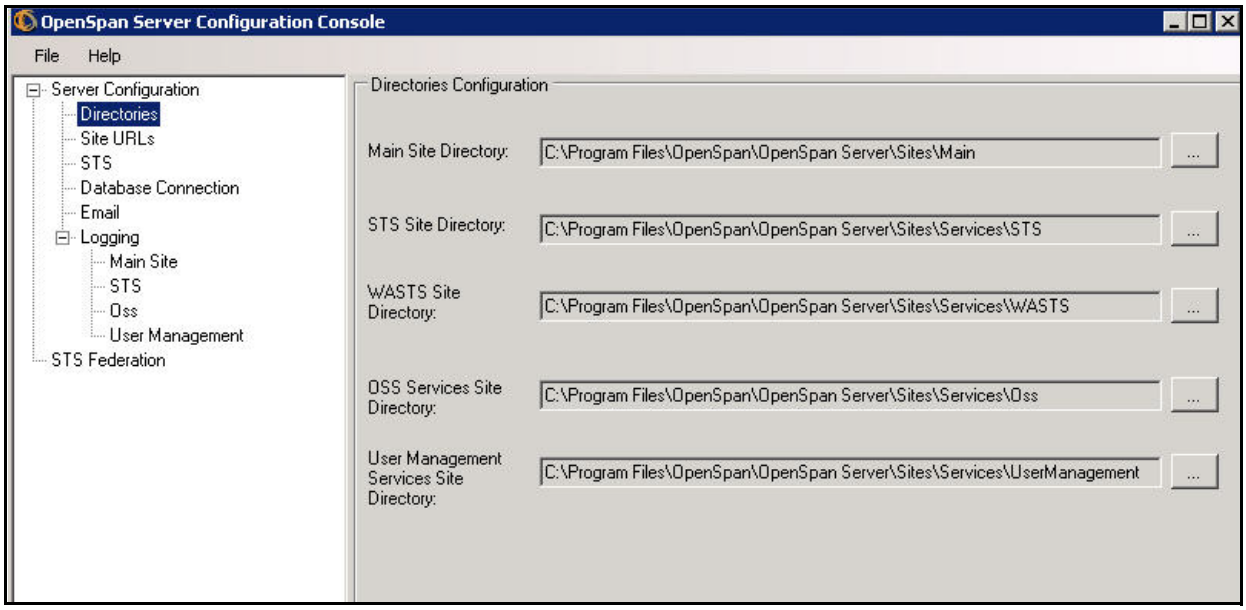
The OpenSpan Server Configuration Console is used to enter initial settings and establish communications with the database.

Note: Open the Configuration Console as the Administrator.

Caution: Once settings are entered and verified in the Server Configuration Console, changing them will change base configuration files for many interrelated server properties. The effects could be adverse.

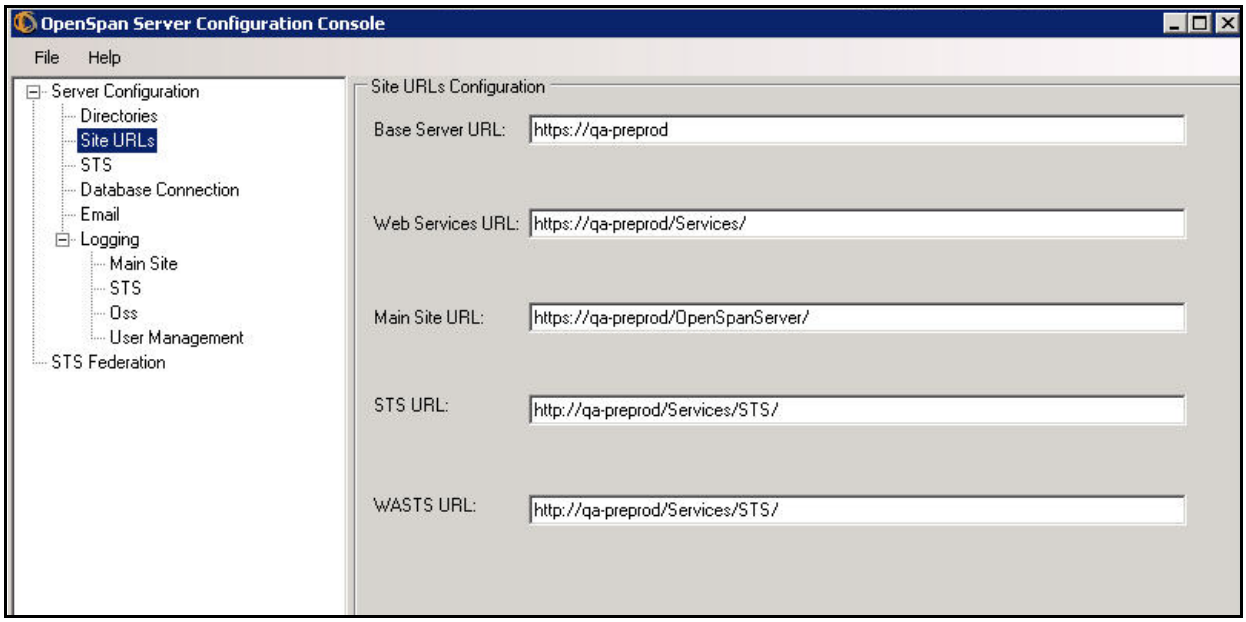
Server Configuration

1. Open the OpenSpan Server Configuration Console to complete the configuration of the OpenSpan Server after installation.
2. Navigate to the Directories node and verify the directory structure. By default these should be located under the OpenSpan Server installation folder under the Sites subdirectory as follows:



- Main Site Directory: Sites\Main
- STS Site Directory: Sites\Services\STS
- Windows Authentication STS Directory: Sites\Services\WASTS
- OSS Services Site Directory: Sites\Services\Oss
- User Management Services Site Directory: Sites\Services\UserManagement

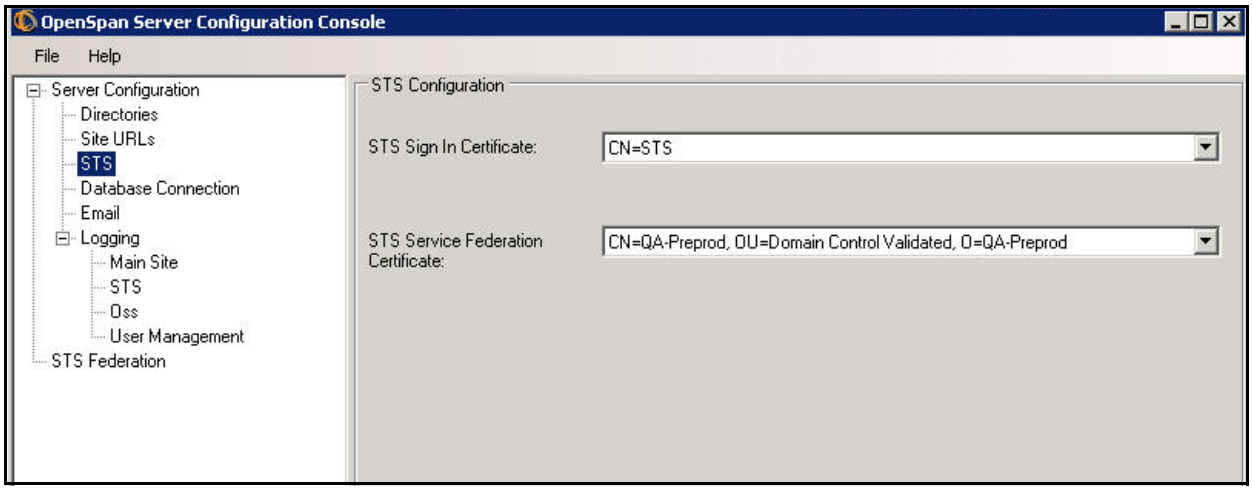
3. Navigate to the Site URLs node and verify that the URLs for each website are correct.



Note: The default settings enabled for https or unencrypted http are shown here.

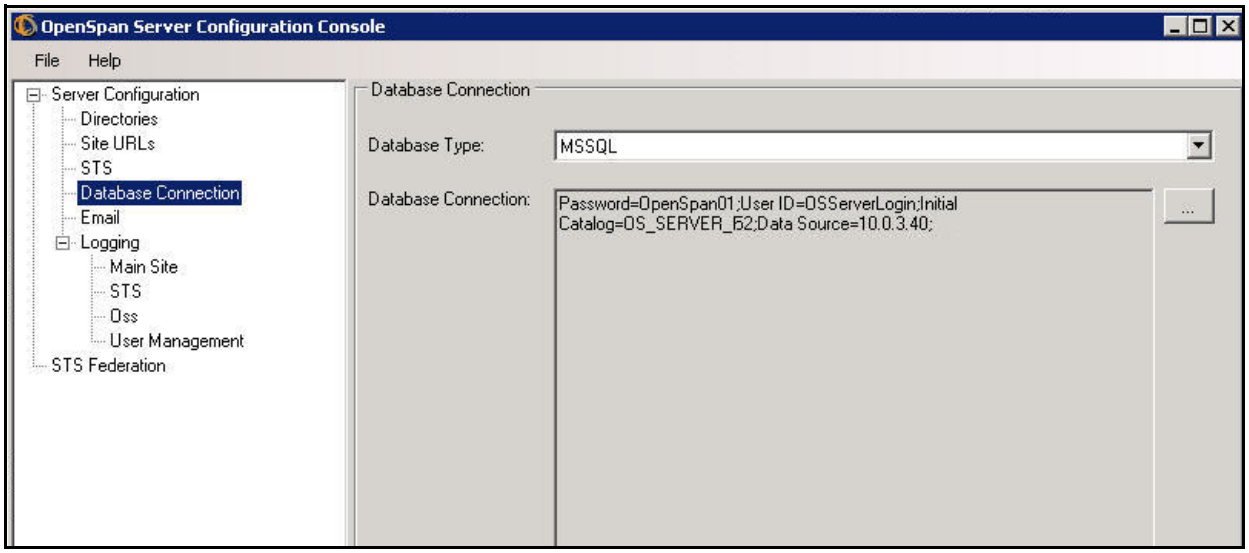
- Base Server URL: The URL to the physical server on which the OSS is installed, typically a string like https:// hostname.
- Web Services URL: The URL to the OSS services base address, typically a string like https://hostname/Services.
- Main Site URL: The URL to the main OSS web site, typically a string like https://hostname/OpenSpanServer/.
- STS URL: The URL to the security token service web site, typically a string like http://localhost/ServicesSTS/.
- WASTS URL: The URL to the Windows Authentication STS site, a string like http://localhost/Services/STS.

4. Navigate to the STS node and select the appropriate certificates for the STS.

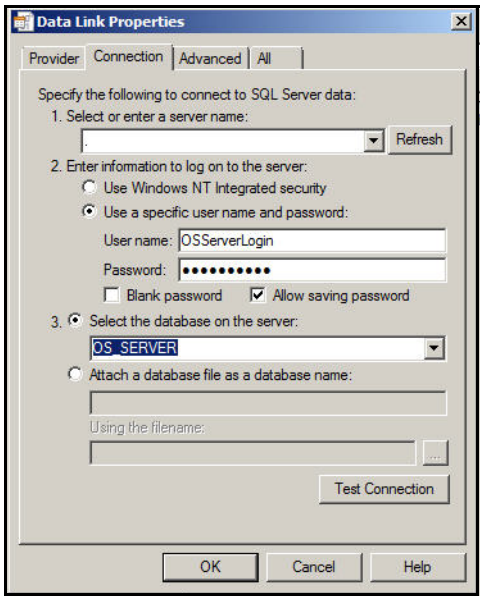


- STS Sign In Certificate: The certificate used by the STS for sign in verification.
- STS Service Federation Certificate: The certificate used for SSL/STS machine binding.

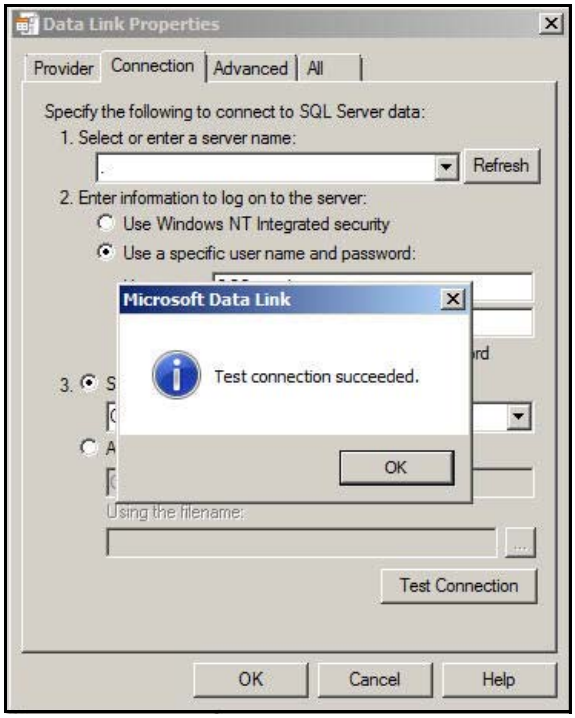
5. (Ensure first that the proper databases are created or updated with the scripts located in C:\Program Files\OpenSpan\OpenSpan Server\Database\Scripts\MSSQL.) Navigate to the Database Connection node and configure the connection to the **RDBMS used by the OSS services**:



- Database Type drop-down — **Note**: Supported databases for OpenSpan Server 5.2 are limited to MSSQL.
- Click the button adjacent to the Database Connection field. This will display the Data Link Properties dialog as shown above for the appropriate database type. Enter the connectivity information and click **OK**. Check the Allow Saving Password button if storing the login credentials in the database connection string.

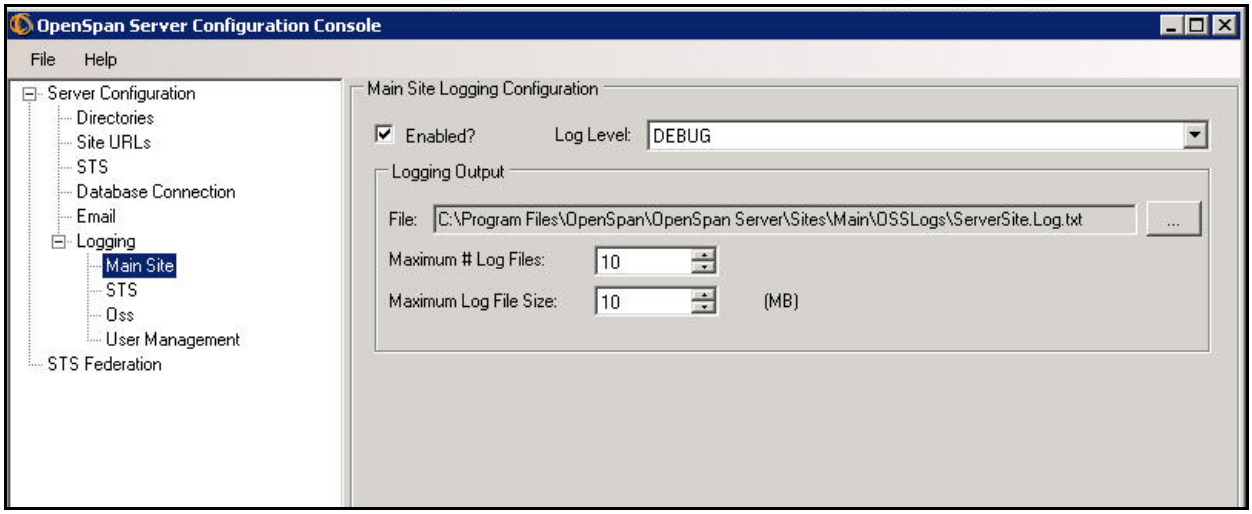


- Test the connection to ensure the database can be contacted successfully.

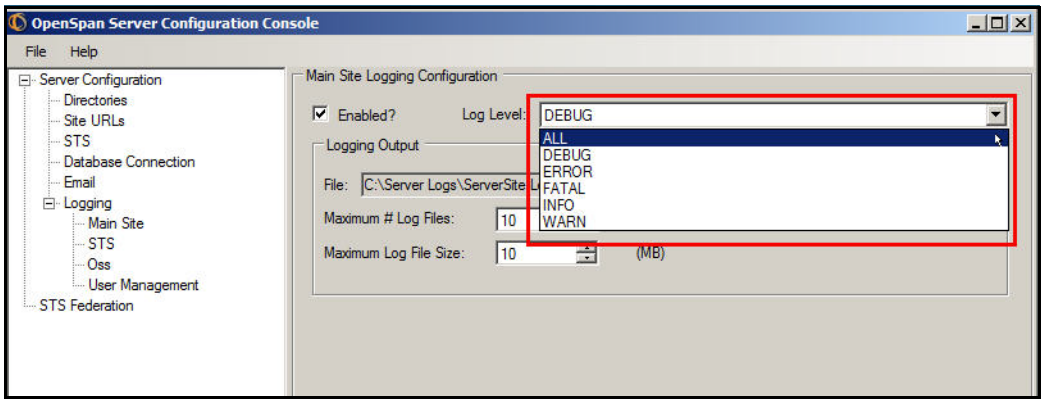


Logging

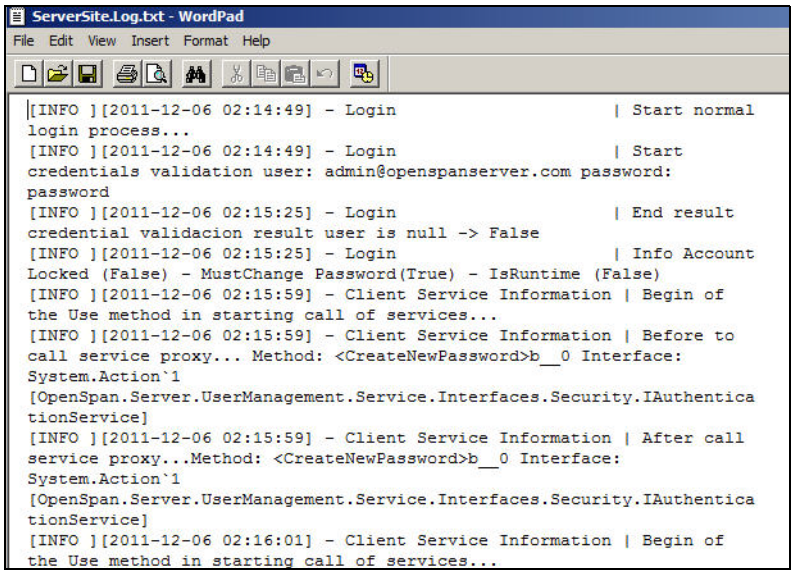
Navigate to the various Logging nodes to configure the logging features for the individual web sites — each page is identical, but operates on the logging configuration for the indicated site by the node name.



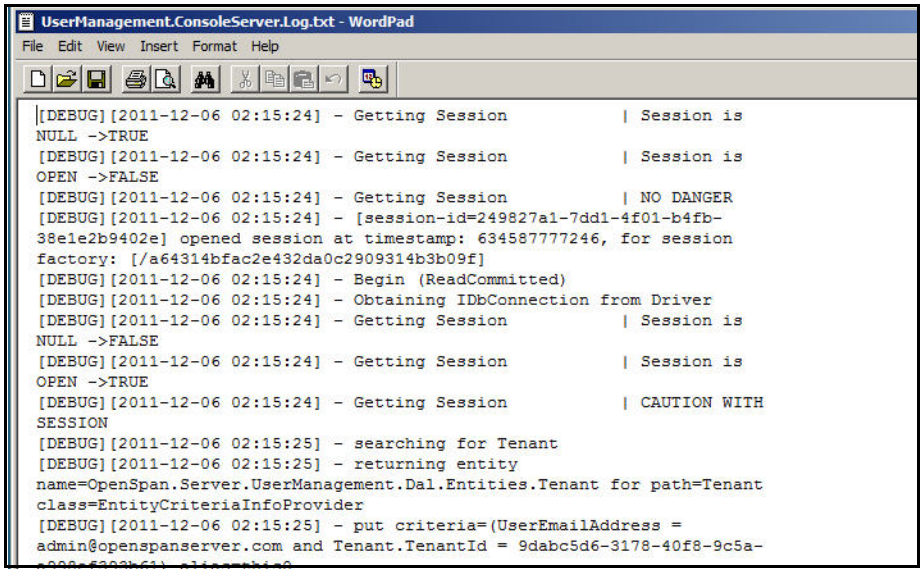
- The **Enabled** check box will enable or disable file-based logging for the web site.
- The Log Level drop-down list box allows for filtering of logging based upon the severity of the items being logged. Debug is the most verbose, whereas Fatal is the least verbose.



- Level names are appended to the log reports, as shows in the examples here.
  - **Info** Logging

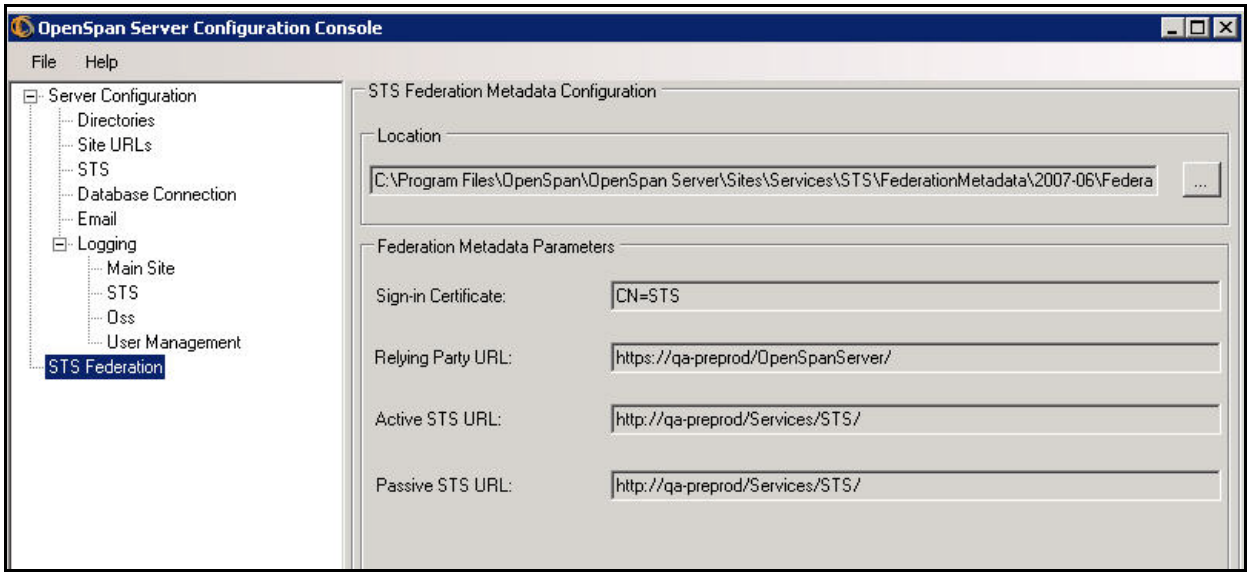


  - **Debug** Logging



STS Federation

1. Navigate to the STS Federation Metadata node and verify the following:



- Note the location of the FederationMetadata.xml file for the STS in the Location field above, verifying that the folder location exists. This file will be written with revised federation metadata when the configuration is saved via the **File | Save** menu.
- The Update STS Federation Metadata button provides the ability to immediately generate the FederationMetadata.xml file in the location indicated using the supplied parameters. Note that the parameters are obtained from the other pages in this utility.
- Token security is configurable upon installation and after with the assistance of an OpenSpan consultant.

2. Select **File | Save**. This will save the configuration information in the web.config files of the four OSS websites, update the STS FederationMetadata.xml file, and then execute federation between the main OSS website and the STS. A progress window is displayed showing the steps in the process.

3. Exit the OpenSpan Server Configuration Console.

Sign-on to OpenSpan Server

Signing On

1. Open a Web browser — Internet Explorer is recommended. Navigate to the OpenSpan Server Uniform Resource Locator (URL) specified by your OpenSpan Server installation team. The sign-on dialog box appears.

2. Administrators typically have an administrator set of logon credentials.

   Default Tenant Administrator login:

   - Username: **admin@openspanserver.com**
   - Password: **password** (all lower case)

3. Other users have different levels of access, depending on how the user account is set up. See “Groups and Users” on page 4-6.

4. Click **Sign In**.

5. If prompted, **Change your password**. **Continue**. The Welcome Screen Displays.

Chapter 3 OPENSPAN STUDIO 5.2 CONFIGURATION

In this Chapter

This chapter of the OpenSpan Server 5.2 Administration and User Guide describes how OpenSpan Studio 5.2 is configured to work with OpenSpan Server 5.2 to produce and upload OpenSpan solution projects. This chapter is organized into the following topics:

Note

OpenSpan Studio 5.2 is a prerequisite for OpenSpan Server 5.2.

Configuring OpenSpan Studio for OpenSpan Server

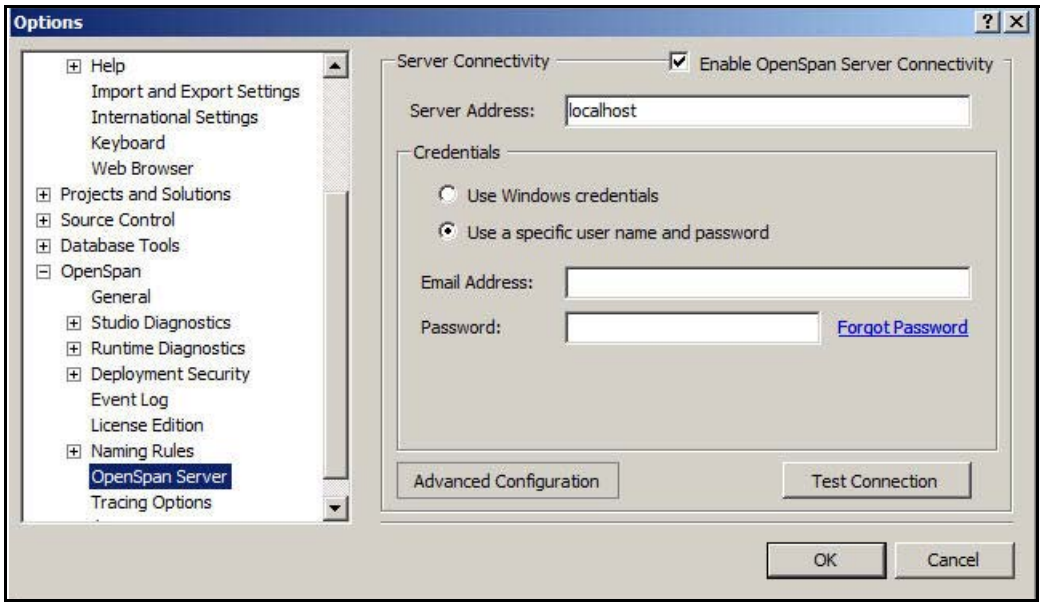
This section explains how to configure OpenSpan Studio to communicate with OpenSpan Server to allow upload of OpenSpan solution runtime packages.

Notes

- If working remotely and disconnected from a network connection to OpenSpan Server, i.e. in a Virtual Machine or other isolated environment, you may wish to disable Server Connectivity when working in OpenSpan Studio. This is because the configuration automatically seeks a network path to OpenSpan Server and can cause delays, timeouts, and sluggish behavior while trying to connect. Re-enable when a network connection is valid.
- Connections between OpenSpan Studio and OpenSpan Server are “domain-sensitive.” Ensure you are connected to an instance of OpenSpan Server within the same domain.

User Steps

1. From OpenSpan Studio, access the configuration panel through the menu **Tools | Options**. The following window displays. Select **OpenSpan | OpenSpan Server** to expand the configuration choices.



2. The following fields are used to specify the connectivity information for the server as well as the client security credentials:

TABLE 1. OpenSpan Server Options

Property	Description
Server Address	The URL to the base services site. Autofills other required server-side entries such as STS. Enter a domain address such as www.servername.com or an IP address. Default is **localhost**; entering protocols such as http:// or https:// will generate an error message. Server information is written to the StudioConfig.xml file. Additions or changes may be necessary to match particular server configurations. OpenSpan consultants can provide the exact settings.
Use Windows credentials	The user or workstation's Windows Authentication login.
Use a specific user name and password	Other logins manually added via OpenSpan Server. See "Manually Add New User" on page 4-7.

3. From the **Options | OpenSpan Server** panel, choose **Advanced Configuration | General**. The following dialog box displays.

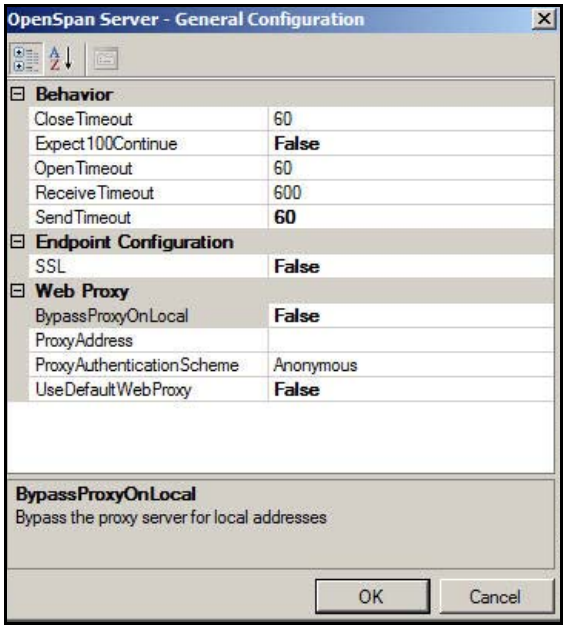


TABLE 2. OpenSpan Server General Configuration

Property	Description
Close Timeout	Time elapsed for a connection is directed to close before an exception message is generated.
Expect 100-Continue	Defaults to False. Sets 100-Continue behavior for put and post commands such as authentication challenges.
Open Timeout	Time elapsed for a connection is directed to open before an exception message is generated.
Receive Timeout	Time in seconds an application waits for a request to complete.
Send Timeout	Time in seconds an application waits for a response to complete.
SSL	True/False. Enable Secure Sockets Layer. Server must be configured for SSL as well (address preamble: https://).
BypassProxyonLocal	Default is False. If a proxy is configured, allows connections to bypass for local addresses.
ProxyAddress	Uniform Resource Identifier of the proxy.
ProxyAuthenticationScheme	Default is Anonymous. There is a range of selectable values. Contact your server administrator for your parameter.
UseDefaultWebProxy	Default is False; sets use of local auto-configured HTTP proxy.

4. From the **Options | OpenSpan Server** panel, choose **Advanced Configuration | Event Services**. The following dialog box displays.

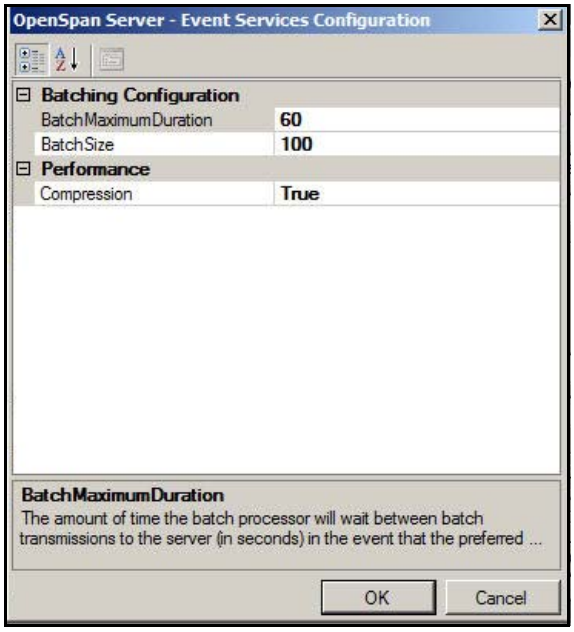


TABLE 3. OpenSpan Server Event Services Configuration

Property	Description
BatchMaximumDuration	The amount of time in seconds the batch processor waits between batches before sending to the server.
BatchSize	Number of messages per batch. Special note: For High-Level Events, batching is often required because of the high volumes of events records generated. It quite often happens, however, that when batching is invoked it's forgotten temporarily when testing throughput. Administrators who are waiting for the OS_Events table in the OS_Server database to be populated can think that no events are being reported, when in fact there is latency due to batches being assembled before being written to the table.
Compression	Default is True. Enables compression for batches sent to the server.

Certificates

Certificates are generally not required for OpenSpan Studio 5.2 to connect to OpenSpan Server 5.2, unless required to sign runtime packages upon upload. In that case, a certificate matching the server's certificate authority should be imported to the workstation's certificate store under Trusted Root Certification Authorities.

Matching Configuration and Promotion Levels to Runtime Packages for Debugging

Because OpenSpan Server 5.2 allows Business Analysts and people with similar roles to further modify Runtime deployment packages, OpenSpan Studio 5.2 has an additional toolbar for developers to select different versions of the same solution so that debugging and changes made in OpenSpan Studio are to the correct variant.

The variables are:

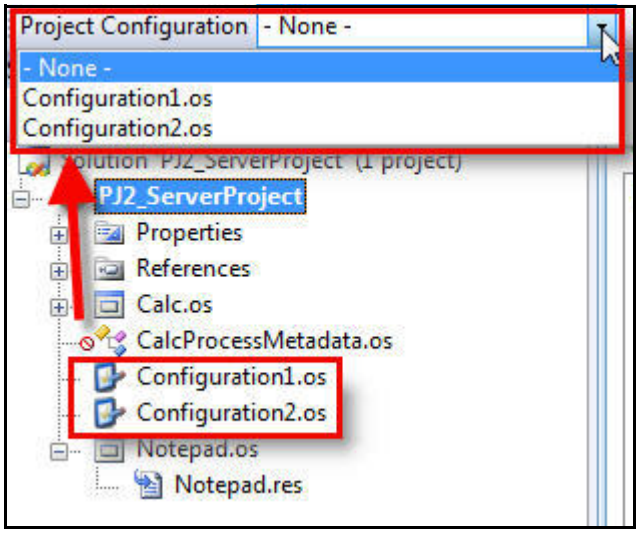
- Project Configuration — Assigned configuration(s) from OpenSpan Studio
- Run as Group — Reflects the Group assignment(s) made to the package on OpenSpan Server
- Feature Set Level — Development or Production, as designated on OpenSpan Server

Technical advisory: Solutions are given a unique package ID in OpenSpan Studio. The use of the toolbar in OpenSpan Studio described in this section in conjunction with runtime package administration on OpenSpan Server assumes that a solution's package ID is unchanged. Be aware that if you perform a "Save As" for a solution in OpenSpan Studio, the package will receive a completely new package ID and version features such as this toolbar will be ineffective. Additionally, of course, OpenSpan Server will not recognize the new package ID and changes such as feature sets would be lost.

Project Configurations

Developers can add project items called Configurations in OpenSpan Studio that capture and enforce certain default properties from interrogated application controls. For a full explanation of Configuration components, see the OpenSpan online help; the main point in this publication is to show that when working jointly between OpenSpan Studio and OpenSpan Server on a solution that has multiple configurations assigned, the developer now can ensure that any debugging or changes to the solution are being done to the correct variant.

The Project Configuration default is **None**. Any available configurations are accessible from the drop-down list box as shown here.

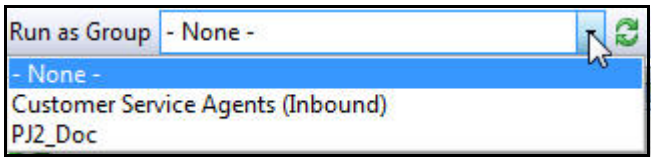


See also “Assigning Configurations to Groups” on page 4-31.

Note: When deploying configurations from OpenSpan Studio to OpenSpan Server, ensure you select **Deploy All Configurations** from the solution menu choices.

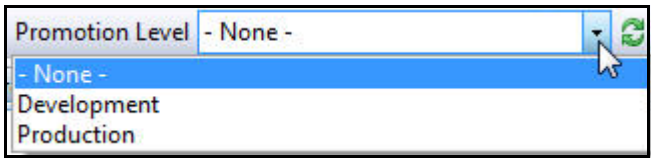
Run as Group

And finally, developers making changes to deployed solutions in OpenSpan Studio can also select the correct Group-assigned parameters for a solution as shown here. The Server-side explanation is available here: “Assigning Groups to Packages” on page 4-34



Promotion Level

OpenSpan Studio now allows the same flexibility in selecting appropriate Promotion levels for a solution/ deployment package. To read more about how OpenSpan Server assigns these settings, see “Assigning Feature Set Levels” on page 4-33. When debugging solutions in OpenSpan Studio, ensure that you select the correct Promotion Level for the package’s feature sets.



Producing Process Automations

General

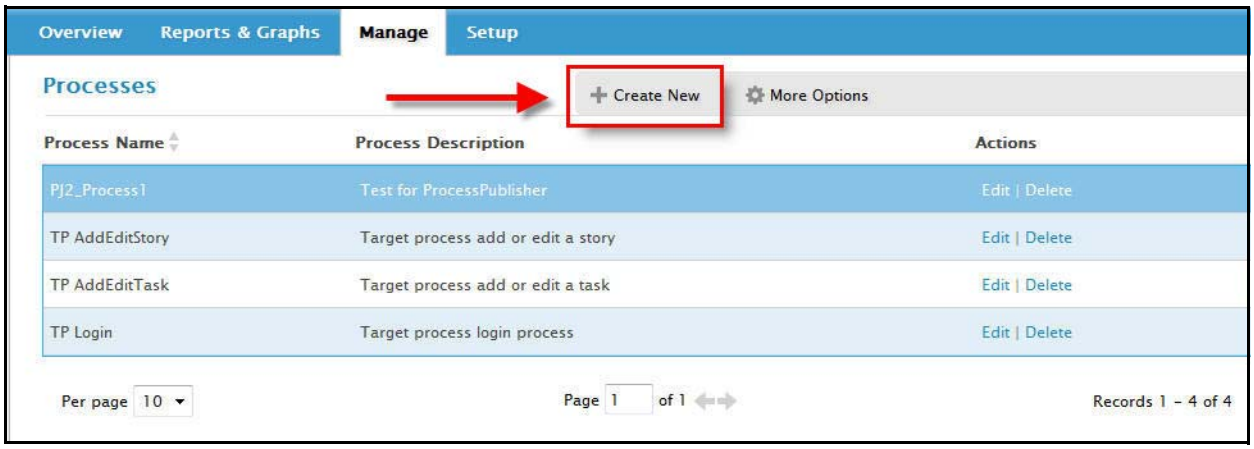
And finally, Process automations created in OpenSpan Studio can send specific Events to the Events table in the OS_Server database that don't change the process state from desktop Runtime solution packages. There are infinite ways to build these automations based on different business cases, but they use a new OpenSpan Studio 5.2 component called the ProcessPublisher, which writes its own category of data to the Events table.

Process Automation Creation

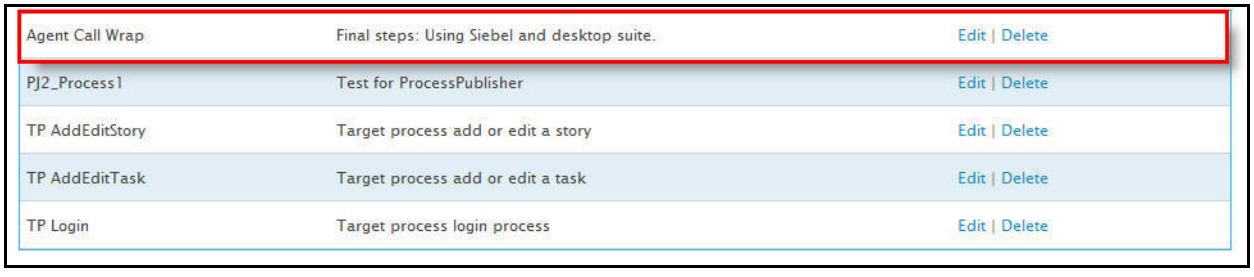
Processes and metadata are defined on OpenSpan Server 5.2, as explained in "Managing Processes" on page 4-36.

New Process

First a process is created. A Server user creates a new process...

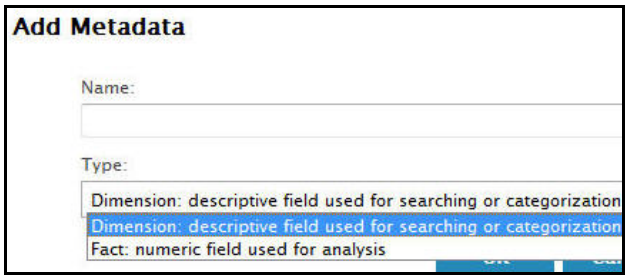


...and saves to server. This list of processes will be available to OpenSpan Studio, if connected to the server. The individual processes can be assigned to Runtime users via OpenSpan Server, with or without a runtime package. The assigned processes will appear in the drop-down list box on the Process Toolbar (see "Process Toolbar" on page 5-7).



Metadata Creation

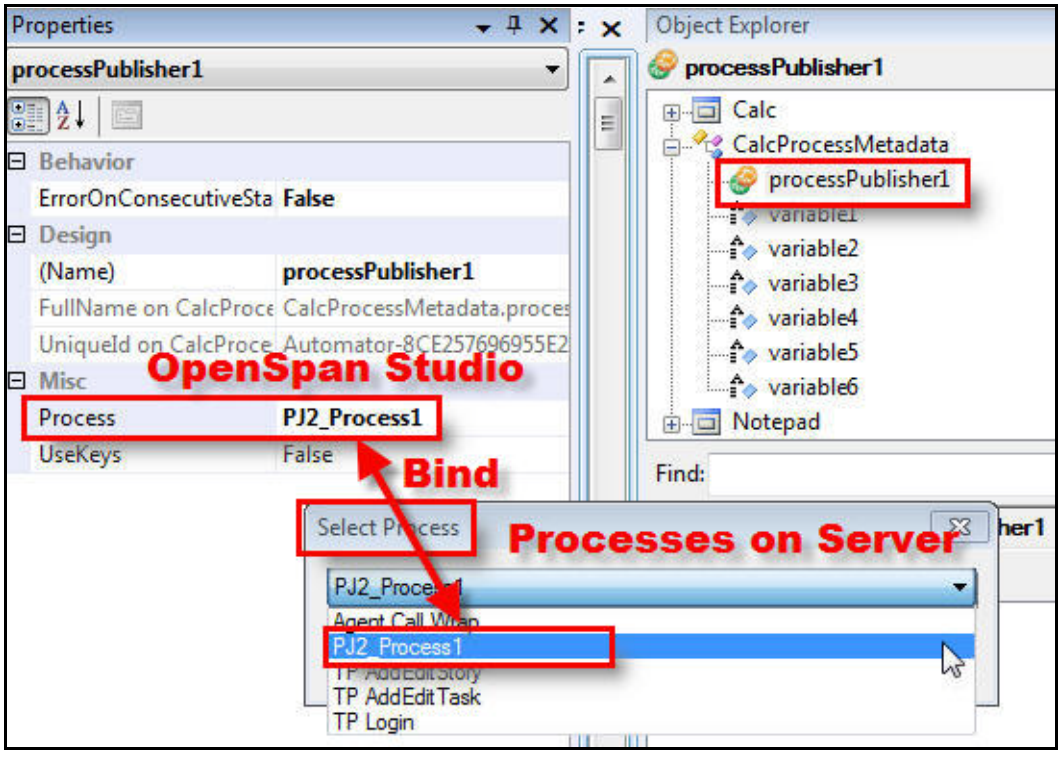
Then metadata — dimensions or facts — is added to the Process.



Once the metadata is associated with a Process on OpenSpan Server, then Processes and associated metadata are ready for inclusion in process automations.

ProcessPublisher

A ProcessPublisher component is added to the solution in OpenSpan Studio, and a process bind property connects the ProcessPublisher to the metadata for any defined process residing on OpenSpan Server.



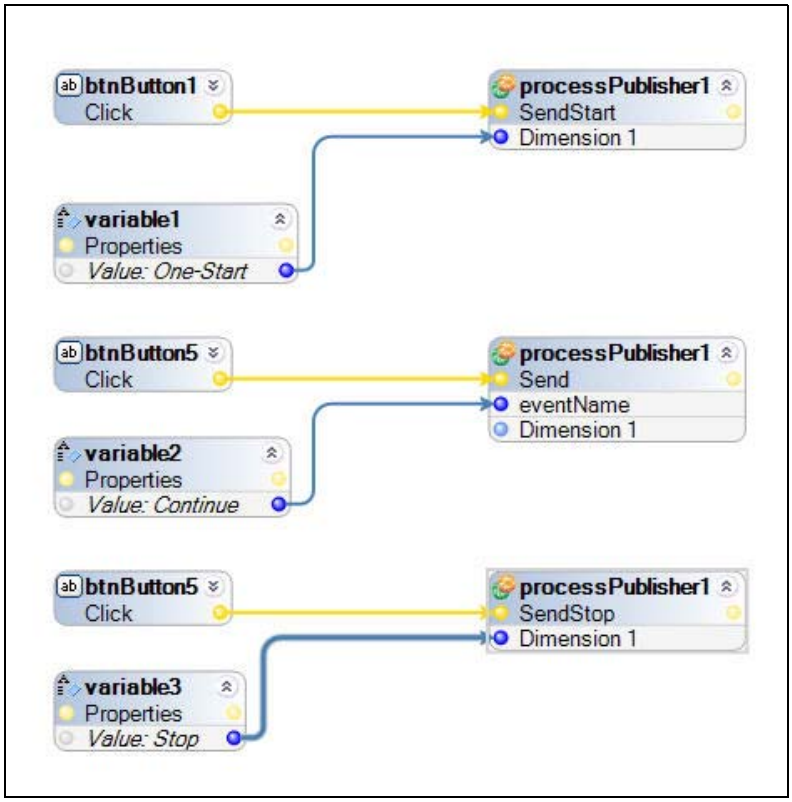
1. Access the Properties for the ProcessPublisher.
   - Set ErrorOnConsecutiveStarts to **True** to display an error if SendStart happens more than once before a corresponding SendStop in the same automation.

- If set to **True**, configure exception handling such as Try-Catch blocks between components. Additionally, set the Suppress Errors property to **True** for the automation
- Set ErrorOnConsecutiveStarts to **False** (default) to accept SendStarts without terminating SendStops in the automation.
- The ProcessPublisher has a method, **Abort**, to be employed in automations to refresh component's state to brand new. It is the equivalent of the Abort button in the Process Toolbar (see "Process Toolbar" on page 5-7), and interrupts and resets the process state.

2. Bind the automation to a Process existing on OpenSpan Server. Choose the **Process property**, and the Select Process dialog box displays. Choose the correct process for the automation.
3. Set UseKeys to **True** to enable keys to account for each instance of the automation as a different process. (The default is **False**.)

Example Automation

Then depending on business and reporting objectives, the SendStart and SendStop ProcessPublisher methods can be used to "tag" or identify important desktop user actions for analysis. Process Events are sent to the Events table of the with a property of 4. This example attaches metadata to button controls and sends it when the controls are clicked as part of a pre-defined process. A middle event has been inserted in this example, just to show that intermediate events of interest can be placed in the automation and even intermixed with ProcessPublisher methods, depending on the need. ProcessPublisher is a very flexible component.



Upload to Server and Runtime Download

Process automations are uploaded to OpenSpan Server along with their parent solutions. Upon download to the end-user desktop, they can function independently of other automation or added Feature logic, but continue working at all times when triggered to send custom metadata to the OS_Events database.

Chapter 4 OpenSpan Server 5.2 Administration and Management

In this Chapter

Welcome to the Server Administration and Management Chapter of the OpenSpan Server 5.2 Administration and User Guide. This chapter is organized into the following topics:

General

OpenSpan Server 5.2 is a Web-based application environment that allows users of OpenSpan software to:

- Set up Groups of Users, and import Users
- Receive uploads of runtime deployment packages from OpenSpan Studio
- Add features such as tooltips and default values to end-user desktop applications referenced by OpenSpan Runtime solutions
- Assign the following to Groups:
  - Runtime deployment packages
  - Feature sets
  - Different runtime deployment package configurations
  - A Process Toolbar for instrumenting user processes
- Send High-Level and Custom events data from user desktops to the Events database and generate visualizations and reports

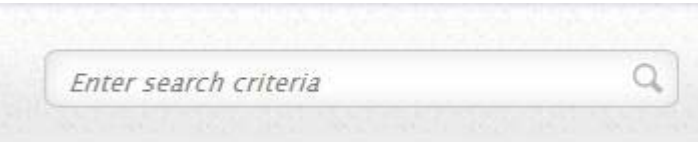
OpenSpan Server 5.2 serves as a crucial middle piece in an architecture consisting of OpenSpan Studio, OpenSpan Server, and OpenSpan Runtime. While it is still possible, as in earlier OpenSpan releases to develop solutions in OpenSpan Studio and deploy them directly to the desktop, OpenSpan Server provides a ready-made administration tool.

Dependencies

OpenSpan Studio 5.2 is a prerequisite for OpenSpan Server 5.2. It creates the runtime solutions that are uploaded to the server for distribution to user desktops.

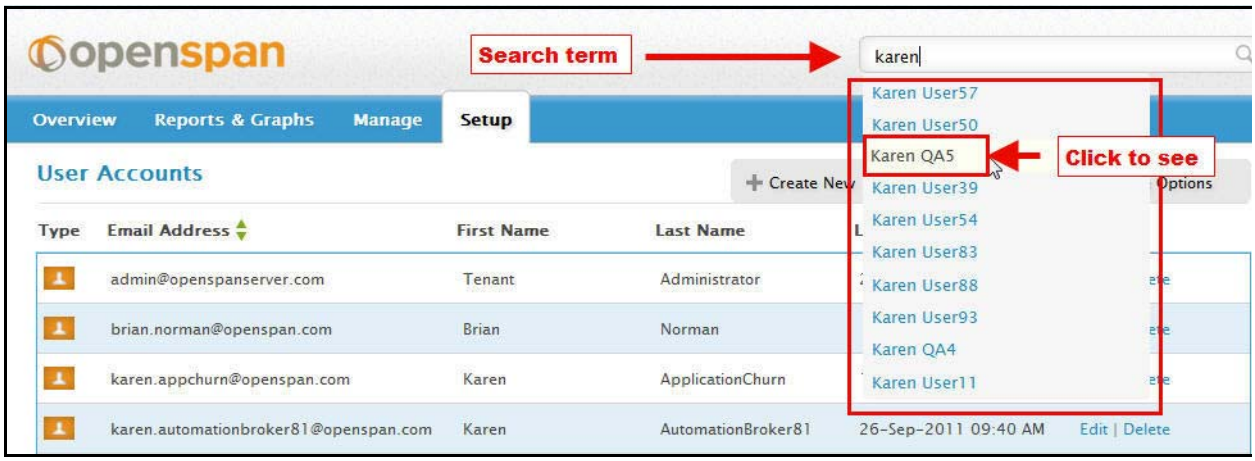
Master Tip — Find Searchable Content

A brief word on one of OpenSpan Server's common-menu features, the Search box. On screens having searchable lists, such as Users, Groups, and in other selection screens, use the Search box to quickly find your selections.

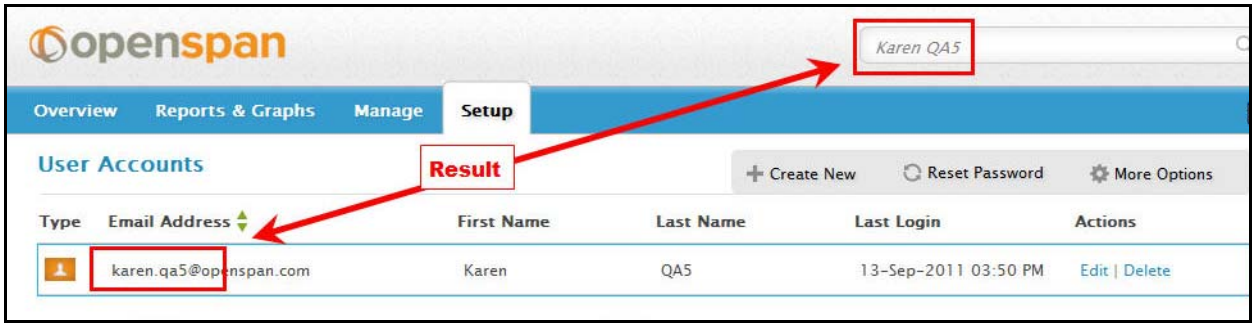


The example below shows a search for the name **Karen** in User Accounts.

1. Enter the desired search term string.
2. The Search box expands downward with any items located meeting the criteria.



3. Choose a result from the drop-down list and click; the choice displays.



Note: The Search box is a powerful tool. It accepts basic text strings, and also contextual wildcards, such as the **%** sign. Adding a % sign indicates that the text preceding or following the pattern can be ignored. For example, **%Bob** will include items that have **Bob** somewhere in the string, not just at the beginning.

Sign-on to OpenSpan Server

Signing On

1. Open a Web browser — Internet Explorer is recommended. Navigate to the OpenSpan Server Uniform Resource Locator (URL) specified by your OpenSpan Server installation team. The sign-on dialog box appears.

2. Administrators typically have an administrator set of logon credentials.

   Default Tenant Administrator login:

   - Username: admin@openspanserver.com
   - Password: password (all lower case)

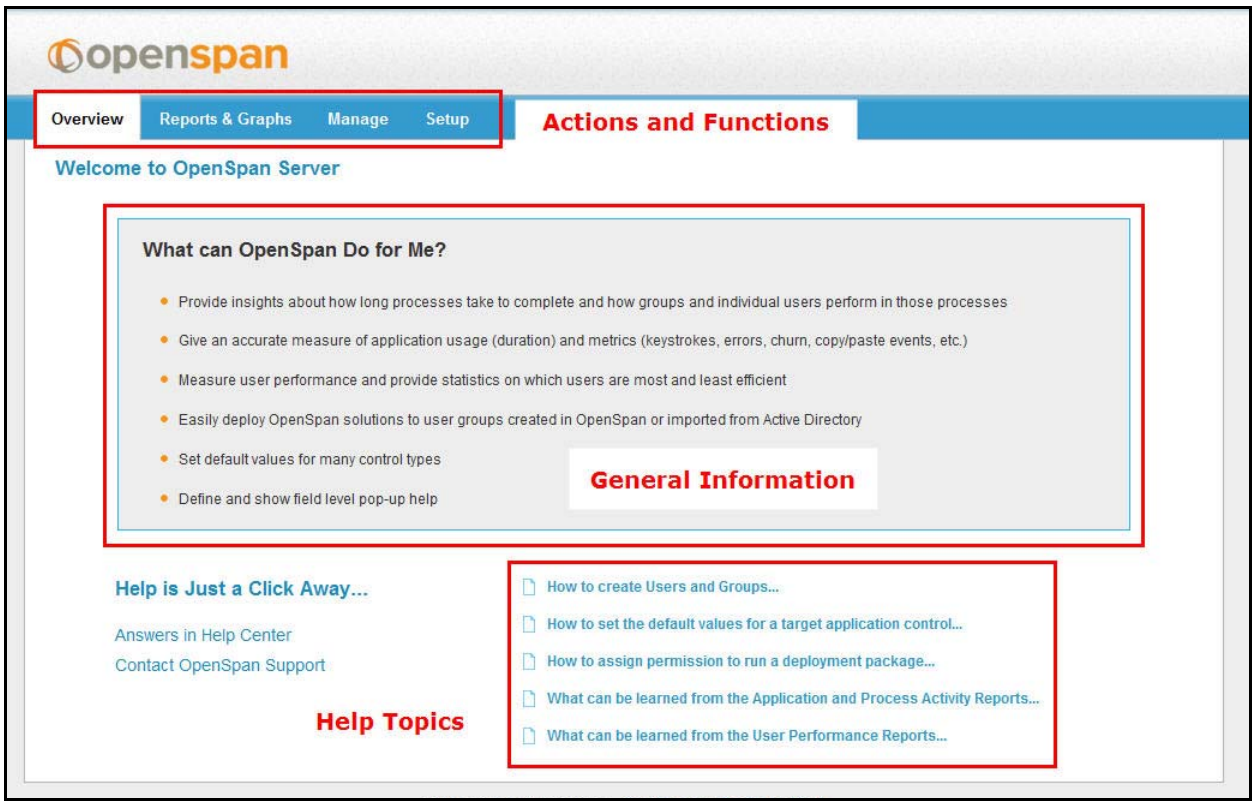
3. Other users have different levels of access, depending on how the user account is set up. See “Groups and Users” on page 4-6.

4. Click **Sign In**.

5. If prompted, **Change your password**. **Continue**. The Welcome Screen Displays.

Welcome Screen

The OpenSpan Server Welcome Screen has a number of elements.



- The top menu of actions and functions contains Reports and Graphs, Management, and Setup tasks.
- The general information area explains basic facts about the application.
- Help topic links on the bottom assist in basic tasks.

Groups and Users

A key organizing principle of the architecture between OpenSpan Studio, Server, and Runtime is the use of Groups and Users who make up those Groups. Groups and Users are administered via OpenSpan Server. There are no pre-loaded Groups, for reasons that will become clear as we look closer, and users are added based on your enterprise's user population.

Some Facts about Groups under OpenSpan Server

- Groups are a central concept in the OpenSpan Server architecture. Group membership defines what a user does and doesn't do. For example, a desktop OpenSpan Runtime user might automatically download and run a process improvement automation that is completely different from one that is designated at the server level for another user who is assigned to a different group. Or, the same desktop automation may be assigned with different tool tips to two similar end-users based on native language or skill level. This is possible if the two users are assigned to different Groups on OpenSpan Server.
- Groups are managed in OpenSpan Server by choosing **Setup | Groups**.

About Users

- Users are the people who make up Groups. They are given logon credentials to access OpenSpan Server. The way they log on and what they do depends on their job or source. Here are some examples.
- **Users created on OpenSpan Server** have four roles:
  - Server **administrators** are usually skilled IT specialists who install, configure, maintain, and update the OpenSpan Server software, the server operating system (such as Windows Server 2008), and database operations. They have full access to all functions.
  - Software **developers** using OpenSpan Studio 5.2 can automatically connect to OpenSpan Server and upload Runtime deployment packages destined for other user desktops, but don't necessarily have to use other server functions for administering or modifying the packages. Developers cannot set up new Users and Groups, or access site settings such as Events enabling for Groups.
  - **Business Analysts** may access OpenSpan Server to modify Runtime desktop packages and assign them to Groups of users. Business Analysts cannot set up new Users and Groups, or access site settings such as Events enabling for Groups.
  - **Report Users** receive access to the reports menus and functions.
  - **Runtime users** normally connect automatically from their desktops to OpenSpan Server using OpenSpan Runtime, which sends events data, downloads runtime solutions packages, and handles other functions to improve end-use processes.

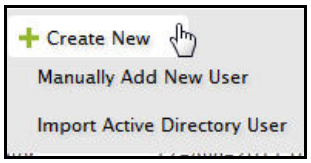
Note: You can not change your *own* user role once set up. An administrator must change your role. And an administrator's role must be changed by a different administrator.

- Users from Active Directory (AD) Import may be divided prior into different criteria and groups depending on their AD status.

Adding Users

Users can come from two sources:

- Manually Add New User (below)
- Import Active Directory User, page 4-10. **Note**: Initial importing of Active Directory Users and Groups is performed by OpenSpan Services consultants using special utilities to ensure proper configuration.



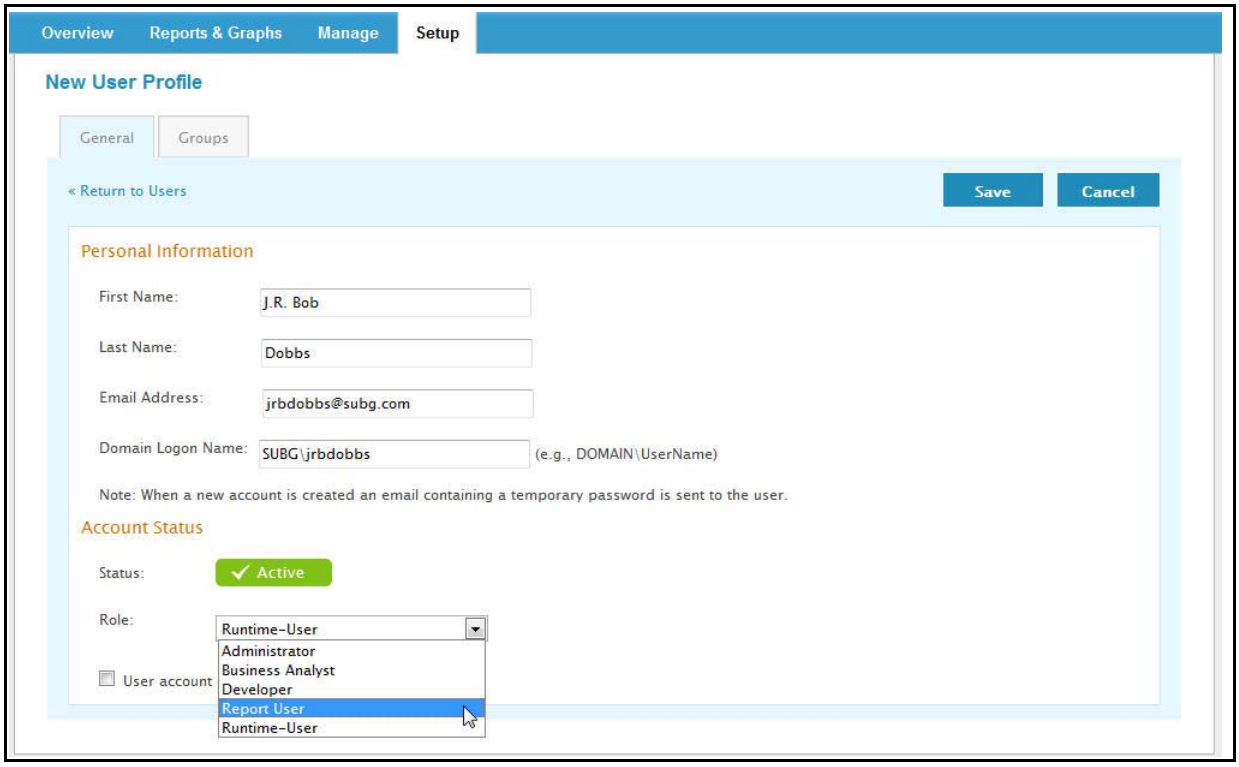
Manually Add New User

1. From the **Setup | Users** menu, choose **Create New**. The New Users Profile Screen displays. Enter a user First Name, Last Name, and email address. **Note**: The email address will be used to send password reset notices and other user communications.

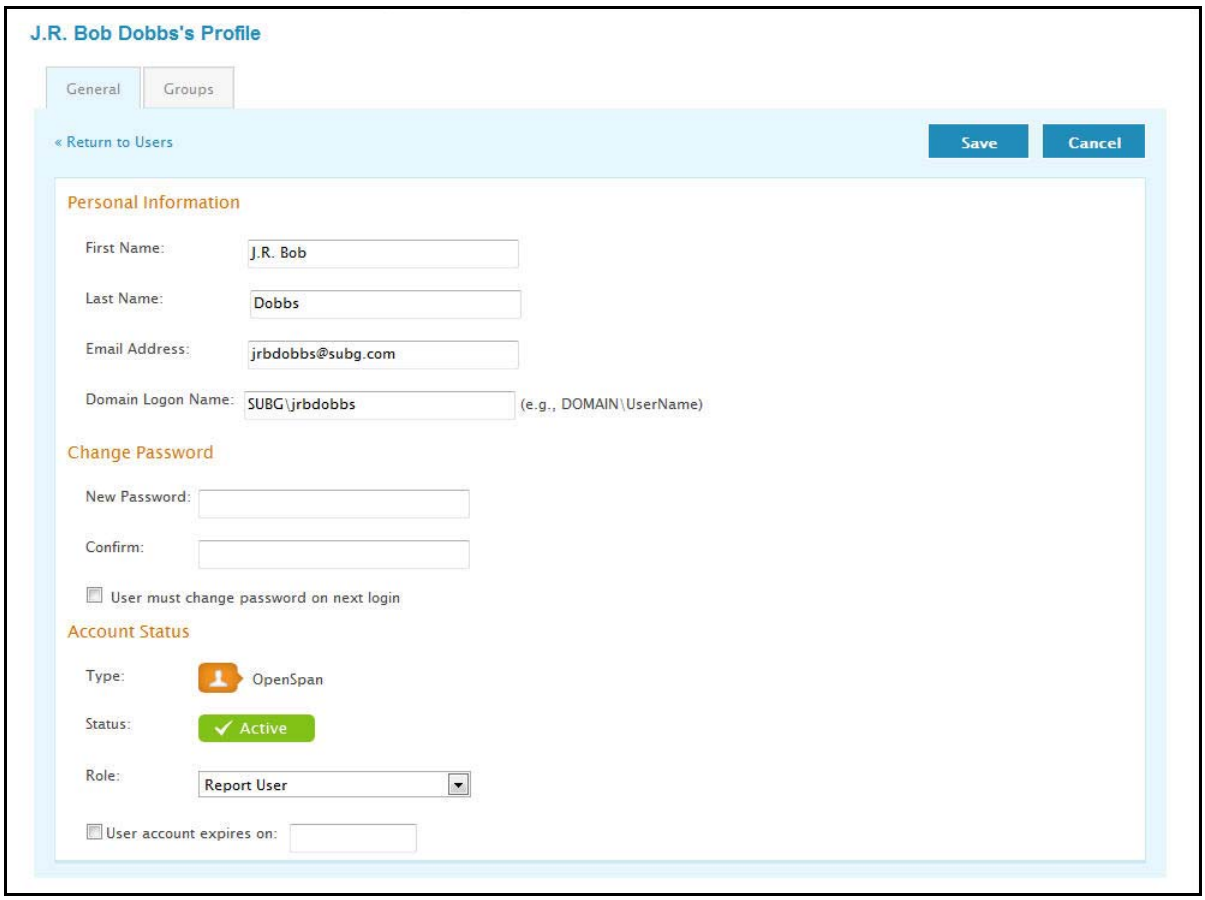
   You have the option of designating a user as:

   - Runtime user
   - Administrator
   - Developer
   - Report User
   - Business Analyst

Additionally, you can set **password expiration** in this screen.



2. Click **Save**. A New User is created. The Status defaults to Active. For Type, Yellow = non-Active Directory, Red = User is from Active Directory but no longer exists, Green = User is from Active Directory and is valid.



3. Next, click the **Groups** tab to assign the user to a Group.

- If there are no Groups to assign a User to, see “Adding Groups” on page 4-14 below.
- If at least one Group exists, see “Assigning Users to Groups” on page 4-17.

Import Active Directory Users

Note: Some Active Directory import functions will be deprecated after the release of OpenSpan Server 5.2 from the user interface below in favor of other import utilities. The OpenSpan Services specialist who enables your deployment has the latest console utilities for your deployment.

Importing Active Directory Users first requires communication between OpenSpan Server and the Active Directory server. See “Site Settings” on page 4-44.

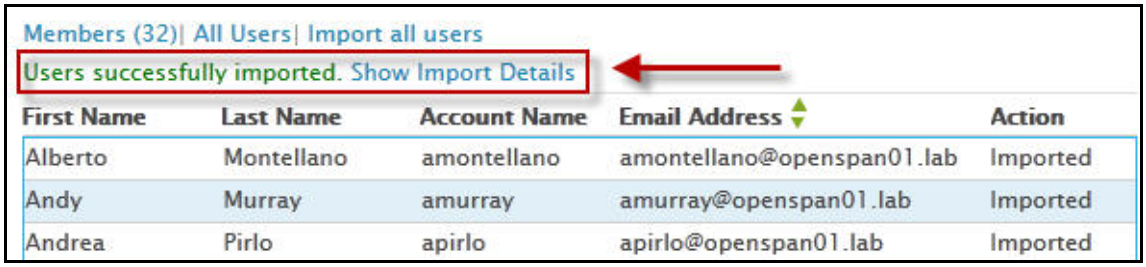
The import procedure follows the same pattern as other steps that move Users from the Available column to the Selected column.

1. From **Manage | Users**, choose Create **New | Import Active Directory User**.

Note: Only users with an email address entered in the imported database are populated in the grid. This is because email is a required field in OSS.

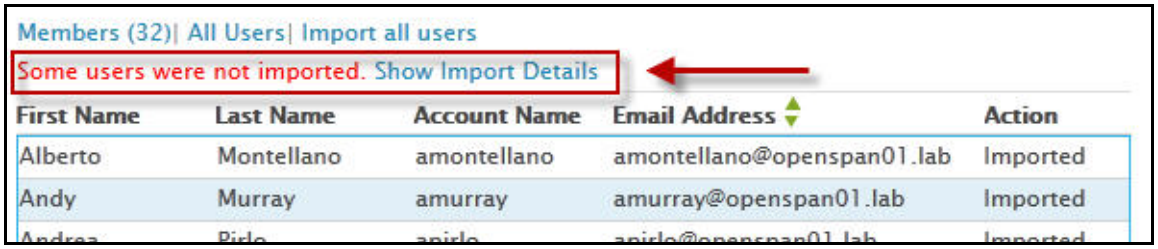
2. The grid allows three options.
   - **Members**. This option shows the already-imported users.
   - **All Users**. This option shows all available users in Active DIrectory. Users previously imported don’t have the **Add** link in the Action column next to their names, but show as **Imported**. Clicking Add imports a user to OpenSpan Server and if the user is a member of a previously-imported Active Directory Group, the User/ Group relationship is automatically made in OpenSpan Server.
   - **Import all users**. Use this option to import all available AD users to OpenSpan Server. Group affiliations are automatically assigned if the user is a member of a previously-imported Active Directory Group.

3. If there are no errors on import, the green **Users successfully imported** message displays, with a **Show important details** link.

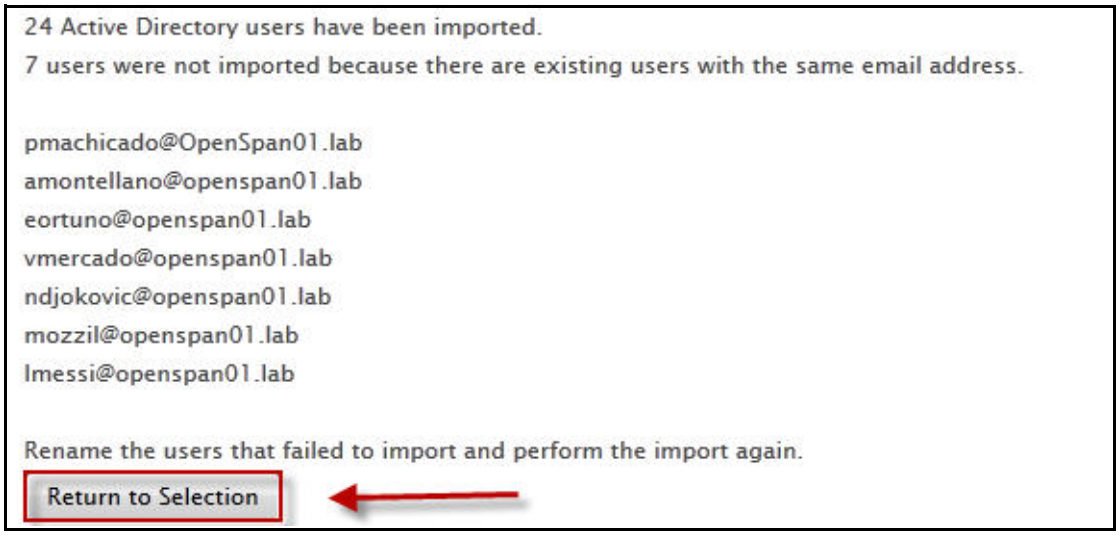


4. Clicking **Show import details** provides information on the import results. Click **Return to Selection**.

5. However, if there were import errors, the **Some errors were reported** message displays. Click S**how Import Details** to access information on the errors.



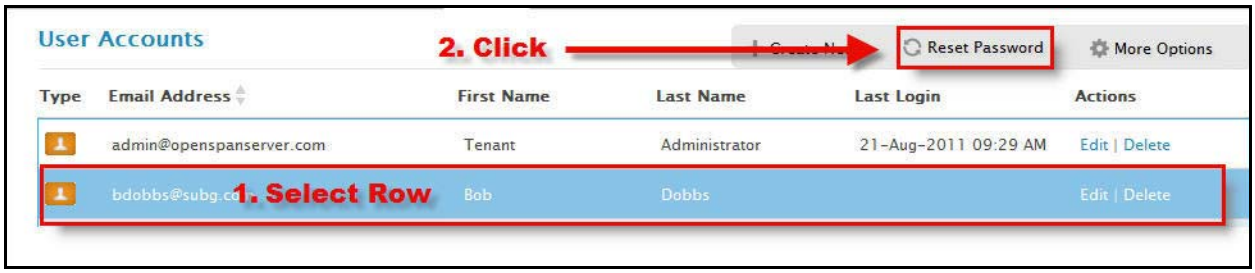
In this case, the number of successful versus failed imports displays, as well as information on how to fix the errors. **Return to Selection** takes you back to the grid.



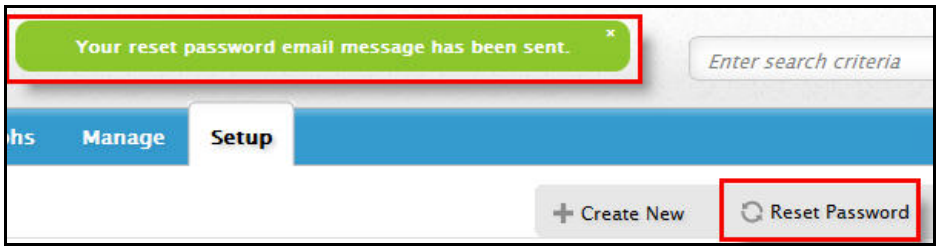
Reset Password

Use the steps below to reset a User's password. **Note**: These steps require an accurate email address for the User, whether or not the User account is manually created or imported from Active Directory.

1. Access **Setup | Users**.
2. Select the user whose password you wish to reset by clicking the user's row in the user account list.
3. Click **Reset Password**. This sends a reset email to the address in the user account list.



4. A confirmation dialog displays notifying you that a reset email was sent to the user.



Delete, Deactivate, and Reactivate Users

Two options exist for removing Users: **Delete**, which is permanent, and **Deactivate**, which retains the User account but removes it from active processes on OpenSpan Server. The steps below explain how to perform deletions, deactivations, and how to reactivate a User.

1. Choosing **Delete** from the User view allows three possibilities:
   - Deactivate — Keep the User, but remove from active service.
   - Delete — Remove the User and all attributes from the OpenSpan Server database.
   - Cancel — Return to the Users screen.

If you choose **Deactivate** and wish reactivate the User:

1. Choose **More Options | Show Inactive**.
2. The list of inactive Users displays.
3. Select the User you wish to reactivate, and choose **More Options | Activate User**.

4. The User is now active.

Adding Groups

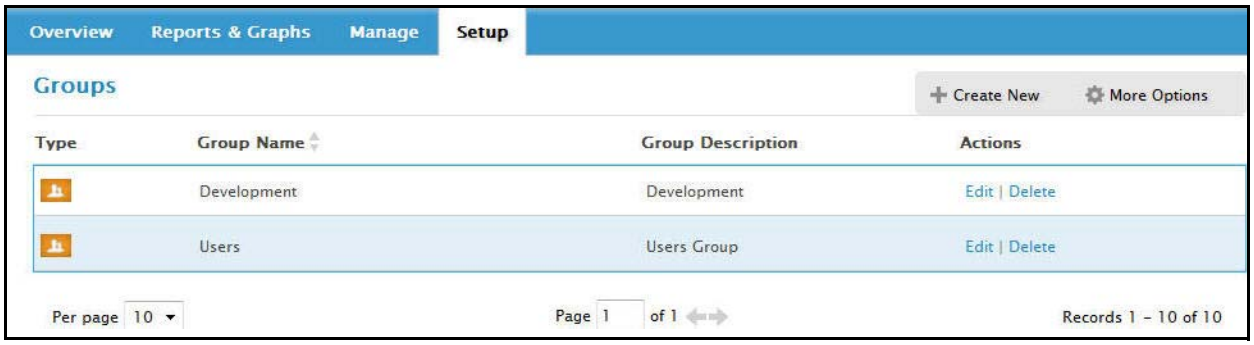
Once business requirements have defined the types and general composition of the Groups of end-users who will ultimately user OpenSpan Runtime on workstation desktops, add those Groups to OpenSpan Server.

Get started adding Groups by following these steps:

1. From the **Setup** menu, choose **Groups**.

2. The Groups administration screen displays. **Note**: If there are no Groups created, this screen will be empty.

   Note: From this screen, you can always click to highlight a Group, and then select Edit to change information or Delete. See “Delete, Deactivate, and Reactivate Groups” on page 4-16.

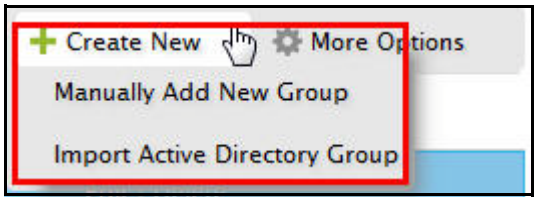


There are two sets of options:

- Create New
- More Options

Create New — Manually Add New Group

1. To create a new group, select **Create New**.

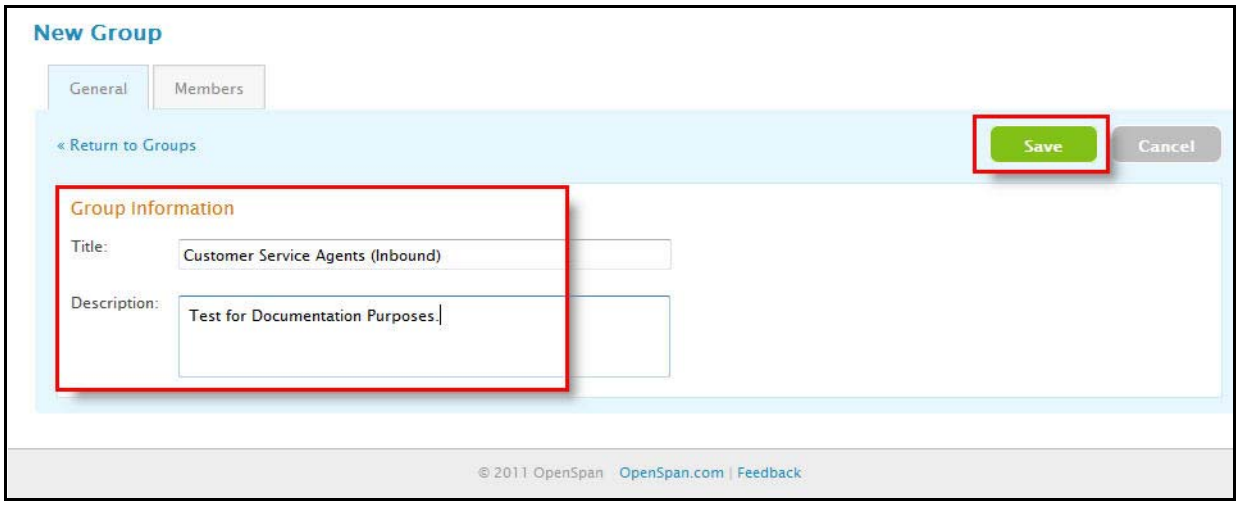


There are two further options:

- Manually Add New Group

a. Select **Manually Add New Group**. The New Group Window Displays.

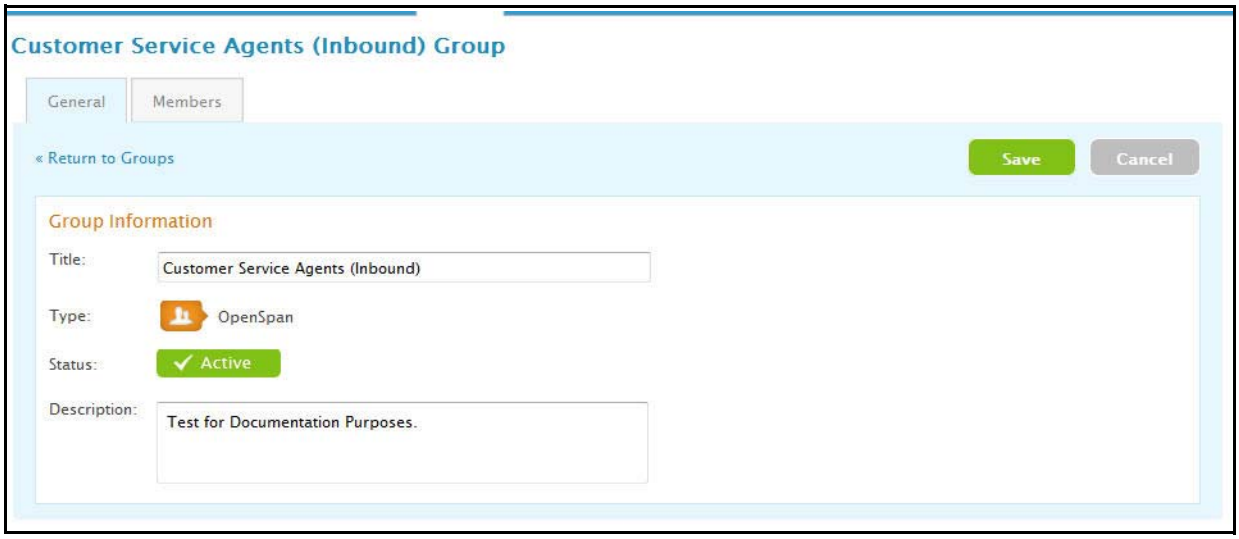
b. Enter a Group name and a description.



c. Click **Save**. The created Group screen displays.

- Title: Name of the Group (editable).
- Type: Yellow = Locally created (not imported from OpenDirectory); Red = From Active Directory, Status Unknown; Green = Active Directory/Valid. See “Adding Users” on page 4-7.
- Status: Green = Active. Red = Inactive.

d. The Group is now available for assigning Users. Select the **Members** tab. Current Users assigned to the Group display. Or, to add the User to a Group, follow the procedures under “Assigning Users to Groups” on page 4-17.



Create New — Import Active Directory Group

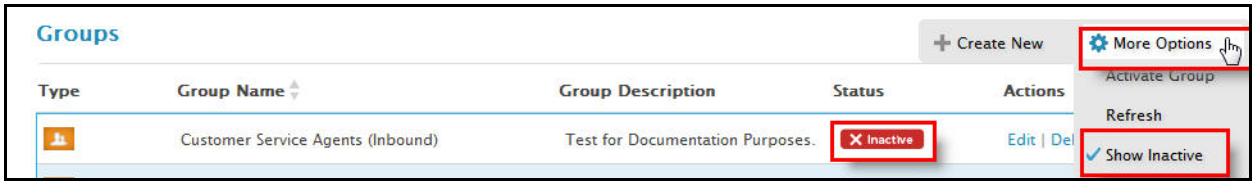
To import Groups from Active Directory, first establish a connection between OpenSpan Server and the Active Directory source. See “Site Settings” on page 4-44. The procedure for import is very similar to that of importing Users. See page 4-10.

Delete, Deactivate, and Reactivate Groups

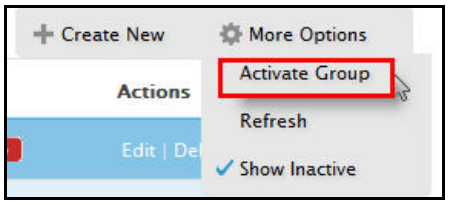
Two options exist for removing Groups: **Delete**, which is permanent, and **Deactivate**, which retains the User account but removes it from active processes on OpenSpan Server. The steps below explain how to perform deletions, deactivations, and how to reactivate a User.

1. Choosing **Delete** from the Group view allows three possibilities:
   - Deactivate — Keep the Group, but remove it from active service.
   - Delete — Remove the Group and all its attributes from the OpenSpan Server database.
   - Cancel — Return to the Group screen.

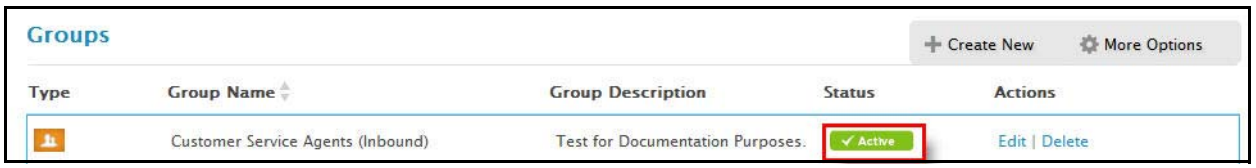
2. If you Deactivate, and wish to Reactivate the Group, then use the **More Options** menu and select **Show Inactive**. The Inactive Groups are indicated under the Status column.



3. To Activate the Group, click the Group in the display to select, then from More Options choose **Activate Group**.



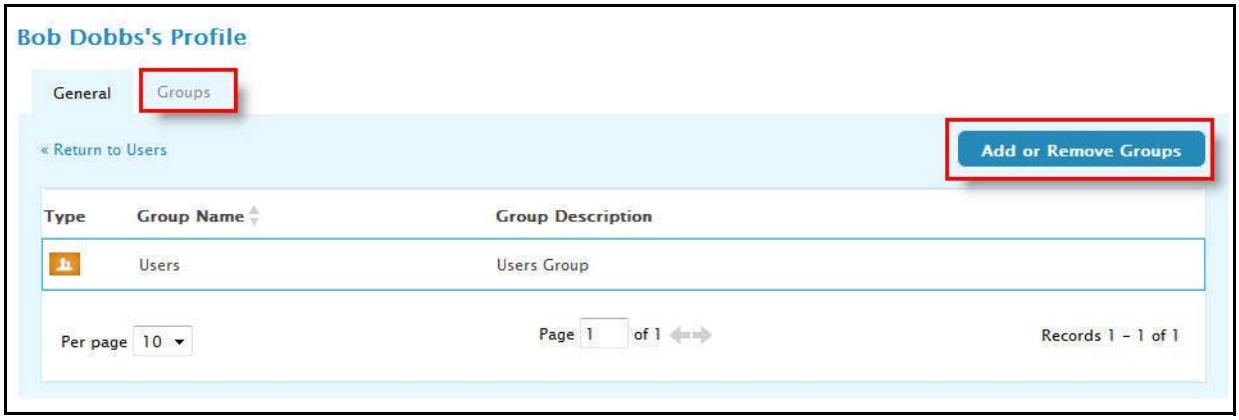
4. The Group's Status is now Active.



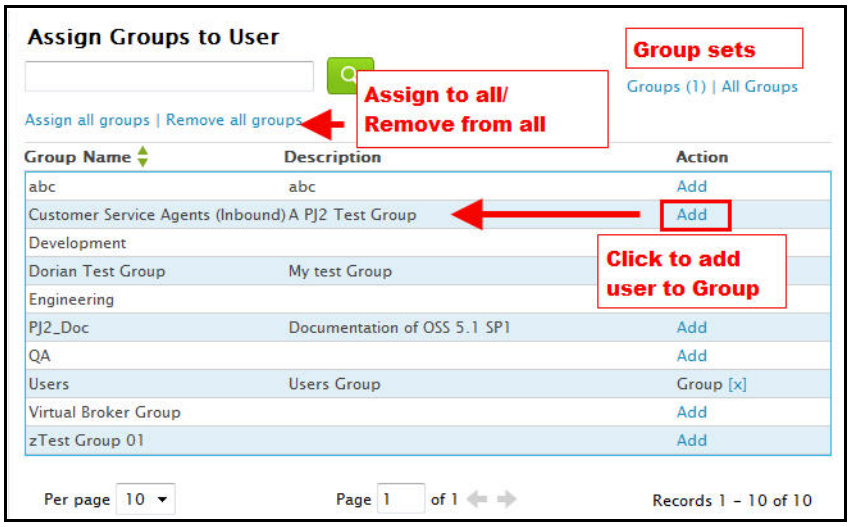
Assigning Users to Groups

Once at least one User and one Group are created, the next step is to assign a new User to a Group.

1. If you just added a User, then you are displaying the User's Profile. If you have exited the Profile screen return by choosing **Setup | Users**. The User Accounts screen displays. Select the User, and **Edit**.
2. From the User's Profile screen, select the **Groups** tab, and click **Add or Remove Groups**.

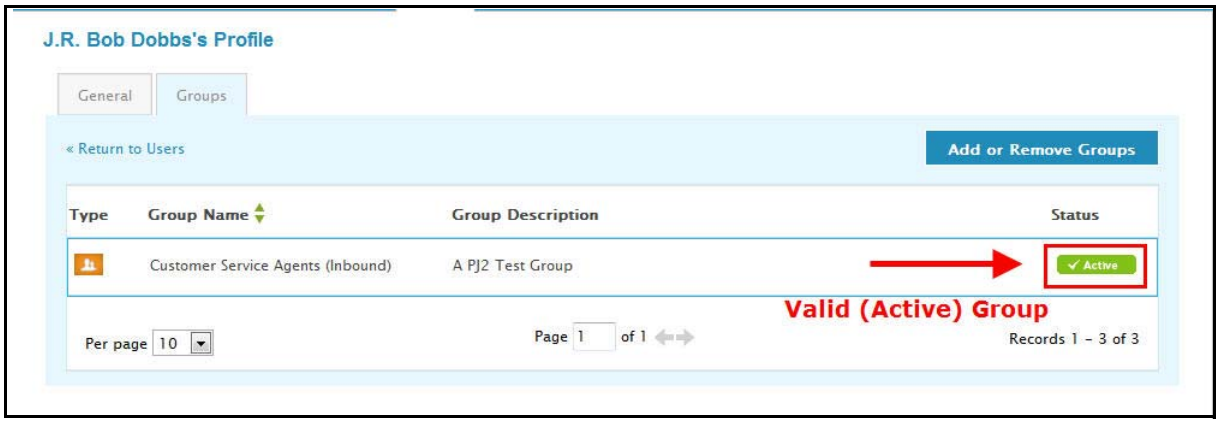


3. The Assign Groups window displays.
   - Click Add to add the user to a specific Group.
   - Quickly assign to all Groups, or Remove with one click.
   - Sets of Groups can be selected from this window for assignment.



4. The User is now assigned to the new Group (Customer Service Agents [Inbound]). To remove this user from the Group, simply click the linked **(X)**.

5. Click **OK**. This returns you to the User's Profile screen. Note that the User is now assigned on the Profile screen to the Customer Service Agents (Inbound) Group. The Group's description also displays for reference. A prominent icon displays confirming that the Group is active. If inactive, see "Delete, Deactivate, and Reactivate Groups" on page 4-16.



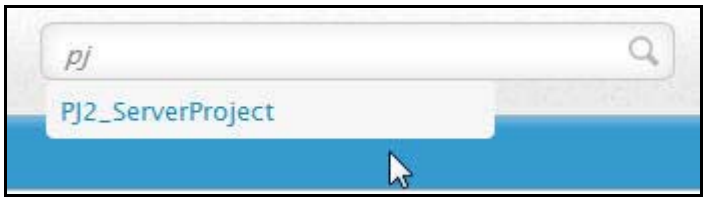
Managing Deployment Packages

OpenSpan solutions developed in OpenSpan Studio 5.2 are configured (with specific runtime parameters), if appropriate, and deployed via direct upload to OpenSpan Server. There, they can be accessed by registered OpenSpan Server users and modified, promoted, and assigned to Groups that were set up in the section “Adding Groups” on page 4-14.

Users assigned to those Groups (see “Assigning Users to Groups” on page 4-17) have login credentials to desktop workstations equipped with OpenSpan Runtime 5.2, a licensed small-client application. Depending on how OpenSpan Runtime is configured, a User logging in to a Runtime-equipped workstation automatically downloads a runtime package of the original OpenSpan solution that has made its way from OpenSpan Studio to OpenSpan Server where it has likely been customized for each User Group. **Note**: Runtime Users may only be assigned events reporting or use or the Process Toolbar. See Chapter 5, *OpenSpan Runtime 5.2 Configuration* for more information.

Tip: Use the Search box

Find the Package you’re looking for in less than a second using a string prompt in the Search box, then click the appropriate result.



You go through the database table of available packages to the correct one.

Adding, Editing, Deleting Features

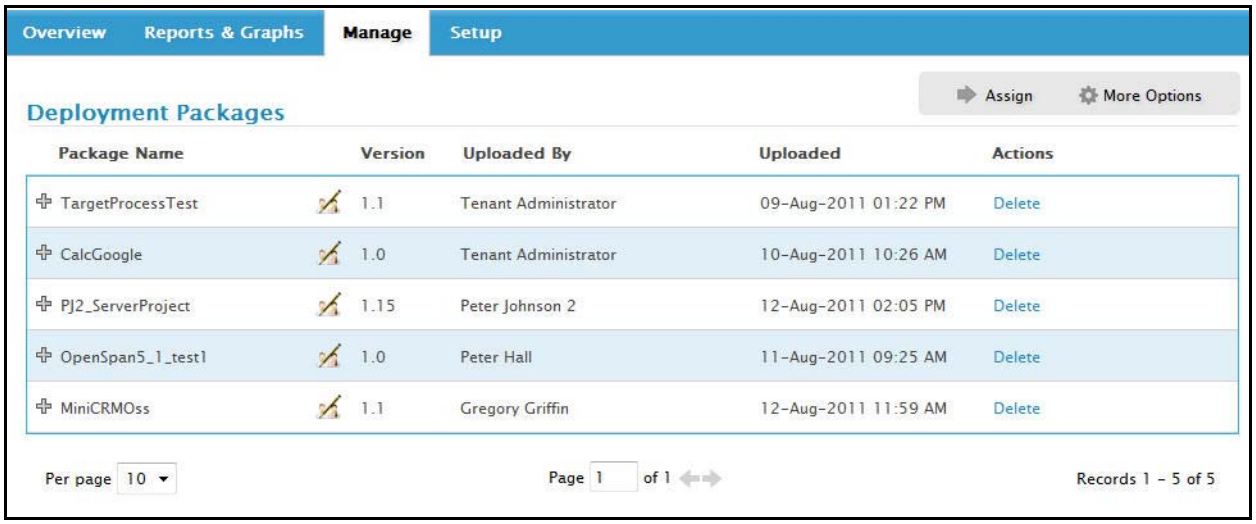
After a deployment package is uploaded to OpenSpan Server from OpenSpan Studio, a server user with appropriate credentials can assign Features that follow the package through download to individual user desktops based on User and Group assignments.

There are two categories of Features available in OpenSpan Server 5.2:

- **Default Values**. Certain categories of interrogated controls can be set with default values. Examples include text fields that can be set to contain specific text, masking properties, or are automatically blank on application open. Other examples might be properties with boolean radio buttons (On/Off, Yes/No) or similar selectable values.
- **Pop-up Help**. This feature allows you to set pop-up balloons with titles, text, a informational or custom icons. When a user selects a control that accommodates pop-up help, the Feature displays. Examples include items requiring specific process guidance, especially when the same control must be explained in different languages.

General

1. Access the main package directory via **Manage | Deployment Packages**. The Deployment Packages window displays. Expand a package by clicking the Plus (+) icon to the left of the package name.

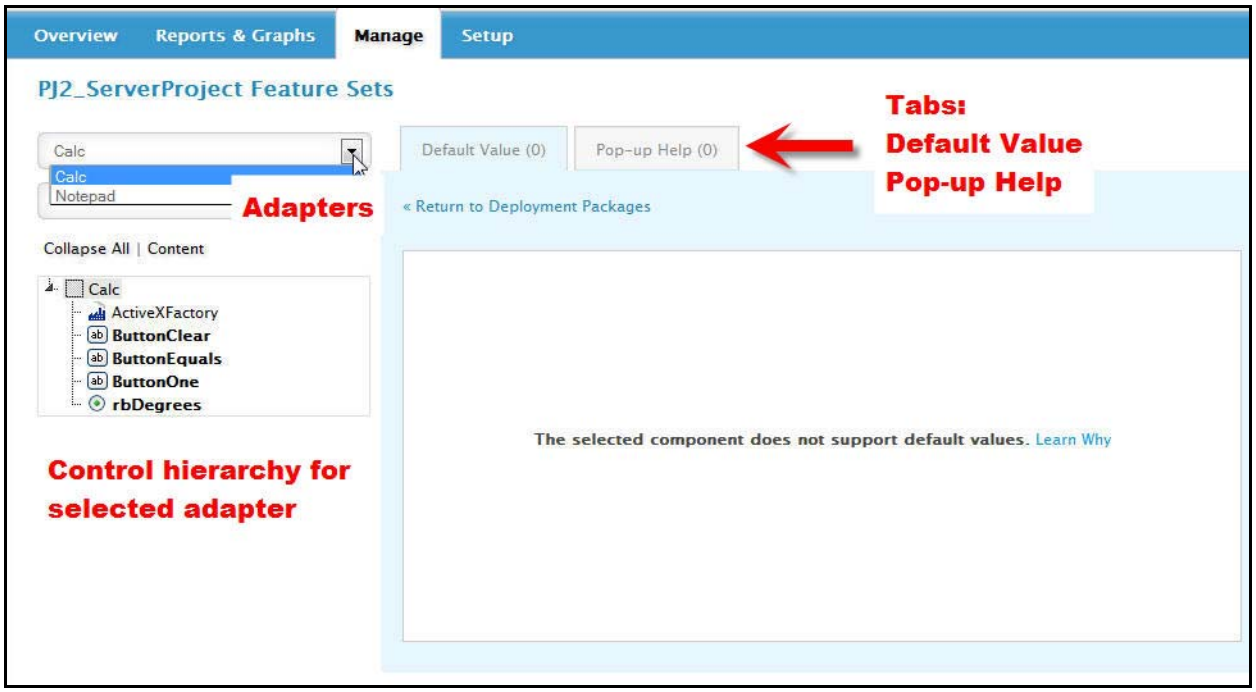


This displays the Feature Set Levels for that package along with the ability to Edit or Delete Features.

2. Select **Edit**. In this example, we'll modify the Feature Set Level assigned to Development. You can also modify Production, if the package has been promoted, though this is not recommended as saving may cause unwanted replacements of package features. For more, read about package promotion under the section "Assigning Feature Set Levels" on page 4-33.

3. The Feature Sets window displays. The main parts are:
   - Drop-down list box of adapters included in the OpenSpan runtime deployment package
   - An expandable hierarchy (tree view) of interrogated controls for each adapter.
   - Tabs for adding or changing the Default Values, if appropriate, or adding Pop-up Help to a control

4. Select an adapter, and its interrogated controls display in a hierarchy.

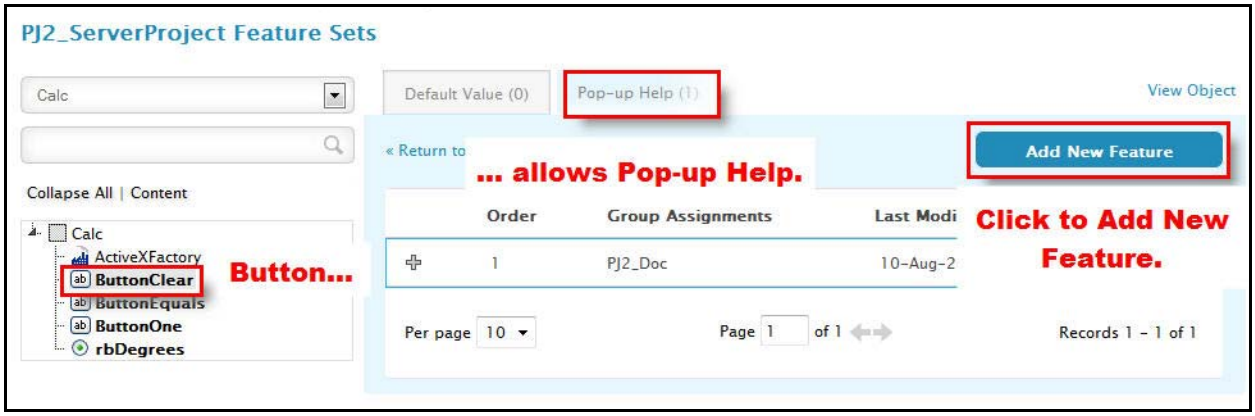


5. Select a control from the hierarchy, then select a tab to add a Default Value or Pop-up Help. If the Feature isn't appropriate for the control, an information message appears with a link to explain why. For example, a clickable button won't normally accept a Default Value — the value is whatever string resource comes with the control, such as a numbered calculator button.

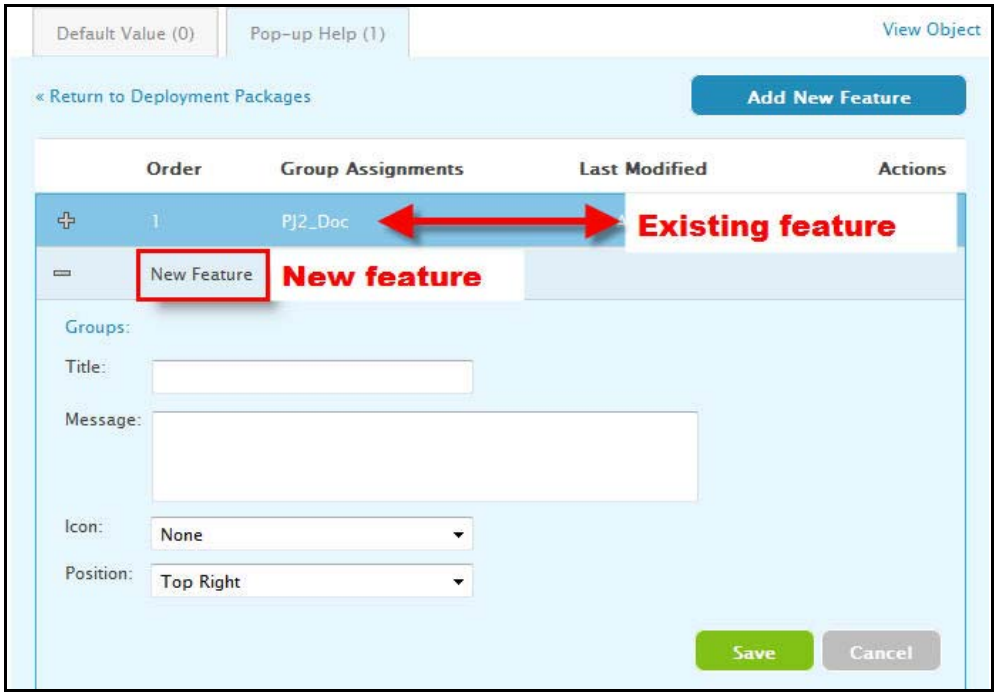
Adding Pop-up Help

Selecting a button or similar control does, however, allow you to add Pop-up Help.

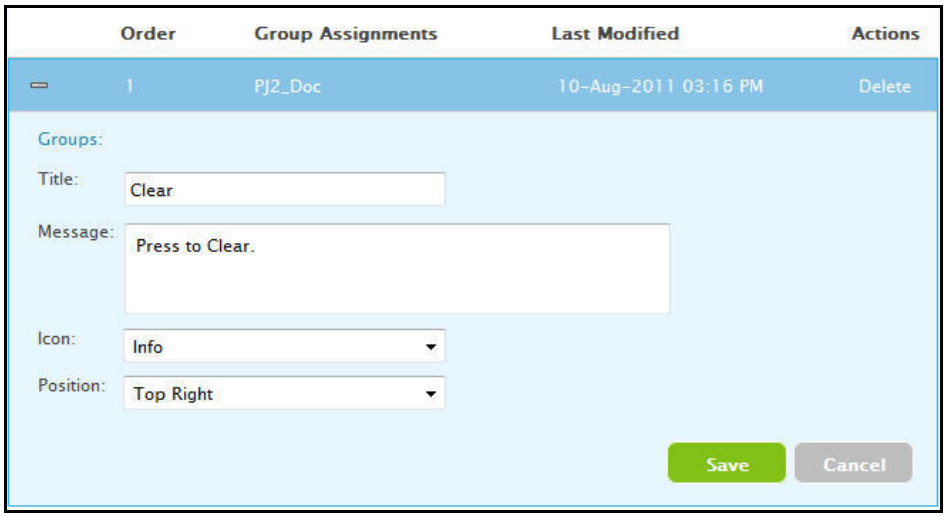
1. Select a control from the hierarchy.
2. Select the **Pop-up Help** tab.
3. Click **Add New Feature** to continue.



If there are existing Pop-up Help Features, the new feature working area displays under them.

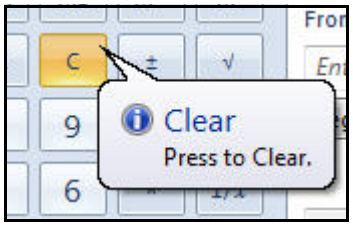


4. Assign the following properties for Pop-up Help:
   - Title — Appears at the top of the Pop-up
   - Message — Text for the user
   - Icon — Choose None, Info, Warning, or Error, depending on the context of the message
   - Position — Location in the Pop-up Help dialog bubble: Top Right, Bottom Left, Bottom Right



5. Click **Save**. The new Feature appears in the Features list.

How will this feature appear on the Runtime User's desktop? The User clicks the button and the Pop-up displays.

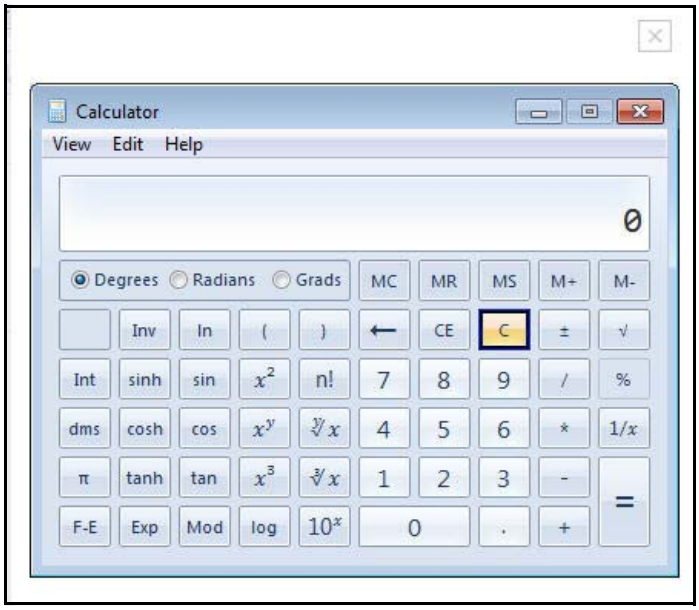


Tip: View Screenshot

OpenSpan provides visual confirmation of the controls you use as Feature Set attachment points for an interrogated application. With the control selected, click **Screenshot** in the upper right to see an image of the interrogated control.

The image capture follows the package from OpenSpan Studio to OpenSpan Server and is a handy reference when adding features to potentially hundreds of controls across an enterprise.

This illustration shows the **screenshot** accompanying the **Clear** button control in the deployment package from OpenSpan Studio to OpenSpan Server.



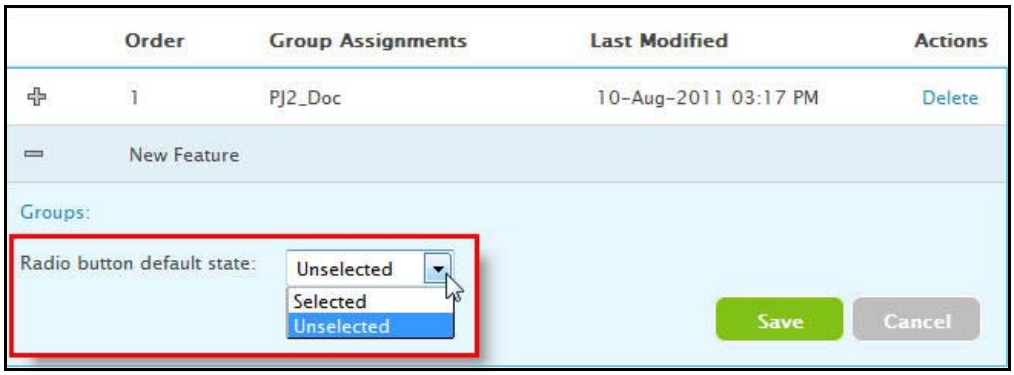
Adding Default Values

Next, let's see an example of adding a default value to a control.

Boxes and Buttons

Begin with the same steps as in "Adding Pop-up Help" on page 4-24.

1. Select a control from the hierarchy.
2. This time, select the **Default Values** tab.
3. Click **Add New Feature** to continue. A different window displays for adding default values.
4. Check- or radio-box controls allow **Checked** or **Unchecked**.



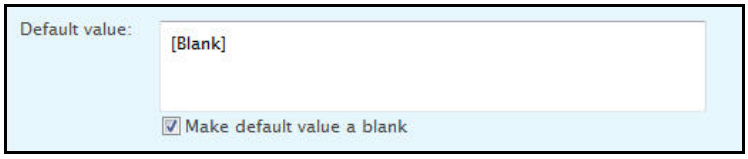
Text Controls

Text controls are a different story and automatically are available if the control you choose is a text field. These allow flexibility of default values for business purposes such as:

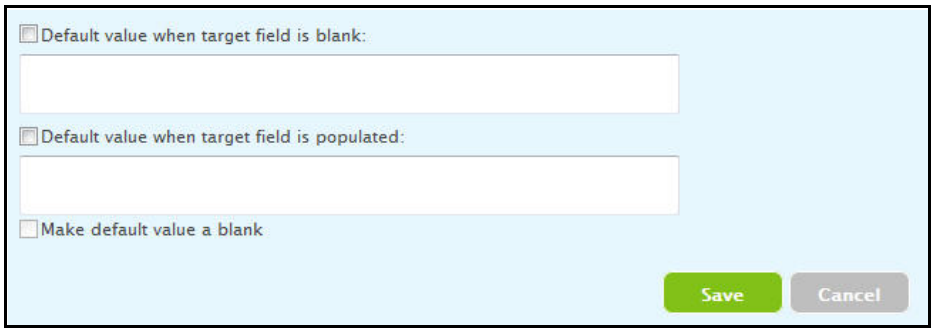
- Clearing a field (making it blank)
- Automatically populating a field with conditional information
- Masking sensitive data
- Completely blanking sensitive data

Note: Selecting the text Use different default values for populated and blank fields toggles the text options.

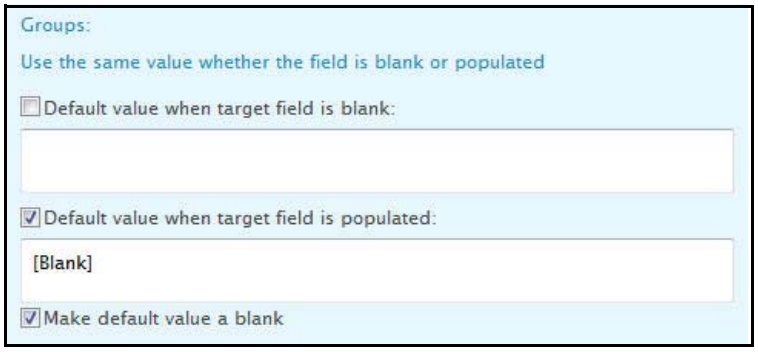
1. For text controls, the New Feature window automatically opens a Default Value window for free entry. If you check **Make default value a blank**, the place-holder [Blank] populates the entry area.



2. To exercise other options, select **Use different default values for populated and blank fields**.
3. Check one of these:
   a. Default value when target field is blank. Then add the text you wish to display.
   b. Default value when target field is populated. The text you enter here will replace whatever is in the field when the control is initially populated with any value.



   c. Or — Selection (b) above enables the Make default value a blank check box. Selecting this inserts that [Blank] placeholder.



Assigning Feature Sets to Groups

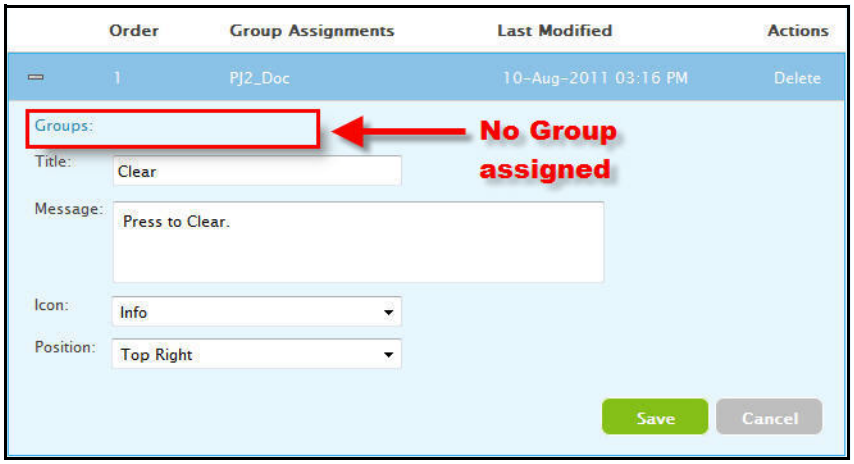
General

Note: If you attempt to Save a feature without assigning a Group, you will see a prompt saying **At least one group must be assigned**.

Steps

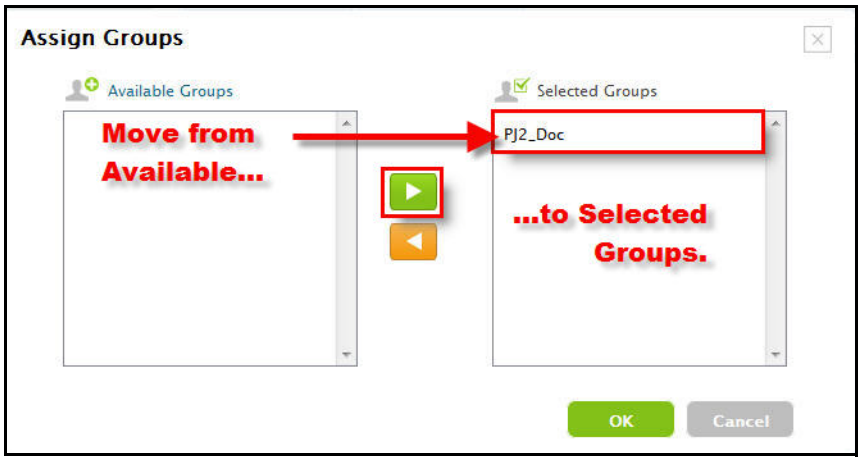
The beginning state for these steps is:

- A Feature has been added
- There is no Group assigned to the Feature



1. Click the highlighted **Groups:** text on the Feature window. The Assign Groups panel appears.
2. Select a **Group** from the Available Groups list. **Tip**: Control-click or Shift-click allows you to select multiple Groups from the Available Groups column.

3. Click the right-arrow move control to transfer the Group(s) to the Selected Groups column.

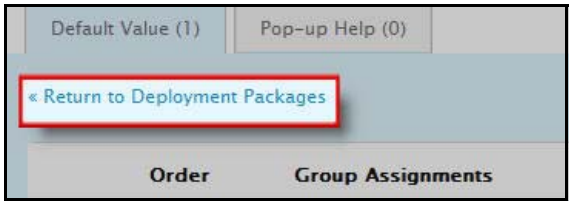


The Group name now appears under Selected Groups. The Feature is now assigned to all instances of the parent runtime package being downloaded by Users assigned to this Group.

Note: It is possible to assign the Feature to multiple Groups.

4. Click **OK**. This returns you to the New Feature screen.

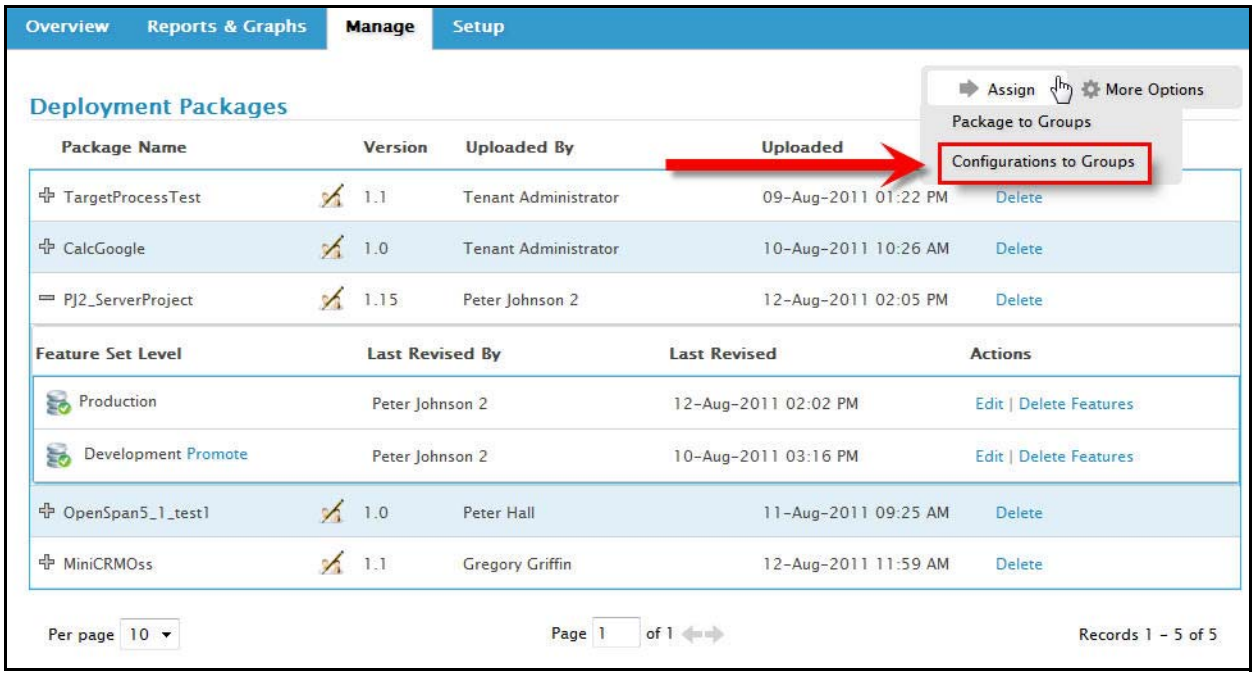
5. Click **Save** to create the new Feature. Select the highlighted screen text to **Return to Deployment Packages**.



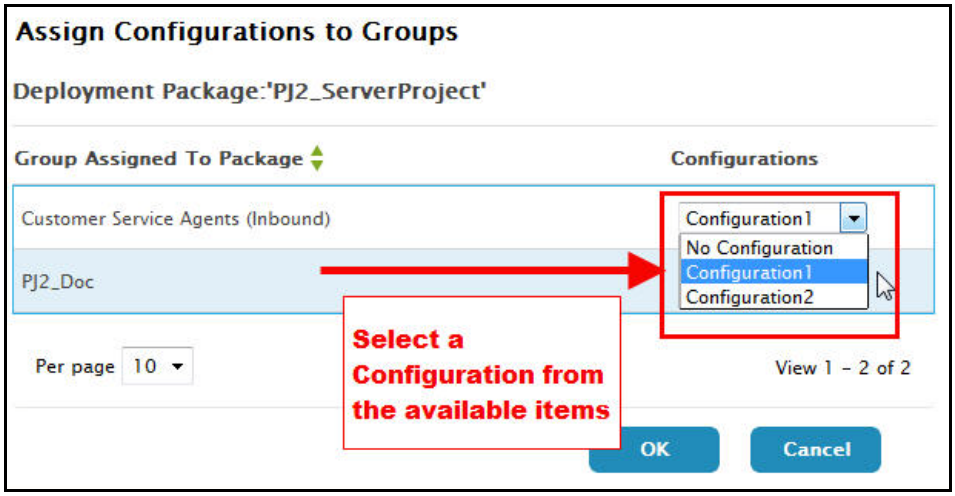
Assigning Configurations to Groups

OpenSpan solutions sometimes contain components called Configurations. These are custom sets of properties for specific adapters. OpenSpan Studio 5.2 now has the ability to synchronize different attributes such as Configurations with the assignments made on OpenSpan Server. For more information on that, see “Project Configurations” on page 3-6 of Chapter 3, *OpenSpan Studio 5.2 Configuration*.

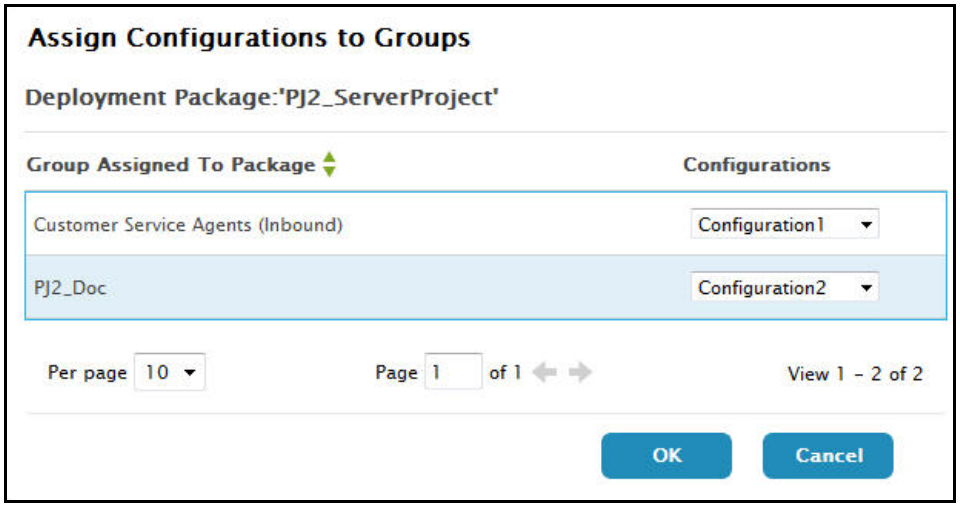
1. Choose **Assign | Configurations to Groups**.



2. A dialog box displays allowing you to select a configuration for each Package/Group assignment from the set of available Configurations added in OpenSpan Studio.



3. Assign the appropriate Configurations to Groups, then click **OK**.



4. The Configurations are now available when their Runtime packages are downloaded by the respective Users who are members of the Groups assigned.

Assigning Feature Set Levels

OpenSpan Server 5.2 lets you work on a package in "sandbox" mode in the Development level, then promote it to Production status. This makes it the version that is downloaded when Runtime Users assigned to the appropriate Groups connect from the desktop.

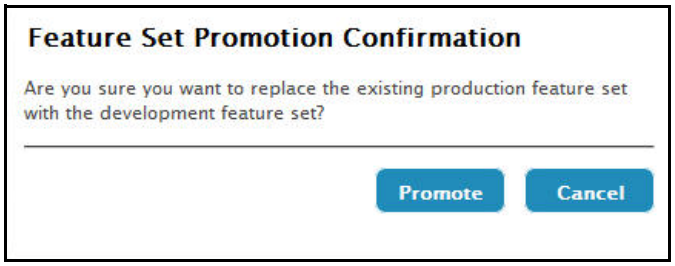
1. From **Manage | Deployment Packages**, expand the details of a deployment package by clicking the Plus (+) icon.
2. When you have finished all development work on the package and wish to promote to Production, click the highlighted screen text **Promote**.

OpenSpan Studio 5.2 allows developers to work on the package in either Development or Production status by making a toolbar selection. See "Promotion Level" on page 3-8 of Chapter 3, *OpenSpan Studio 5.2 Configuration*.

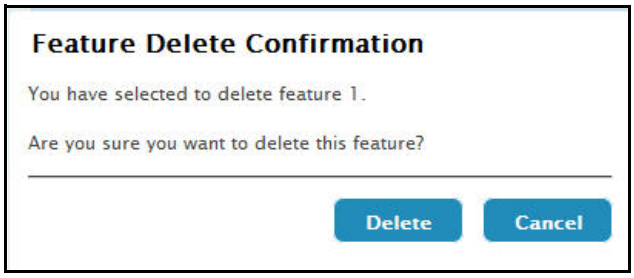
Notes on Feature Set Promotion

Exercise caution when promoting Feature Sets from Development to Production, or changing Feature Sets in Production. Doing this without careful planning can cause great confusion and possibly derail an otherwise excellent matrix of downloaded Runtime, user-focused packages.

- Promotion Immediately replaces the Feature Set in Production with the Feature Set in Development. **This can not be undone.** A dialog box asks you to confirm the choice.



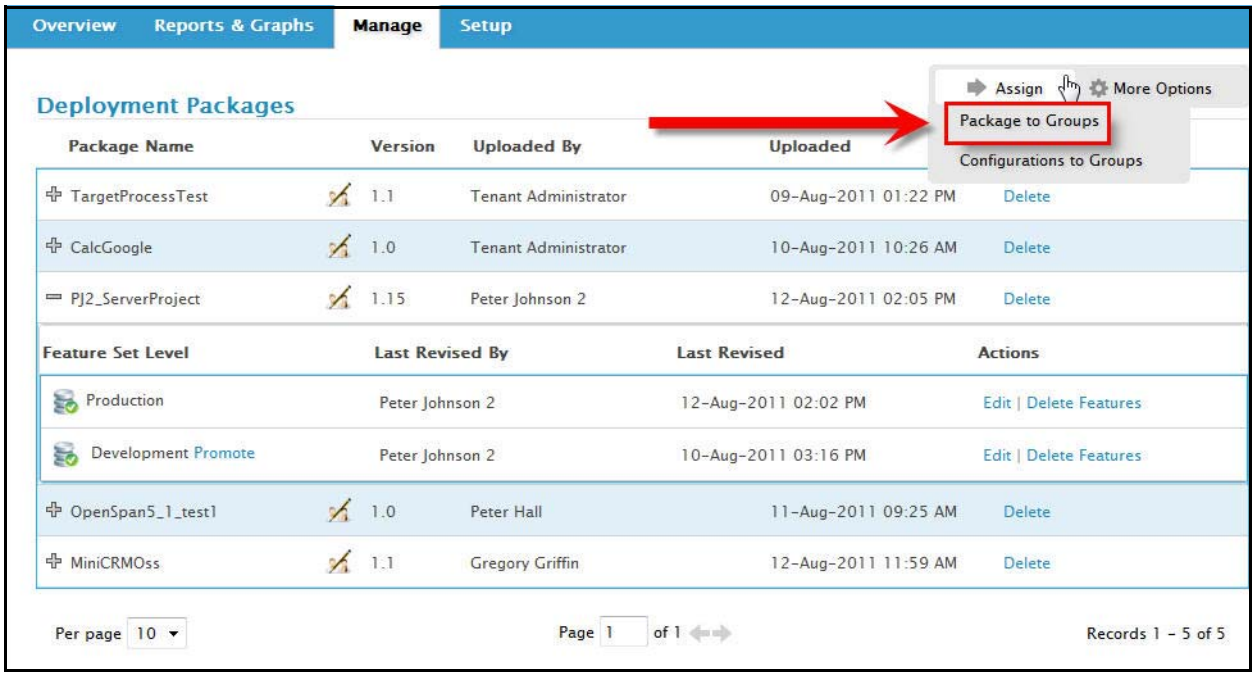
- Likewise, however, if you delete Features from Production, the Development version of the package will "sync up" and the Feature will also be lost from Development. A dialog box asks you for confirmation.



Assigning Groups to Packages

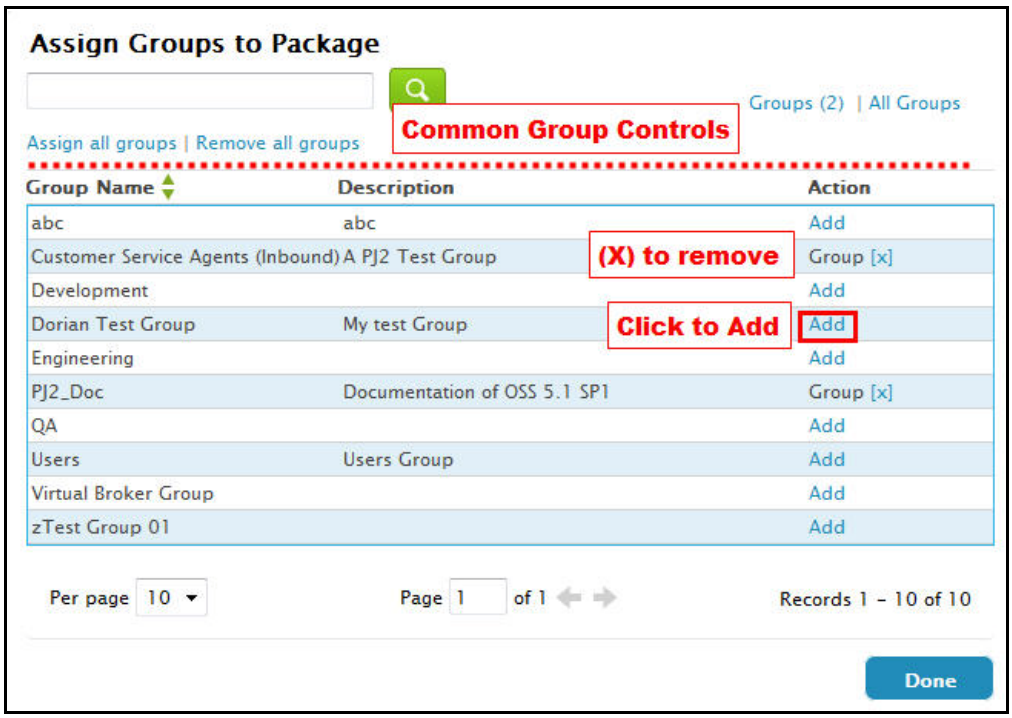
Group assignment for packages works in a similar same manner as the previous topics.

1. From **Manage | Deployment Packages**, select a package by clicking to highlight.
2. Choose **Assign | Packages to Groups**.



3. A dialog box appears with Group names similar to the one in "Assigning Feature Sets to Groups" on page 4-29.

4. Assign the appropriate Groups, using the in-window controls. **Add**, **(X)** to remove. Other Group controls function the same as in other Group selection windows.



OpenSpan Studio 5.2 also allows developers to choose the same Group package assignment to ensure the proper variant of the solution is being changed. See “Run as Group” on page 3-7 in Chapter 3, *OpenSpan Studio 5.2 Configuration*.

Managing Processes

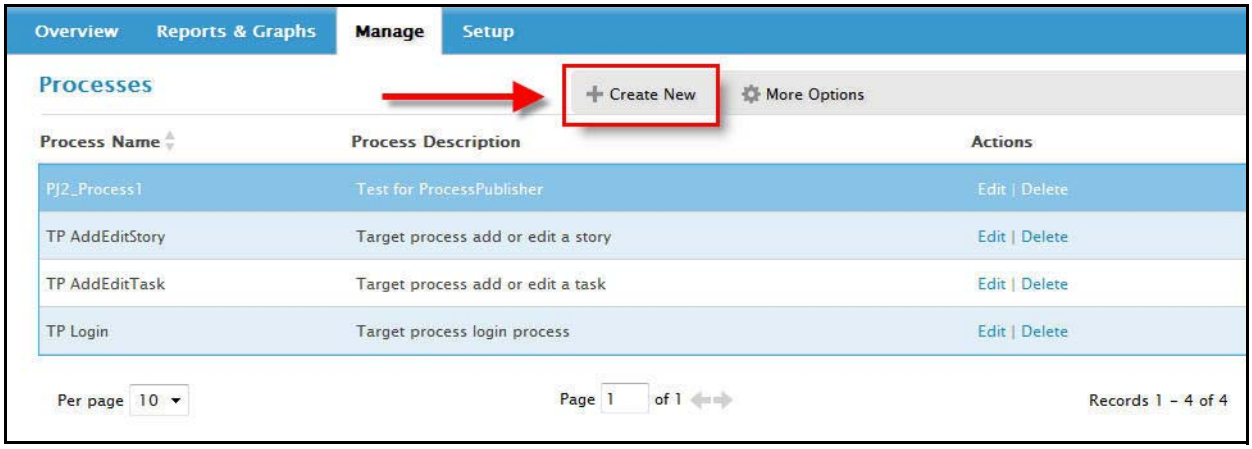
Processes are named and given metadata to supply another level of Events reporting to the OS_Server database's Events table. Once created on OpenSpan Server, the metadata can be used in OpenSpan Studio with bindings from the ProcessPublisher to move data from OpenSpan Runtime desktop solutions to the database. See "Producing Process Automations" on page 3-9 of Chapter 3, *OpenSpan Studio 5.2 Configuration*.

Additionally, based on assignments, the processes populate the drop-down menu on the Runtime Process Toolbar. See "Process Toolbar" on page 5-7 of Chapter 5, *OpenSpan Runtime 5.2 Configuration*.

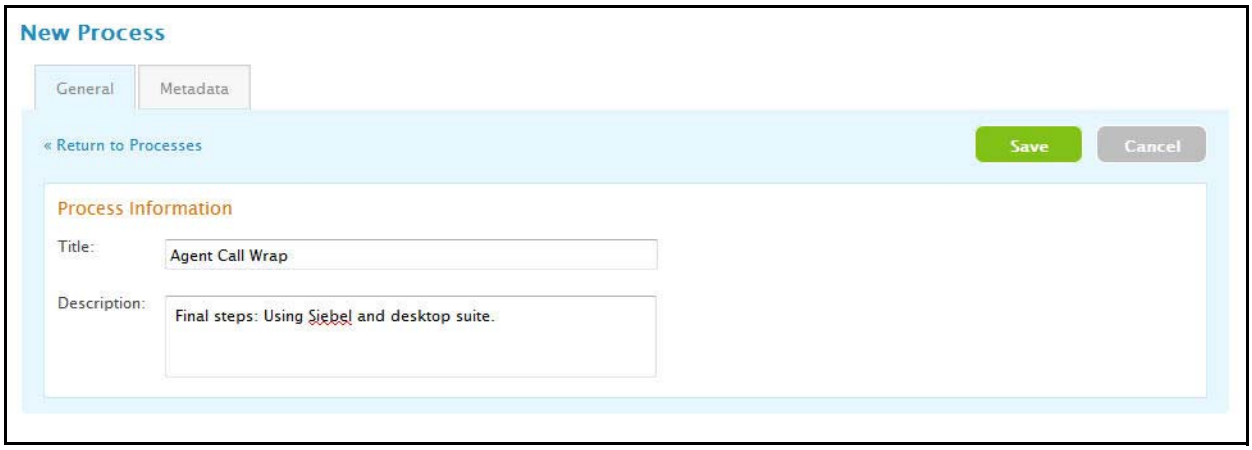
- Create New.
- Edit. Accessed under **Manage | Processes** in the **Actions** column.
- Delete. Deletion offers the options to permanently delete or Deactivate. If you choose to Deactivate, the same menu controls allow you to Activate at a later date. Use **Manage | Processes | More Options**.

Creating a New Process

1. Create a New Process. From **Manage | Processes**, choose **Create New**.



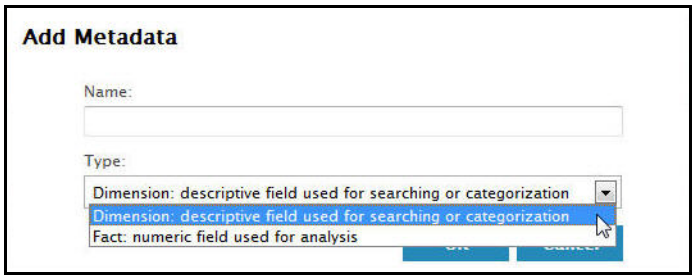
2. The **New Processes | General** tab displays. Enter a Title and Description.



3. Click **Save**.

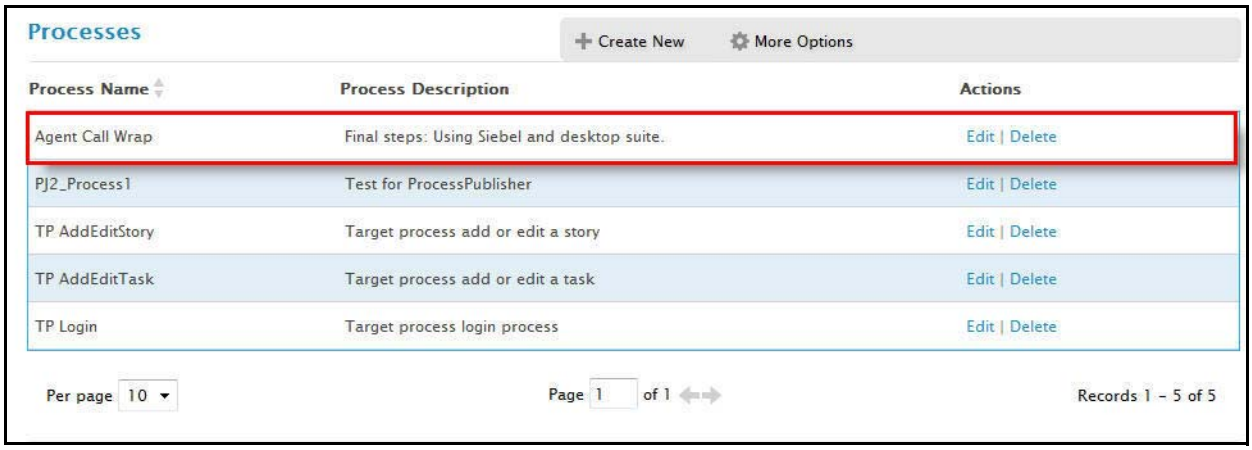
4. Select the **Metadata** tab. Click **Add Metadata**.

5. In the Add Metadata window, name the metadata (no spaces allowed) and select as **Dimension** or **Fact**.



6. When finished, click **OK**.

7. The new Process displays in the list.



If metadata is deleted, **Other Options** include:

- Refresh
- Reactivate
- Show Inactive

...for previously deleted Metadata.

Editing, Deleting a Process

Editing and Deleting work the same way that other major items are managed in OpenSpan Server 5.2. **Edit** accesses the data for changes. Delete allows full delete or Deactivation. Use **More Options** to Activate.

Managing Group Runtime Settings

General

When you have:

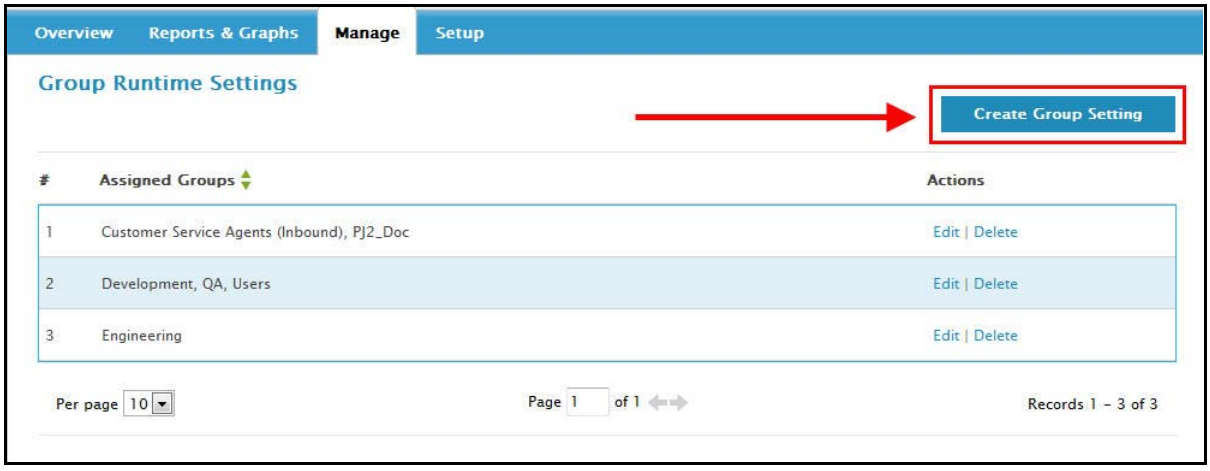
- Created Users
- Created Groups
- Assigned Users to Groups
- Made Package/Feature other Group assignments

... it's time to configure Group Runtime Settings.

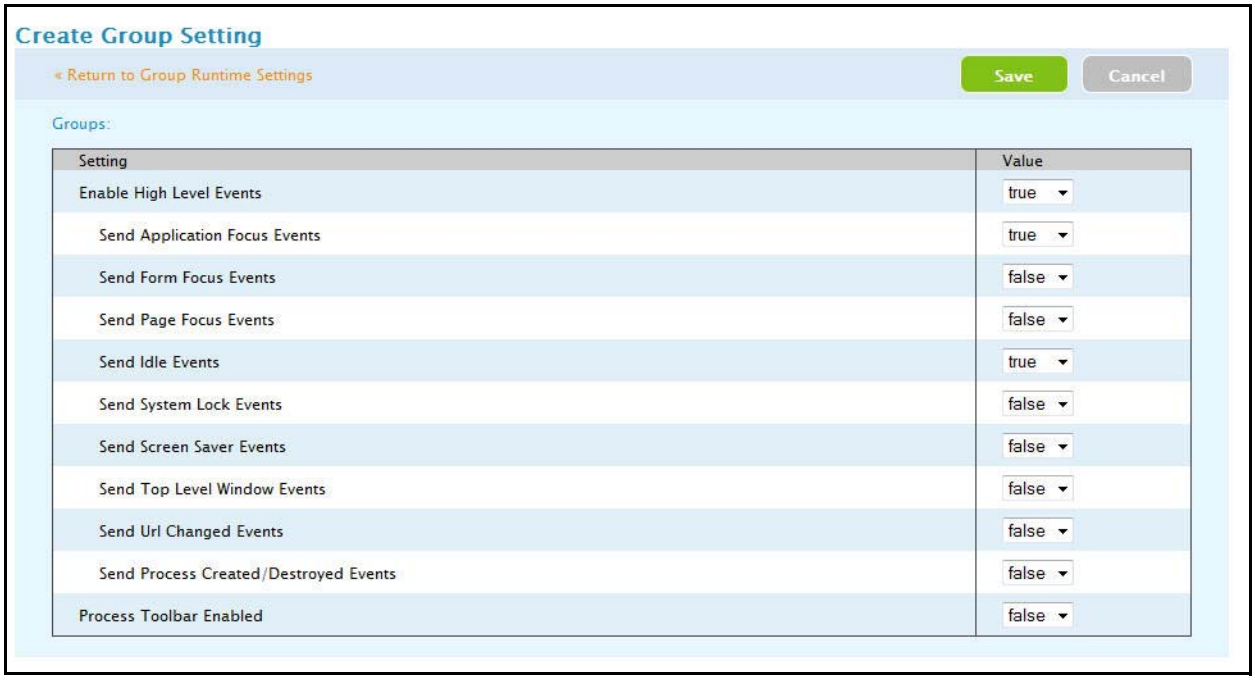
Create Group Runtime Settings

This set of instructions assumes you have created at least one Group in OpenSpan Server 5.2. If not, see "Adding Groups" on page 4-14.

1. Select **Manage | Group Runtime Settings**. The Group Runtime Settings screen displays.
2. Click **Create Group Setting**.



The Create Group Setting screen displays.

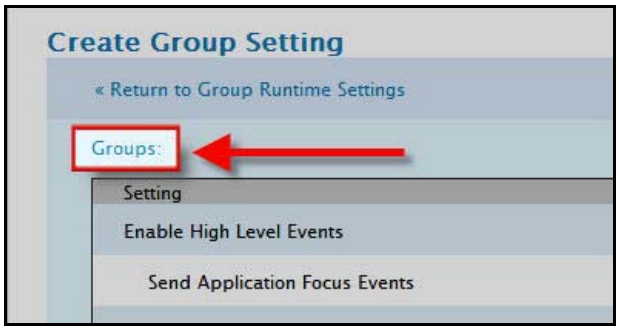


Group Settings are displayed in table form with drop-down list boxes for each setting. Select the following values:

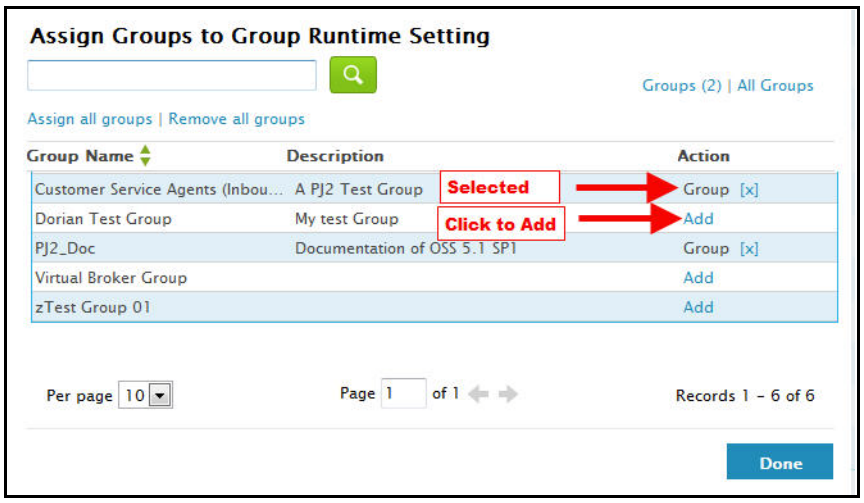
Setting	Value
Enable High Level Events	Default is **True**. Enables the list of High-Level Events in the categories below.
Send Application Focus Events	Default is **False**.
Send Form Focus Events	Default is **False**.
Send Page Focus Events	Default is **False**.
Send Idle Events	Default is **True**.
Send System Lock Events	Default is **False**.
Send Screen Saver Events	Default is **False**.
Send Top Level Window Events	Default is **False**.
Send Url Changed Events	Default is **False**.
Send Process Created/Destroyed Events	Default is **False**.
Process Toolbar Enabled	Default is **False**. However, many Runtime users will need this component assigned. See “Process Toolbar” on page 5-7 in Chapter 5, *OpenSpan Runtime 5.2 Configuration*.

More in-depth information on High-Level Events is available in the OpenSpan online help.

3. After selecting settings on the Group Runtime Settings screen, click **Save**.
4. Next, attach the list of settings to a Group. Click the **Groups** highlighted text on the Create Group Setting screen.



5. The Assign Groups window displays. Select one or more of the **Available Groups** and assign by clicking **Add**. Remove by clicking the **(X)**. You can also **Assign** or **Remove all groups**.

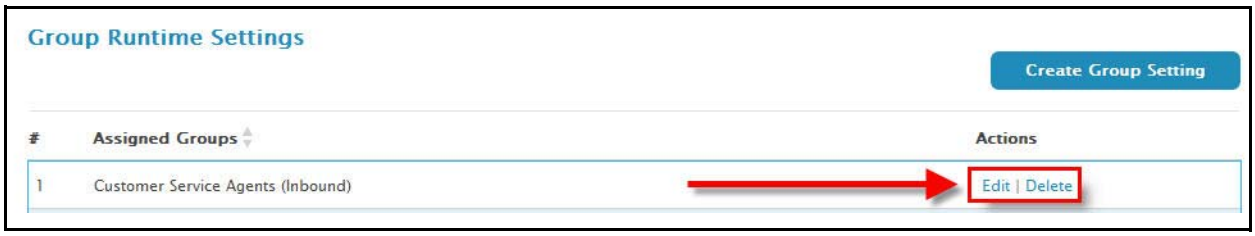


The name(s) of the assigned Group(s) now displays on the Create Group Setting screen. You can change this at anytime by clicking the Groups text and repeating the steps above.

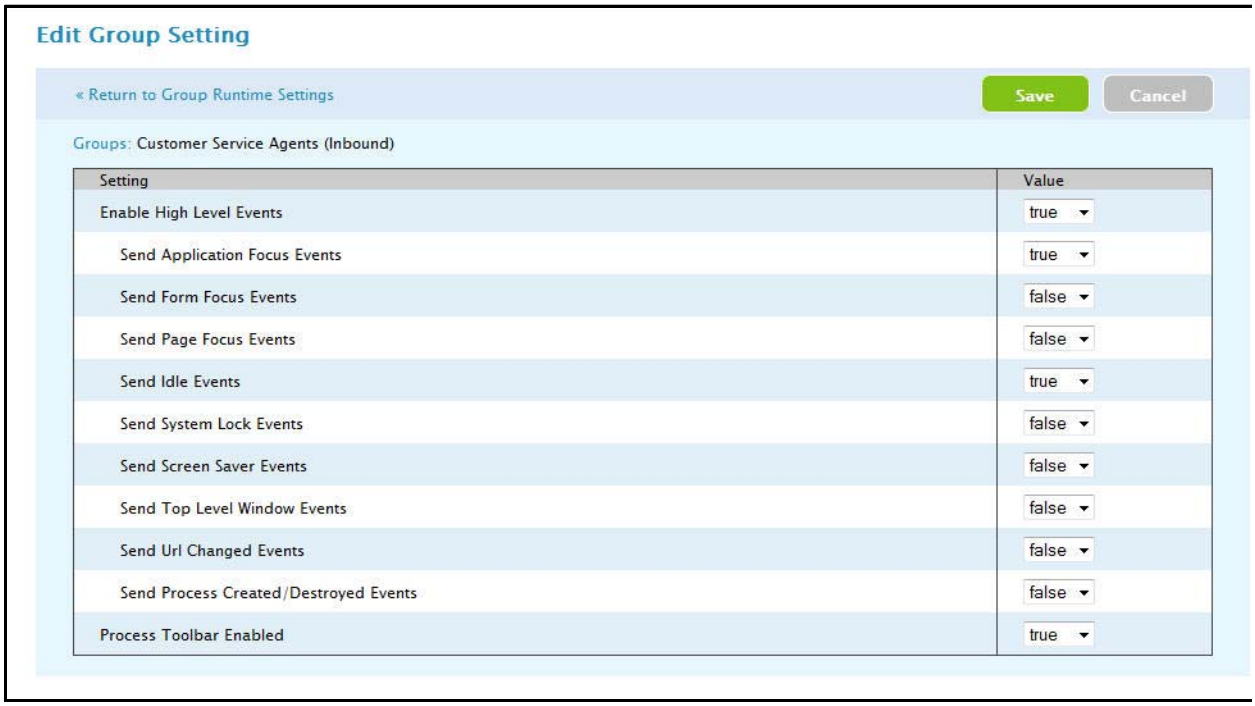
Edit and Delete Group Runtime Settings

Edit

1. From the Group Runtime Settings screen, click **Edit**.



2. The Edit Group Setting screen displays. This is a copy of Create Group Setting. Reset properties as appropriate, then click **Save**.

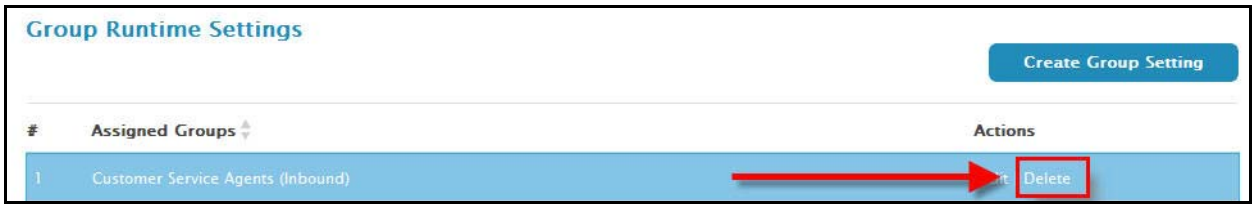


3. It's possible to add or change the assigned Groups at this point. See "Create Group Runtime Settings" on page 4-39.

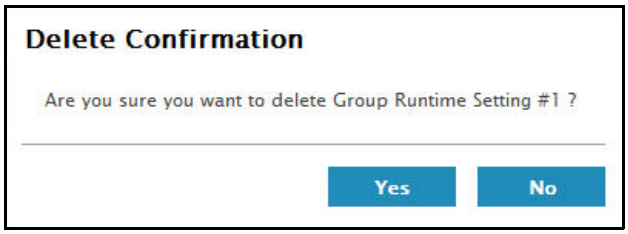
Delete

Deleting Group Runtime Settings is permanent. Unlike Users and Groups, there is no “soft delete” or Deactivate option.

1. Access **Manage| Group Runtime Settings**.
2. Select an **Assigned Group** by single-clicking the row.
3. Click **Delete**.



4. The Delete dialog displays.



5. Click **Delete** to permanently remove the Group Settings, or **Cancel**.

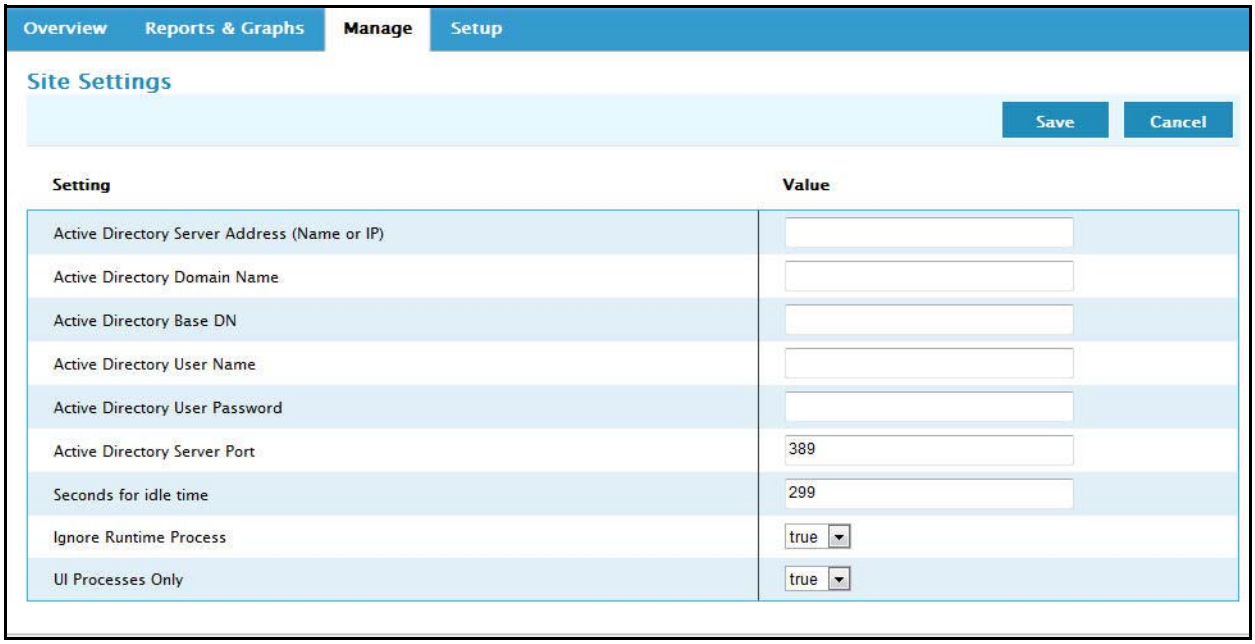
Site Settings

General

Initial setup and configuration of OpenSpan Server 5.2 requires several more properties to be set by the server Administrator. These properties affect interaction with Active Directory (used for some User and Group sourcing and administration) and activation of High-Level Events for collection by desktop Runtime packages and storage and reporting on OpenSpan Server.

User Steps

1. Access **Manage | Site Settings**. The Site Settings window displays.



Most of the values are manually entered, except for two drop-down list boxes.

TABLE 1. Site Settings

Setting	Value
Active Directory Server Address (Name or IP)	The name or IP address of the source Active Directory server supplying User or Group names, if applicable.
Active Directory Domain Name	Domain name of the same Active Directory server from the previous step.
Active Directory Base DN	LDAP (light weight directory protocol) connection string for connecting with AD to read AD users/groups. “dc” stands for “domain controller.” Will vary depending on Active Directory source.
Active Directory User Name	Supplied by Active Directory administrator.
Active Directory User Password	Supplied by Active Directory administrator.
Active Directory Server Port	Supplied by Active Directory administrator.
(High Level Events) - Seconds for idle time	Default is 120 seconds. Can be set to any value. **Note**: If set too short, the resulting idle time detections across multiple desktops could create data volume problems and make reports less valuable.
Ignore OpenSpan Runtime Processes	Default is **True**. Setting this value to True prevents self-referential reporting of the Runtime application itself on the user desktop.
Generate High Level Events Only for User Interface Processes	Default is **True**. Setting to True ensures that end-user desktop system messages and events are not sent to the Events table in the OS_Server database or processed for reports.

Chapter 5 OPENSPAN RUNTIME 5.2 CONFIGURATION

In this Chapter

This chapter of the OpenSpan Server 5.2 Administration and User Guide describes how OpenSpan Runtime 5.2 is configured to work with OpenSpan Server 5.2 to download and run OpenSpan solution projects. This chapter is organized into the following topics:

General

Desktop users often don't see or interact much with the OpenSpan Runtime executable application, but rather use the runtime deployment packages that run on the desktop. OpenSpan Runtime is designed to function as unobtrusively as possible, behind-the-scenes, to automate applications and send events to OpenSpan Server for analysis. The information in this publication is meant mostly for developers and business analysts understand how solutions are executed and events collected from the desktop.

OpenSpan Runtime 5.2 has an installation dialog to pre-set the address for OpenSpan Server; most other actions or settings for OpenSpan Runtime 5.2 are either written to the runtime solution package and dynamically enforced or are implemented via a number of server-side settings, such as enabling Events collection, Feature Sets, etc.

OpenSpan Runtime and OpenSpan Server

This section explains the configuration of OpenSpan Runtime 5.2 and how its settings interact with OpenSpan Server, including High-Level Events and runtime packages.

Certificates

For most deployments, workstation certificates are not needed to download and consume runtime solution packages from OpenSpan Server 5.2.

RuntimeConfig.xml

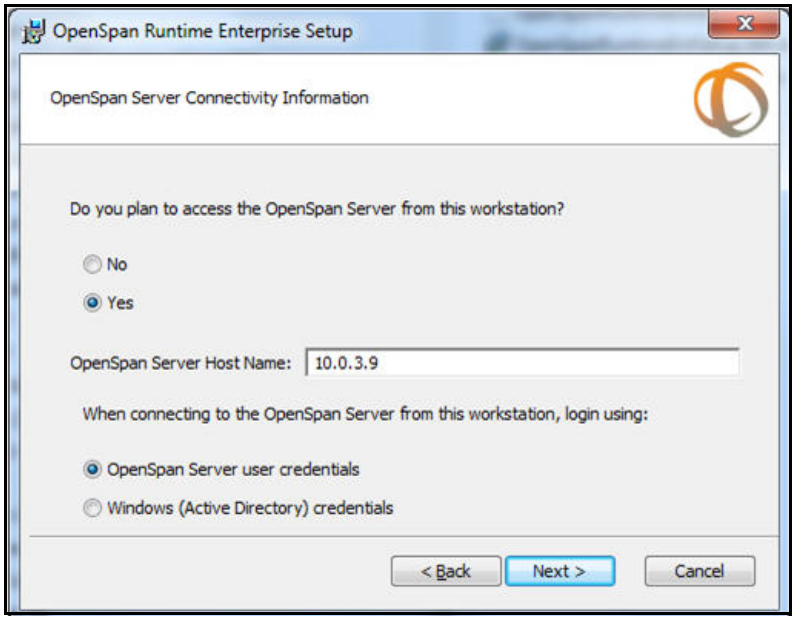
Generally speaking, changes to individual desktop Runtime properties are written to the local RuntimeConfig.xml file, located in the **Program Files | OpenSpan | OpenSpan Runtime** directory. **Note**: OpenSpan Studio writes to a local version of the RuntimeConfig.xml file for development and debugging, located in the **localuser | My Documents | OpenSpan Studio for VS 2010** directory. For a full explanation of the RuntimeConfig.xml file settings, refer to OpenSpan's online help.

Connecting OpenSpan Runtime with OpenSpan Server

When installing OpenSpan Runtime, a dialog box lets you enter the Server connect information. If you choose not to enter the Server connection information at this time, the desktop user will be presented a Connection dialog box when initially running OpenSpan Runtime. This information must be entered or OpenSpan Runtime can not communicate with its Server counterpart.

User Steps

1. During installation, the OpenSpan Runtime Enterprise Setup dialog box displays.



2. Select **Yes** for “Do you plan to access the OpenSpan Server from this workstation?
3. Enter the IP or other form of address for the OpenSpan Server instance that OpenSpan Runtime will communicate with.

   Note: Only use the address or Host Name; do not include prefixes or suffixes such as “http://” or “/”. Doing so will double-write the syntax to OpenSpan Runtime’s RuntimeConfig.xml file and connection will not be possible.

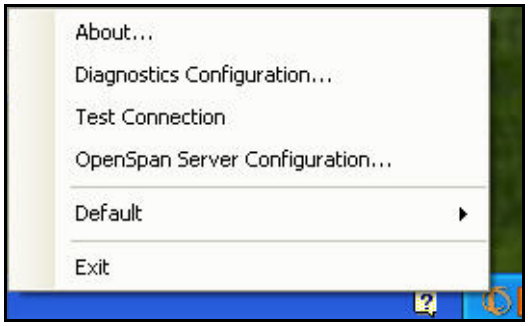
Examples of the RuntimeConfig.xml server connectivity entries are displayed here for your reference. You can modify server addresses in this format as well. The server address corresponds to the test IP address shown in the installation dialog box above. The boolean values are initial defaults. Note that the Server address is written to the URL, baseURL, stsBaseURL entries. Other properties may vary depending on the Windows Communication Foundation (WCF) settings enabled server-side; see the OpenSpan online help.

```
<ServerConfiguration ServerConnectivityEnabled="True">
  <Server serverName="DEFAULT" serverAddress="10.0.3.9" serverURL="http://
10.0.3.9/OpenSpanServer/" serverBaseURL="http://10.0.3.9/"
eventCollectionServerBaseURL="" wastsBaseURL="https://10.0.3.9/Services/
WASTS/" stsBaseURL="https://10.0.3.9/Services/STS/" dnsIdentity="www.QA-
Preprod" useWindowsCredentials="False" authenticateByUserNameOnly="False">
```

4. Next, set the authentication method. If the desktop user will connect to **OpenSpan Server** using **credentials** created locally on the server instance, select "**OpenSpan Server user credentials**." For **Active Directory**, select "**Windows (Active Directory) credentials**."

Runtime Menu

When OpenSpan Runtime loads on the user desktop, by default it displays a small OpenSpan icon in the application portion of the desktop taskbar. Right-click this icon to display the default system tray menu.



Note: You can edit the RuntimeConfig.xml to completely hide the Runtime menu. Reset the defaults as shown here in *bold italic*.

```
<AppSettings>
  <!-- Keys used to hide the runtime system tray icon and define a system
level hot key to toggle its display -->
  <add key="ShowSystemTrayIcon" value="false" />
```

Menu Items and Definitions

About

Select this menu item to check the version number of OpenSpan Runtime.

Diagnostics Configuration

OpenSpan Runtime shares the same rich diagnostic tool set with OpenSpan Studio. Typically, these are only activated and logs generated on advice and guidance from OpenSpan Support, as they can quickly generate overwhelming amounts of data. For setup and details, please reference OpenSpan online help.

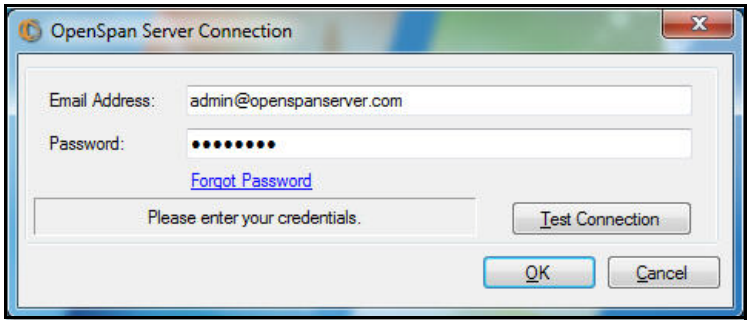
Test Connection

Tests communications with the instance of OpenSpan Server specified in the ServerConfiguration section of OpenSpan Runtime's RuntimeConfig.xml.

OpenSpan Server Configuration

The topic "Connecting OpenSpan Runtime with OpenSpan Server" on page 5-3 above references the Server address installation dialog. If this is left blank, the first time you start OpenSpan Runtime on a desktop, the OpenSpan Server Connection dialog box displays. Enter the credentials for a registered user in your OpenSpan Server hierarchy. Logging off from a workstation and logging a different user on will generate the same process again.

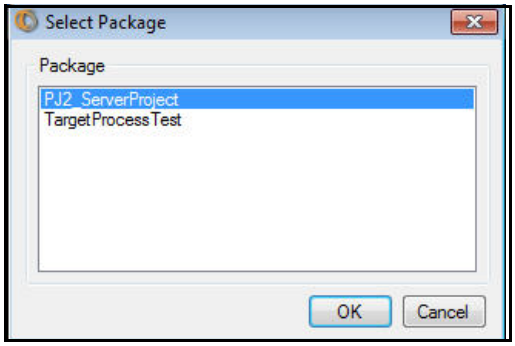
If you wish to change credentials during the same workstation logon session, select OpenSpan Server Configuration from the Runtime menu tray to display the Server Connection dialog box. Enter the appropriate credentials for a registered user on OpenSpan Server. Click Test Connection. When connected, click **OK** to accept or **Cancel**.



Default | (Package)

If no packages are assigned to the Runtime user, none downloads. Runtime will, if configured to, run normally. An example could be using Runtime to only send High-Level Events data to OpenSpan Server. However:

- If a single package is assigned to the user, it downloads on sign-on.
- If the user is a member of multiple groups, a group selection dialog box displays. The user selects the appropriate group and clicks **OK**.
- If multiple packages are assigned to the user, a dialog box displays to choose the package to run. An example is shown below. Select the appropriate Runtime package and click **OK** or **Cancel**.



- If the Runtime user is assigned to multiple groups, a Select Group dialog box displays.

Select the appropriate Group and click **OK** or **Cancel**.

Exit

Select **Exit** to close OpenSpan Runtime.

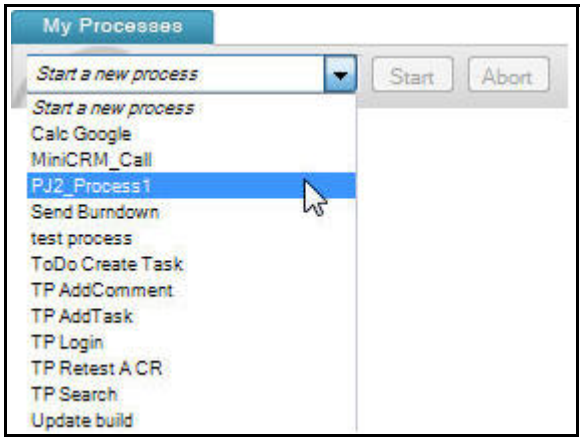
Events

The topic “Default | (Package)” on page 5-6 mentions that properly configured Runtime instances can send High-Level Events from the user desktop to OpenSpan Server for analysis. For information on the Server-side configuration of this option, see “Managing Group Runtime Settings” on page 4-39.

Process Toolbar

If enabled, the Process Toolbar displays on Runtime start. The Process Toolbar is an instrumentation option that lets a user send, stop, or reset process events to the Events table in the database in a way that mimics the SendStart/SendStop/Abort methods from process automations (for more information on process automations, see “Producing Process Automations” on page 3-9). The idea is to bracket user events from discrete business procedures with distinct metadata that allows the events between Start and Stop to be measured by time elapsed, number of keystrokes, etc., and to inject custom dimension or fact metadata in the stream as a bonus. Reports modules in OpenSpan Server display the data. See “Process Activity Reports” on page 6-13, “Process User Performance Reports” on page 6-15, “Comparative Process Activity Report” on page 6-16, and “Comparative Process User Performance Report” on page 6-17.,

1. From the Process Toolbar, select from a menu of **available Processes**. If no processes are defined on OpenSpan Server, none will be visible in this drop-down.



2. Click **Start** to send a Start Event that is tied to the selected process to the Events table of the database,

3. Perform the user desktop steps that correspond to the selected process. When complete, click **Stop**. This sends a Stop Event to the Events table of the database.

4. A third option is available to stop the process without reporting to the Events table of the database; click **Abort**, and the Process Toolbar state resets to ready. No Start or Stop Events for that process instance are recorded. It lets a user start over without leaving orphaned processes awaiting a Stop.

Server-side Settings

For information on enabling display of the Process Toolbar via Server-side configuration, see “Managing Group Runtime Settings” on page 4-39.

Chapter 6 OpenSpan Server 5.2 Reports and Graphs

In this Chapter

Welcome to the Reports and Graphs Chapter of the OpenSpan Server 5.2 Administration and User Guide. When you have finished reading this chapter, you will be familiar with these topics:

General

The Reports and Graphs features of OpenSpan Server provide reporting on the high-level events collected from Runtime users.

OpenSpan Server administrators have the option of enabling high-level events reporting from every desktop running OpenSpan Runtime. Reporting isn't necessarily tied to, or doesn't require, the actual download of a Runtime package. Rather, if enabled with or without a package download, each desktop can report the following information to the Events database:

TABLE 1. High-Level Events Monitored by OpenSpan Server

Event	Notes
Keystroke Count	
Error Keystrokes	
Mouse Click Count	
Mouse Wheel Count	
Copy Count	Limited to keystroke commands, not right-click copies to the clipboard.
Paste Count	Limited to keystroke commands, not right-click pastes from to the clipboard.
Cut Count	Limited to keystroke commands, not right-click cuts to the clipboard.
Window Move/Resize Count	
Scroll Count	Window scrolls.

TABLE 1. High-Level Events Monitored by OpenSpan Server (Continued)

Event	Notes
Key Stroke Error Ratio	Keys divided by backspace/delete.
Total Cut/Copy/Paste	Keyboard input only, not right mouse clicks.
Application Churn	Count of monitored applications gaining/losing focus.
Application Duration	Time monitored applications in focus.

In a standard installation, this data accumulates according to defined schemas in the Events table of the OpenSpan Server database. OpenSpan Server is equipped with standard reports and visualizations to help business analysts refine development and deployment of OpenSpan Runtime packages to user desktops across an enterprise.

Report Display Types

Heat Maps

Heat Maps use rectangle (cell) size and color to correspond to two variables: Duration and Metric. The **size** of the rectangles illustrates **duration** (i.e., time spent in an application or performing a process). The **color** represents the High-level Event **metric count**.

- **Larger:** More time spent in an application or performing a process
- **Smaller:** Less time spent in an application or performing a process
- **Red (brighter, "hotter")**: High metric count; the user or users performed the event many times
- **Blue (darker, "cooler")**: Low metric count; the user or users performed the event less times

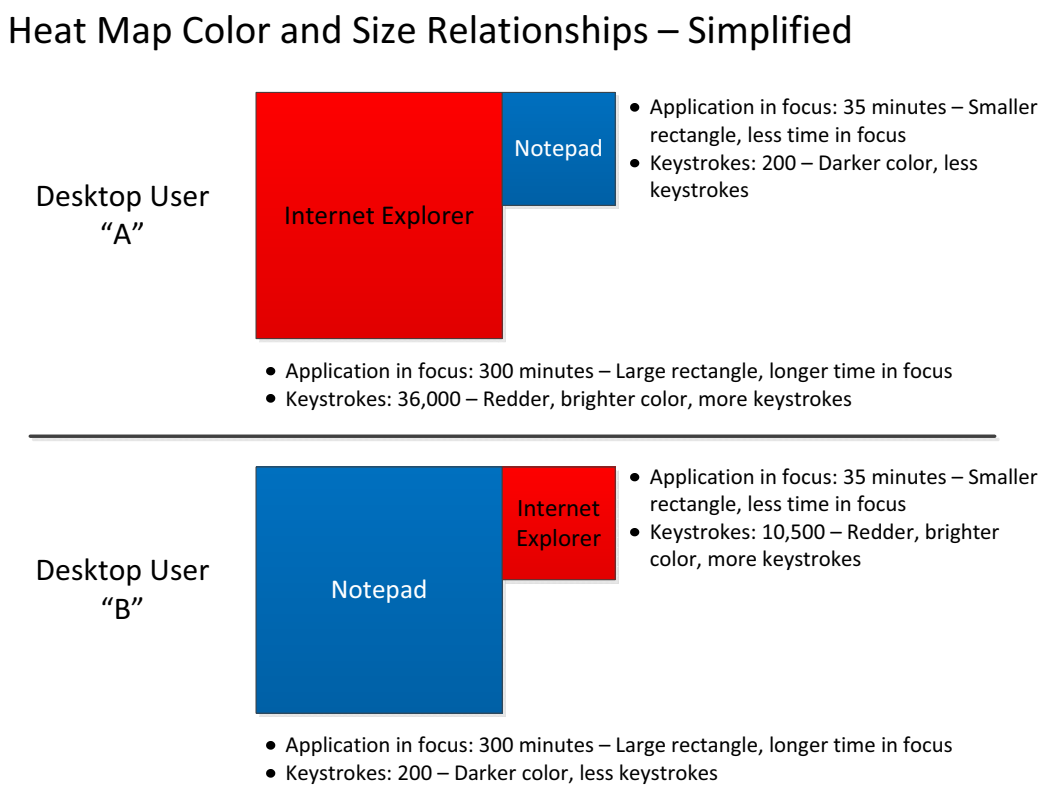
See the next page for an example.

Heat Map Visual Relationships

In this simple comparison, imagine that desktop users “A” and “B” are using Internet Explorer and Notepad. Events reporting is turned on. After a 6-hour period, a business analyst generates a heat map of users “A” and “B”’s reported events. Without even drilling down into the metrics, some things stand out from the heat map.

- User “A” spends significantly more time in Internet Explorer than in Notepad. We know this because the Internet Explorer rectangle is bigger than the Notepad rectangle, reflecting a much longer cumulative time that Internet Explorer was in focus on User “A”’s desktop.
- User “A” also seems to be much busier in Internet Explorer than in Notepad. This is indicated by the color of User “A”’s Internet Explorer rectangle. The redder, brighter color reflects a large number of recorded keystroke events, roughly 120 per minute.
- In contrast, User “B”’s heat map shows completely different use of the same applications. We know from the size of the rectangles that User “B” spent much more time in Notepad than Internet Explorer. However, User “B” recorded only 1.5 keystrokes per minute in Notepad, versus 300 per minute in Internet Explorer for a much shorter period of time.

Clicking a heat map cell opens the corresponding user performance *histogram*. See the next section.



Histograms

Histograms are vertical bar charts that represent the distribution of Runtime user performance for a selected high level event (i.e., report metric). Examples are show under “Application Activity Report Heat Maps and Histograms” on page 6-10, “Application User Performance Reports” on page 6-12, and “Process User Performance Reports” on page 6-15.

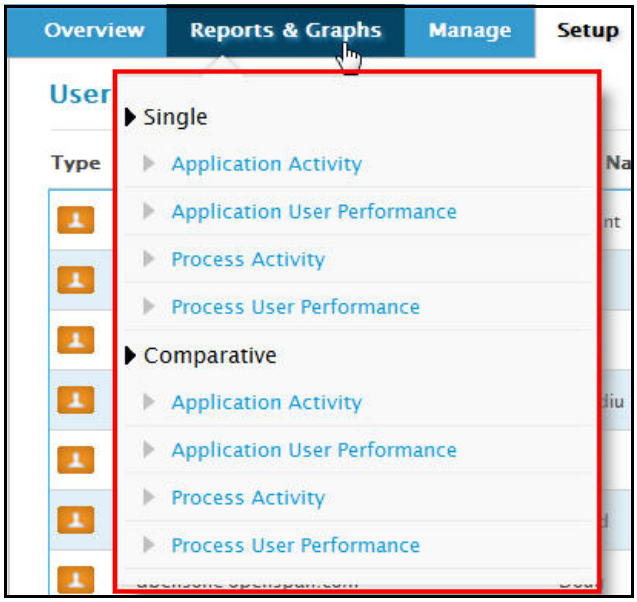
Individual user performance for the report metric falls into ranges (or bins). A chart bar is drawn for each range.

The bar height corresponds to the number of users in that bin/class; the y-axis indicates the user count, and the percentage of reporting users is shown as a box superimposed over the top of a given bar. OpenSpan Server reports use two types of histograms, Application User Performance and Process User Performance.

Accessing Reports and Graphs

Selection

The OpenSpan Server Reports and Graphs facility is available via the main Reports and Graphs menu.



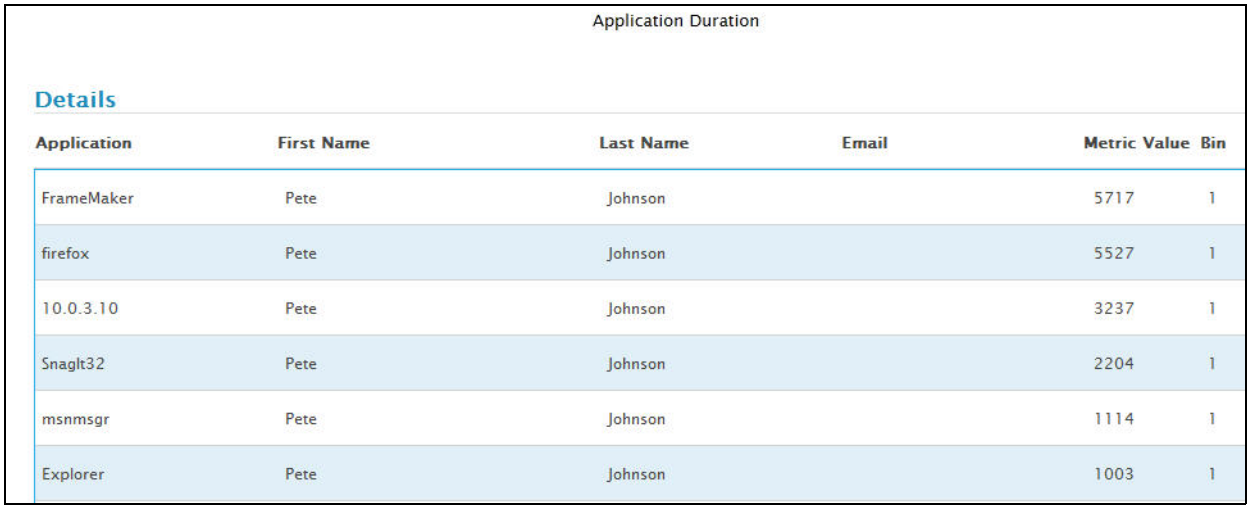
1. Select **Reports and Graphs**. The drop-down menu displays.
2. Select a report.

- Single reports display report results for one case at a time.
- Comparative reports take the single reports and stack two of them side-by-side.

Report Controls and Options

A number of the standard reports have additional on-screen choices.

- **Export link**: Export graph or detail (grid) data to a comma separated value file. Graph data does not include detail on specific runtime user performance.
- **Show Details** (toggles to **Hide Details**) link: Displays a data grid under the report listing each Runtime user with their performance on the report metric and application/process duration such as this example.



Application Duration

Details

Application	First Name	Last Name	Email	Metric Value	Bin
FrameMaker	Pete	Johnson		5717	1
firefox	Pete	Johnson		5527	1
10.0.3.10	Pete	Johnson		3237	1
SnagIt32	Pete	Johnson		2204	1
msnmsgr	Pete	Johnson		1114	1
Explorer	Pete	Johnson		1003	1

Single Reports

Single reports provide a rich data return against the selection of a Group or specific User over a day, week, or month.

Application Activity Reports

Report Criteria

The Application Activity report lets you select the following criteria.

TABLE 2. Application Activity Reports Criteria

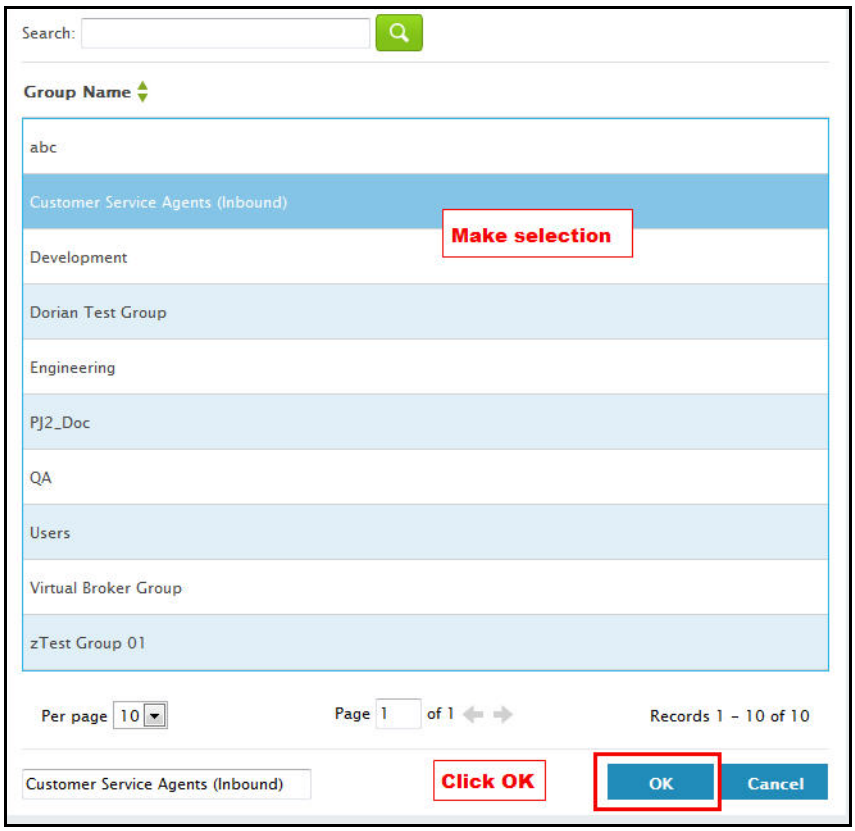
Selection	Details
Metrics	Keystroke Count Error Keystrokes Mouse Click Count Mouse Wheel Count Copy Count Paste Count Cut Count Window Move/Resize Count Scroll Count Keystroke Error Ratio Total Cut/Copy/Paste Application Churn
Activity Period	Day Week Month
Group or User (see “To choose Groups or Users” on page 6-8)	All Users Selected Group Selected User

To choose Groups or Users

This step-list describes the assignment of Groups or Users for the report:

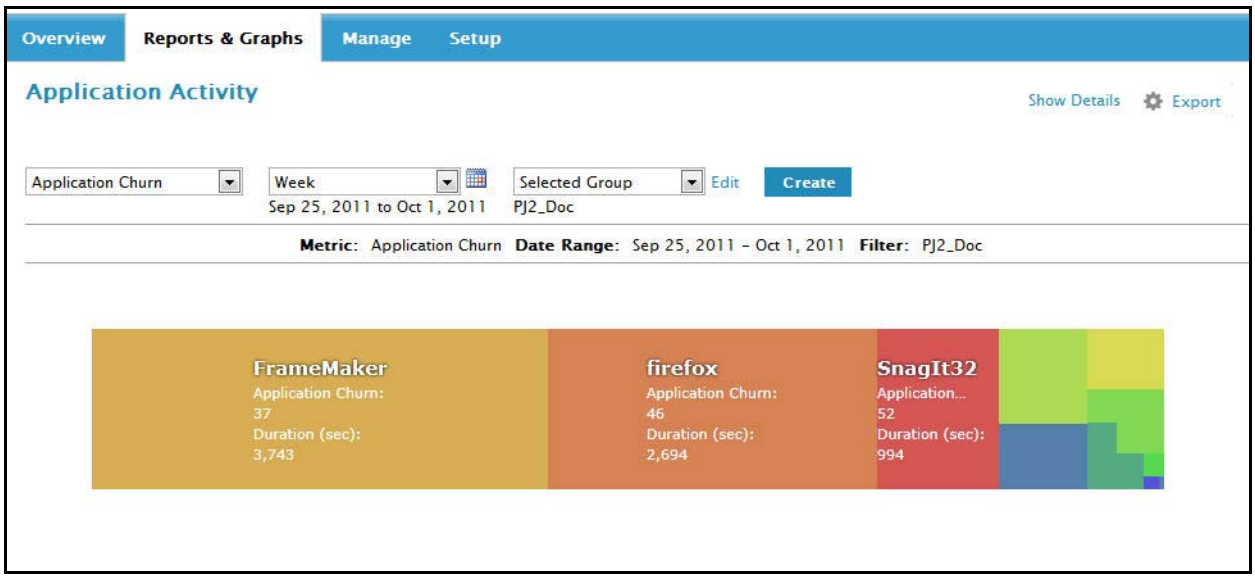
1. Select **Selected Group** or **Selected User** from the drop-down list box. The default is **All Users**.
2. Use the **Search** box as a quick shortcut if the list is long.

3. Click the Group or User of your choice.
4. Click **OK**.



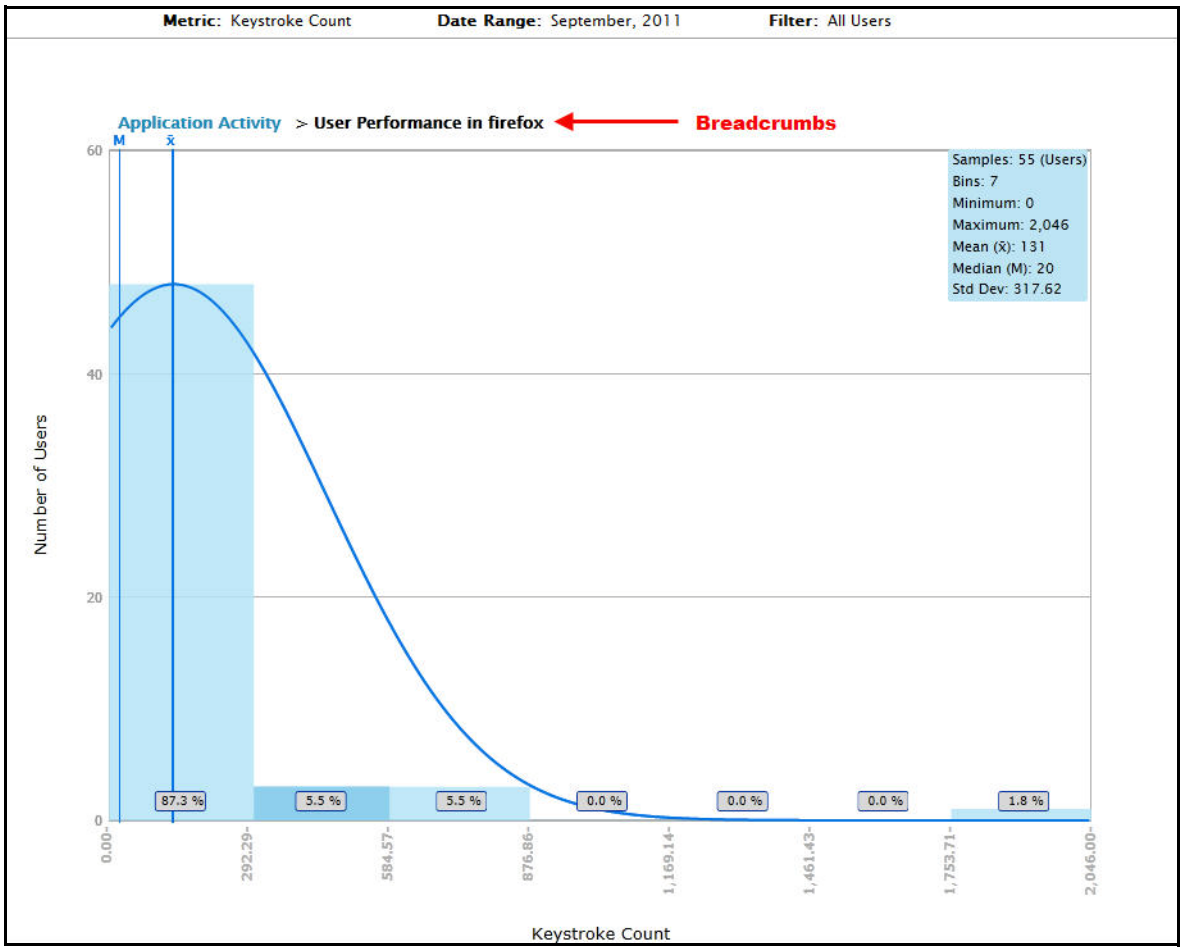
Application Activity Report Heat Maps and Histograms

For Application Activity Heat Maps each rectangle, or cell, represents an individual application used during the selected reporting period. The size of each cell corresponds to the total time the application was in focus for all reporting users combined. Cell color and position corresponds to the report metric. The deeper (brighter) the red color, and the closer the cell is to the top-left of the report, the greater the metric value.



Each heat map rectangle can be clicked and expanded for further analysis.

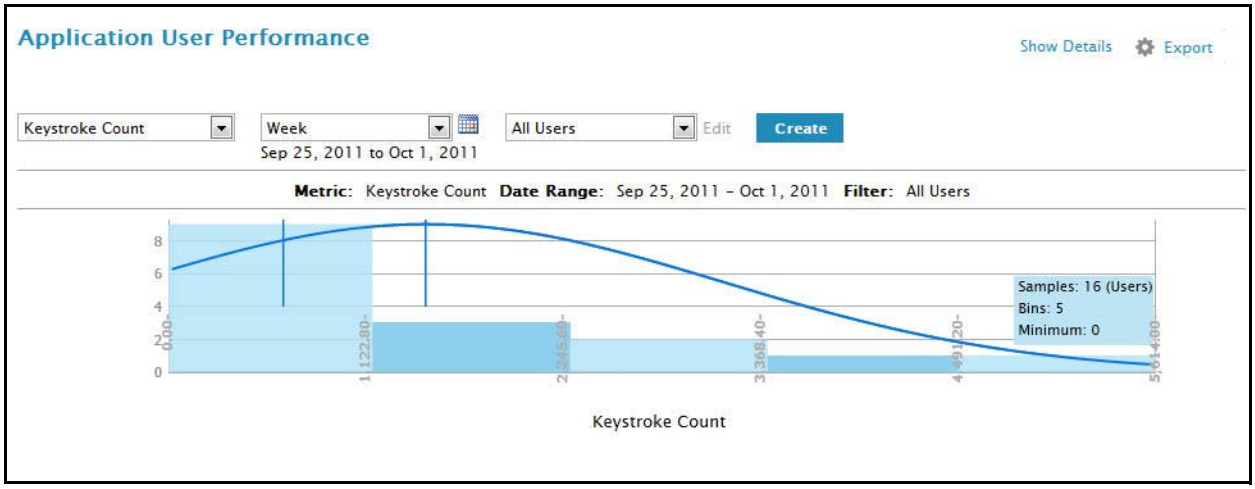
Note: The drill-through report includes breadcrumbs to return to the parent heat map.



Application User Performance Reports

The Application User Performance report functions the same as “Comparative Application Activity Report” on page 6-16, except it adds **Application Duration** under metrics. This displays the length of time that user applications were in focus on the desktop.

This report histogram example represents user performance for a high-level event metric (e.g., keystrokes) in all applications during the selected reporting period. The Y-axis (vertical) shows the number of users within each bin (i.e., frequency) and the x-axis (horizontal) shows how users performed on the selected metric.



Process Activity Reports

The Process Activity report uses the **same criteria** as the **Application User Activity** report, **except** it concerns the **processes defined in the OpenSpan Server Processes grid**. With this report it's possible to display and compare the same metrics for the business processes in use.

Process events accumulate on the client side starting when an application comes into focus and then are sent as a group when the application loses focus, instead of event-by-event as in previous releases. This makes reporting comparisons faster and more efficient.

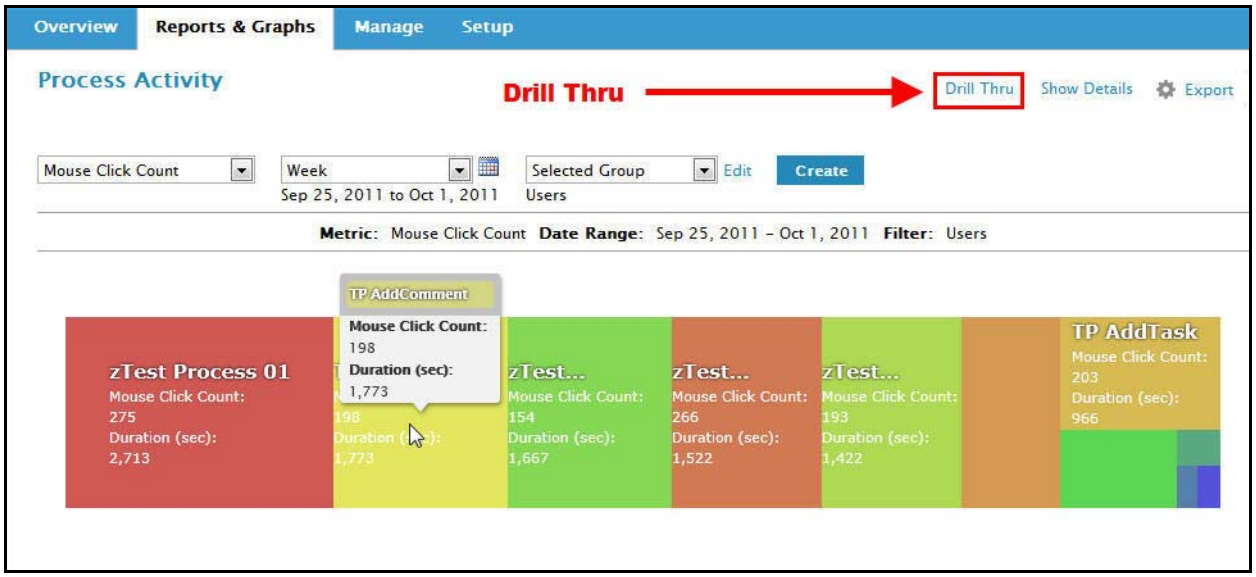
Process Activity Report Heat Maps

Each cell of a Process Activity Heat Map represents an individual process completed (i.e., started and stopped — not abandoned) by one or more OpenSpan Runtime users submitting process start and stop events to the OpenSpan Server during the selected reporting period.

Processes are bounded by start and stop events identified either in OpenSpan solution automations or manually using the OpenSpan Runtime Process Toolbar (see "Process Toolbar" on page 5-7). A process will generally use one more applications and the combined application activity for all reporting Runtime users, within the bounds of a specific process, determines the size, color and position of a heat map cell.

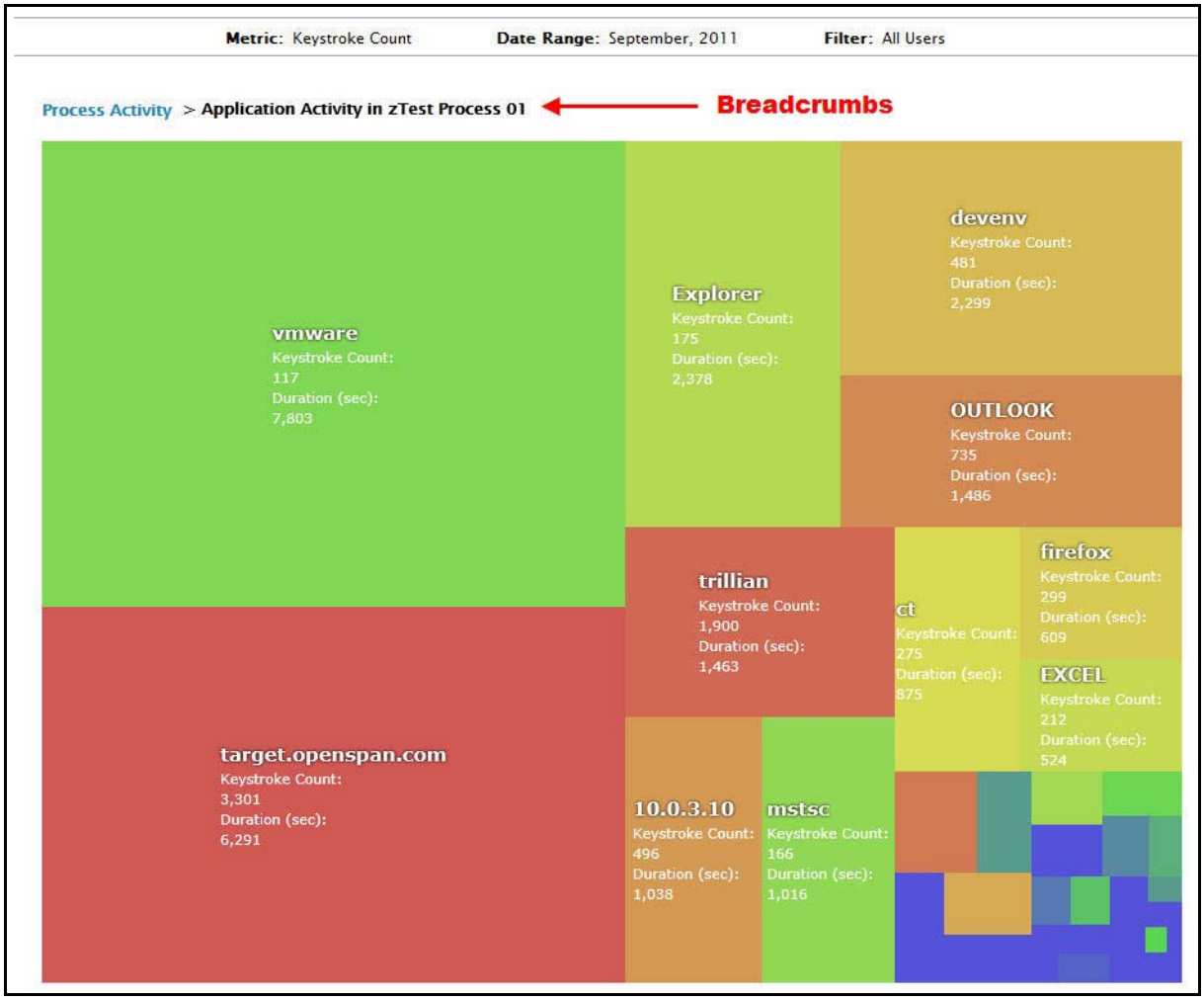
The size of each cell corresponds to the duration (i.e., total time the process took to complete for all reporting users combined). Cell color and position corresponds to the report metric. The deeper (brighter) red in color the cell, and the closer the cell is to the top-left of the report, the greater (higher) the metric value.

Use the Drill-thru link to expand the report and show the applications used by the specific process.



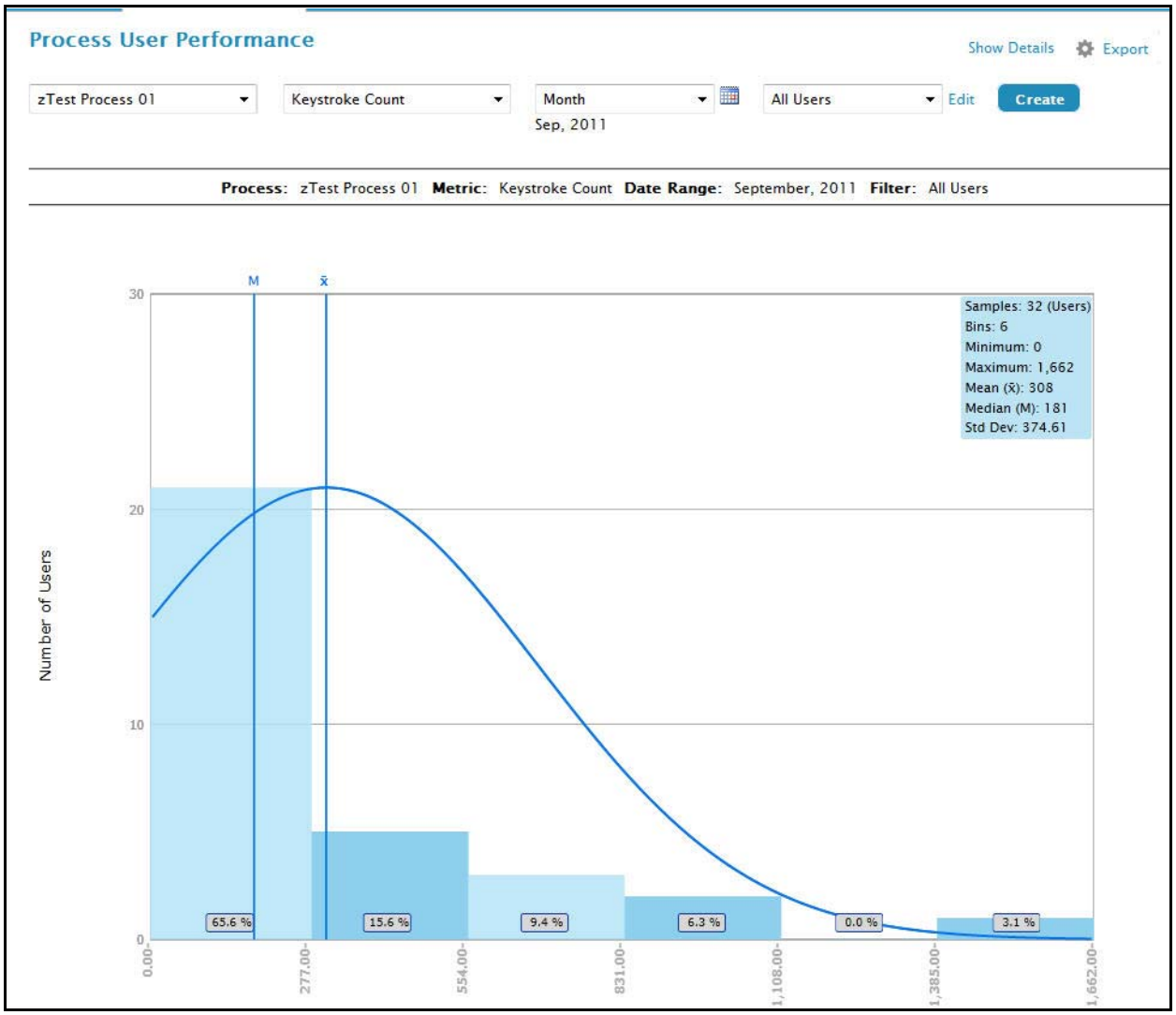
The drill through report inherits the metric and reporting period from the parent heat map. This cannot be changed in the drill-through report.

Note: The drill-through report includes breadcrumbs to return to the parent heat map.



Process User Performance Reports

And finally, the Process User Performance report uses the **same criteria** as the Application User Performance report — adding Application Duration — **but applies to the active Processes**. The Process User Performance histogram represents user performance for a high-level event metric (e.g., keystrokes) for a specific process during the selected reporting period. The Y-axis (vertical) shows the number of users within each bin (i.e., frequency) and the x-axis (horizontal) shows how users performed on the selected metric.

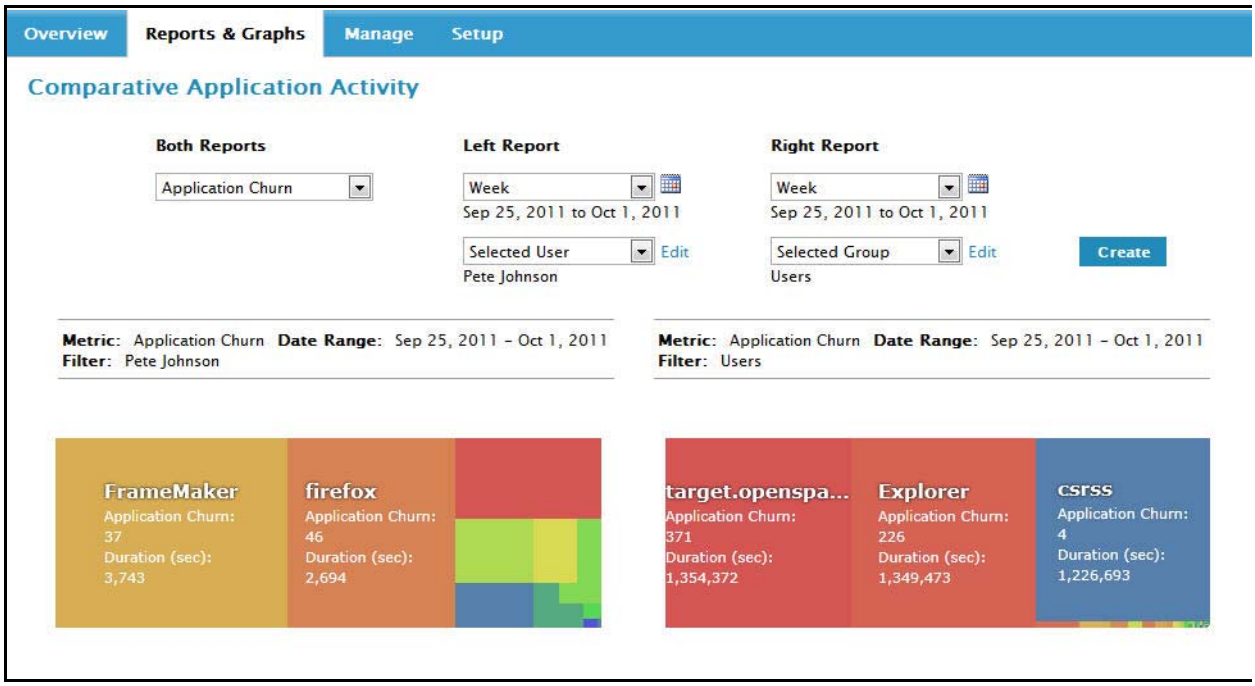


Comparative Reports

Comparative reports let you build two reports for side-by-side comparison using the criteria from the single reports described in “Single Reports” on page 6-8.

Comparative Application Activity Report

This report allows a comparison of events for applications by user or group. For details, see the similar “Application Activity Reports” on page 6-8. The example below shows how a metric (Application Churn) for one user (left) can be compared with the same metric for a Group (Users, right) over a given time period.



Comparative Application User Performance Report

This report adds the **Application Duration** metric reporting to the Comparative Application Activity Report.

Comparative Process Activity Report

Basically the same report as the “Comparative Reports” on page 6-16, but allows comparisons of Groups and Users.

Comparative Process User Performance Report

This report compares two sets of values in the same fashion as the Comparative Application User report, but adds a Process drop-down selection of the Processes defined in the OpenSpan Server Processes grid.