ivanti

ITSM-On-Premises-2025 (ENU)

ivanti

CONTENTS

1. Overview **5**
2. Conventions **6**
3. System Requirements **7**
3.1. Hardware and Software Requirements 8
3.2. About Third-Party Software 17
3.3. Port Requirements 19
3.4. Compatibility Matrix 22
3.5. Troubleshooting - System Requirements 26
4. Installation and Deployment **27**
4.1. Neurons for ITSM Deployment Overview 28
4.1.1. Demonstration or Proof-of-Concept Deployment 29
4.1.2. Minimum Production Deployment 31
4.1.3. Enterprise Production Deployment 33
4.1.4. Security Enterprise Production Deployment 37
4.1.5. Database Deployment Options 39
4.1.6. Operations Console Deployment Options 44
4.1.7. Reporting Services Deployment Options 49
4.1.8. Voice Deployment Options 53
4.2. Neurons for ITSM Installation Prerequisites 55
4.2.1. About Roles 56
4.2.2. About the Different Accounts Used in Neurons for ITSM 58
4.2.3. About Passwords Used in Neurons for ITSM 60
4.2.4. About Using a Virtual Machine with Neurons for ITSM 61
4.2.5. Enabling Attachment File Streaming 62
4.2.6. SSL Configuration 65
4.2.7. Installing Neurons for ITSM Reporting Feature 70
4.2.8. Full Text Search 72
4.2.9. Verifying Server Roles and Features 74
4.3. Installing the Neurons for ITSM System 76
4.3.1. Installing Neurons for ITSM 77
4.3.2. Installing the Reporting Feature 81
4.3.3. Installing Discovery on a Dedicated Server 83
4.3.4. Installing the Knowledge Uploader (Knowledge Import Tool) 84
4.4. Initially Configuring the System 92
4.4.1. Using the System Configuration Wizard 96
4.4.2. Configuring Neurons for ITSM 99
4.4.2.1. Configuring the Configuration Database 100
4.4.2.2. Configuring the Neurons for ITSM Application 106
4.4.2.3. Configuring the Application Server Settings 112
4.4.2.4. Configuring Other Feature Settings 116
4.4.2.5. Configuring the Metrics Server 120
4.4.2.6. Configuring the Discovery Application Server 125
4.4.2.7. Configuring the Discovery Web Server 131
4.4.2.8. Finishing the System Configuration 134
4.4.3. Configuring the Reporting Feature 135
4.4.4. Configuring Discovery 146
4.4.5. Configuring the Deployment on the Neurons for ITSM Operations Console 149
4.4.6. Optional SSL Configuration 154
4.4.7. Optional LDAP Configuration 163
4.4.8. About Configuring with ADFS 164
4.4.9. About Configuring Throttling Settings 165
4.5. Setting up Redis 167
4.6. Logging into Neurons for ITSM 175
4.7. Adding Features and Changing Settings 177
4.8. Upgrading Neurons for ITSM from an Earlier Release 179
4.9. Using the License Manager 186
4.10. Troubleshooting - Installation and Deployment 200
4.11. Chat Configuration 218
4.11.1. Chatting with Self Service Mobile portal Users 221

5. Configuration Database ... 223
5.1. About Logging into the Configuration Database ... 224
5.2. Workspaces that Already Exist ... 225
5.3. Working With Features ... 226
5.4. Securing IIS website ... 227
5.5. Internal Use Only Workspaces ... 229
5.6. Workspaces Populated by the System Configuration Wizard ... 230
5.7. Central Config API Authentication ... 231
5.8. Setting up the Teams Bot Integration ... 236
5.9. Importing Packages with SQL ... 241
5.10. Enabling the Action Service ... 242
5.11. Workspaces that Can Be Used ... 246
5.11.1. Configuring the Message Queue ... 247
5.11.2. Configuring the Message Queue Handler ... 248
5.11.3. Configuring Email ... 249
5.11.4. Viewing All Logs ... 250
5.11.5. Working With Customer Contacts ... 253
5.11.6. Working With Tenants ... 255
5.11.7. Configuring Manual Load Partitioning ... 258
5.11.8. Firebase Configuration for Mobile Push Notification ... 261
5.11.9. Decommissioning FTP and File on Shared folder Integration ... 277
6. Setting up the Teams Bot Integration ... 279
7. Performance Tuning ... 281
7.1. Server and Connection Tuning ... 282
7.1.1. Working with Log Files and Logging Levels ... 283
7.1.2. Configuring the Application Pool Recycling Schedule ... 284
7.2. Platform and Application Tuning ... 285
7.2.1. Planning and Scheduling Maintenance ... 286
7.2.2. Tuning Business Rules and Workflows ... 289
7.2.3. Best Practices for Dashboards and Layouts ... 290
7.2.4. About Using Combo Boxes Sparingly ... 292
7.2.5. Evaluating Slow Running Queries ... 293
7.3. Database Tuning ... 295
7.3.1. Tuning Indexes ... 303
7.3.2. About Using Full Text Search ... 310
7.4. Performance Monitoring ... 311
7.5. Troubleshooting - Performance Tuning ... 317
7.6. Deleting Inactive Tables and Records ... 318
7.6.1. Best Practices for Deleting Tables and Records ... 319
8. Operations Console ... 321
8.1. Operations Console Overview ... 323
8.1.1. Operations Console Interface ... 324
8.1.2. Terminology ... 326
8.1.3. Installing the Operations Console ... 334
8.2. Before You Begin Migrating Data ... 336
8.2.1. Items Excluded from Migration ... 345
8.3. Starting the Data Migration ... 346
8.4. Migrating Data using Heat Development Project ... 352
8.4.1. Example: Migrating Request Offering Changes By Enabling the Ivanti Service Manager Development Project ... 356
8.5. Troubleshooting and Known Issues ... 360
8.5.1. Configuring Landscapes ... 369
8.5.2. Working with Tenants and Tenant Instances ... 377
8.6. Appendix A: Using the Operations Console to Perform Administrative Tasks ... 386
8.7. Appendix B: Migrating Data from One Tenant Instance to Another Using Simple Mode or Advanced Mode ... 394
8.7.1. Migration Scenarios Using Simple Mode and Advanced Mode ... 414
8.8. Appendix C: Using the ISM Development Project to Migrate Data ... 420
8.9. Appendix D: Working with Packages from the App Store ... 431
9. Web Services ... 434
9.1. Request Offerings and Service Requests ... 435
9.2. Searching for Records ... 436
9.3. Understanding the Search Results ... 448
9.4. Web Methods ... 499

9.5. FRSHEATIntegration Web Service URL 523
10. Known Issues 534

1. Overview

Welcome to Neurons for ITSM On-premise online help. If you are an on-premise customers looking for information about installing, configuring, system requirements, ops console, performance tuning, and web services, you are in the right place. Click the tiles below or use the table of content on the left menu to view topics. To search for specific information, use the search bar and search using a relevant keyword.

Installation and Deployment

Configuration Database

System Requirements

Performance Tuning

Ops Console

Web Services

2. Conventions

This guide uses the following conventions:

Convention	Definition and Usage
Bold	Text that displays in a GUI element such as, a menu item, button, or element of a dialog box, and command names are shown in bold. For example: Click **Edit** from the toolbar.
Italic	Variables appear in italics. Important information may be italicized for emphasis.
`Monospace`	Commands, command-line output, and file names are in monospace type. For example: `Execute setup.exe` to install this release.
[]	Square brackets surrounding a command-line argument mean that the argument is optional.
|	Vertical bars separating command-line arguments mean that only one of the arguments can be used.

3. System Requirements

The *System Requirements and Compatibility Matrix for* Neurons for ITSM - *Version* 2025.3 is intended for administrators who are going to install Neurons for ITSM on your local computer or virtual machine. This document contains the following sections:

- **Hardware and Software Requirements**. Contains the hardware and software requirements.
- **About Third-Party Software**. Discusses third-party software that is used in Neurons for ITSM.
- **Port Requirements**. Lists the required ports.
- **Compatibility Matrix**. Lists the compatibility of Neurons for ITSM Release 2025.3 with other software.
- **Troubleshooting**. Contains solutions to common problems related to system requirements.

3.1. Hardware and Software Requirements

This section outlines the hardware and software requirements necessary for the installation of Neurons for ITSM.

Application Server Requirements

This section explains the hardware and software specifications needed to properly install, run, and maintain an application server.

Hardware

Component	Recommended
CPU	• 2 core minimum • 4 core recommended for heavy transaction rates and workflow/escalation usage
RAM	• 8 GB minimum, and more depending on transaction rates and workflow/escalation complexity
Hard Disk	• 20 GB minimum free space (approximately 1.1 GB for Neurons for ITSM files and the remaining space for log files). • Best Practice! We recommend that you install Neurons for ITSM in a location other than on the system drive (typically the C drive).

Software

Item	Supported Version
Operating System	• Microsoft Windows Server 2025 (64-bit) (recommended) (supports ITSM 2025.3 or later) • Microsoft Windows Server 2022 (64-bit) • Microsoft Windows Server 2019 (64-bit) • Microsoft Windows Server 2016 (64-bit) (ITSM 2025.3 is the last version to support Microsoft Windows Server 2016)
Microsoft .NET Framework	• If FedRAMP feature is not enabled, the minimum required version is Microsoft .NET Framework Release 4.6. If FedRAMP feature is enabled, the minimum required version is Microsoft .NET Framework Release 4.8. This requires the latest Microsoft Windows update.
Web Server	• Microsoft IIS Release 10 • Microsoft IIS Release 8.5 (supports Windows Server 2016 or later) • Microsoft IIS Release 8.0 (supports Windows Server 2016 or later. The ITSM 2025.2 is the last version to support Microsoft IIS Release 8.0) • Microsoft IIS Release 7.5 (supports Windows Server 2016 or later. The ITSM 2025.2 is the last version to support Microsoft IIS Release 7.5)

Web Server Requirements

This section explains the hardware and software specifications needed to install, configure, and run a web server.

In multi-server environments, each Neurons for ITSM web server must meet these requirements.

Hardware

Component	Recommended
CPU	• 2 core minimum • 4 core recommended for heavy transaction rates and workflow/escalation usage
RAM	• 8 GB minimum, and more depending on transaction rates and workflow/escalation complexity
Hard Disk	• 20 GB minimum free space (approximately 1.1 GB for Neurons for ITSM files and the remaining space for log files). • Best Practice! We recommend that you install Neurons for ITSM in a location other than on the system drive (typically the C drive).

Software

Item	Supported Version
Operating System	• Microsoft Windows Server 2025 (recommended) (supports ITSM 2025.3 or later) • Microsoft Windows Server 2022 • Microsoft Windows Server 2019 • Microsoft Windows Server 2016 (ITSM 2025.3 is the last version to support Microsoft Windows Server 2016)
Microsoft .NET Framework	• If FedRAMP feature is not enabled, the minimum required version is Microsoft .NET Framework Release 4.6. If FedRAMP feature is enabled, the minimum required version is Microsoft .NET Framework Release 4.8. This requires the latest Microsoft Windows update or that you download the following security patch: https://www.microsoft.com/en-us/download/details.aspx?id=42327
Web Server	• Microsoft IIS Release 10 • Microsoft IIS Release 8.5 (supports Windows Server 2016 or later) • Microsoft IIS Release 8.0 (supports Windows Server 2016 or later. ITSM 2025.2 is the last version to support Microsoft IIS Release 8.0) • Microsoft IIS Release 7.5 (supports Windows Server 2016 or later. ITSM 2025.2 is the last version to support Microsoft IIS Release 7.5)

Database Server Requirements

This section explains the hardware and software specifications needed to install, run, and maintain a database server.

Hardware

Component	Recommended
CPU	• 4 core minimum. Your deployment may require more for heavy transaction rate and workflow or escalation usage
RAM	• 16 GB minimum; however, your deployment may require much more RAM depending on transaction rates and workflow or escalation complexity
Hard Disk	• Best Practice! We recommend that you have separate volumes for the database and the transaction logs. • **Database**: 100 GB minimum free space. For proper disk space sizing, use these guidelines: ◦ Neurons for ITSM application demo database: 500 MB ◦ Average disk space requirement per record (incident, problem, change, configuration): ~200 KB ◦ For storing records more than three years old: more than 60 GB ◦ Average disk space requirement per record (configuration item): ~500 KB. For example, for 20,000 configuration items, we recommend a minimum of 10 GB. ◦ 10 GB minimum free space. This is the minimum. • **Transaction logs**: 60 GB minimum free space.

Software

> If your deployment has the reporting feature installed, you must have Microsoft .NET Framework Release 4.8, which requires the latest Microsoft Windows update or that you download the following security patch: **https://www.microsoft.com/en-us/download/details.aspx?id=42327**.

Item	Supported Version
Operating System	• Microsoft Windows Server 2025 (recommended) (supports ITSM 2025.3 or later) • Microsoft Windows Server 2022 • Microsoft Windows Server 2019 • Microsoft Windows Server 2016 (ITSM 2025.3 is the last version to support Microsoft Windows Server 2016)
Database Management Software	• Microsoft SQL Server 2022 • Microsoft SQL Server 2019 • Microsoft SQL Server 2017 • Microsoft SQL Server 2016 (ITSM 2025.2 is the last version to support Microsoft SQL Server 2016)
Other Software	• iFilters (required for full-text indexing) • An iFilter is a plugin that allows the Microsoft SQL Server to index various file formats so that they become searchable. Download the latest version of the iFilter for the document type (these links are subject to change): ◦ Microsoft Office 2010: **https://www.microsoft.com/en-us/download/details.aspx?id=5829** • Adobe PDF iFilter 64 Release 11.0.01: **http://www.adobe.com/support/downloads/detail.jsp?ftpID=5542**

- Microsoft SQL Server must have full-text search enabled. If it is not enabled, the Neurons for ITSM demo database will not run properly.
- ITSM application does not support case-sensitive SQL Collation.
- ITSM application only supports the SQL Collation of SQL_Latin1_General_CP1_CI_AS.

If you are using Microsoft SQL Server 2022 and consider migrating to a Cloud deployment of Neurons for ITSM, we recommend to downgrade your SQL 2022 database to SQL 2019 prior to migrating.

Reporting Feature Requirements

This section explains the requirements to implement or support a reporting feature in a software system.

Hardware

Component	Recommended
CPU	• 2 core minimum • 4 core recommended for heavy transaction rates and workflow and escalation usage
RAM	• 8 GB minimum, and more depending on transaction rates and workflow and escalation complexity
Hard Disk	• 20 GB minimum free space (approximately 1.1 GB for Neurons for ITSM files and the remaining space for log files). • Best Practice! We recommend that you install Neurons for ITSM in a location other than on the system drive (typically the C drive).

Software

Item	Supported Version
Operating System	• Microsoft Windows Server 2025 (recommended) (supports ITSM 2025.3 or later) • Microsoft Windows Server 2022 • Microsoft Windows Server 2019 • Microsoft Windows Server 2016 (ITSM 2025.3 is the last version to support Microsoft Windows Server 2016)
Microsoft .NET Framework	• If FedRAMP feature is not enabled, the minimum required version is Microsoft .NET Framework Release 4.6. If FedRAMP feature is enabled, the minimum required version is Microsoft .NET Framework Release 4.8. This requires the latest Microsoft Windows update or that you download the following security patch: https://www.microsoft.com/en- us/download/details.aspx?id=42327
Database Management Software	• Microsoft SQL Server 2022 • Microsoft SQL Server 2019 • Microsoft SQL Server 2017 • Microsoft SQL Server 2016 (ITSM 2025.3 is the last version to support Microsoft SQL Server 2016)
Other Software	Microsoft SQL Server Reporting Services (SSRS) • Microsoft SSRS 2022 • Microsoft SSRS 2019 • Microsoft SSRS 2017 • Microsoft SSRS 2016 (ITSM 2025.2 is the last version that supports SSRS 2016)

Microsoft SSRS Report Designer

Business Intelligence Development Studio (BIDS) is a Microsoft component used in creating reports and report templates for Neurons for ITSM. BIDS comes with Microsoft SQL Express edition. For on-premise users, BIDS comes with Microsoft SQL Server license.

Item	Supported Version
Operating System	• Microsoft Windows Server 2025 (recommended) (supports ITSM 2025.3 or later) • Microsoft Windows Server 2022 • Microsoft Windows Server 2019
Hard Disk	• 5 GB minimum free space
RAM	
Processors	• **X86**: Pentium III-compatible processor or faster (processor speed of 1.0 GHz or faster); Windows only; 64-bit x86 PC compatible • **X64**: Minimum: AMD Opteron, AMD Athlon 64, Intel Xeon with Intel EM64T support, Intel Pentium IV with EM64T support (processor speed of 1.0 GHz or faster)
Other Software	Microsoft Business Intelligence Development Studio (BIDS) • Microsoft BIDS 2022 • Microsoft BIDS 2019 • Microsoft BIDS 2017 • Microsoft BIDS 2016 (ITSM 2025.2 is the last version that supports BIDS 2016)

512 MB minimum, 2GB is recommended

Discovery Requirements

This section explains the requirements needed to successfully perform automated asset discovery and service mapping across your IT infrastructure.

Hardware

Component	Recommended
CPU	• 2 core minimum • 4 core recommended for heavy transaction rates and workflow and escalation usage
RAM	• 8 GB minimum, and more depending on transaction rates and workflow and escalation complexity
Hard Disk	• 20 GB minimum free space (approximately 1.1 GB for Neurons for ITSM files and the remaining space for log files). • Best Practice! We recommend that you install Neurons for ITSM in a location other than on the system drive (typically the C drive).

Software

Item	Supported Version
Operating System	• Microsoft Windows Server 2025 (recommended) (supports ITSM 2025.3 or later) • Microsoft Windows Server 2022 • Microsoft Windows Server 2019 • Microsoft Windows Server 2016 (ITSM 2025.3 is the last version to support Microsoft Windows Server 2016)
Microsoft .NET Framework	• If FedRAMP feature is not enabled, the minimum required version is Microsoft .NET Framework Release 4.6. If FedRAMP feature is enabled, the minimum required version is Microsoft .NET Framework Release 4.8. This requires the latest Microsoft Windows update or that you download the following security patch: https://www.microsoft.com/en- us/download/details.aspx?id=42327
Web Server	• Microsoft IIS Release 10 • Microsoft IIS Release 8.5 (supports Windows Server 2016 or later) • Microsoft IIS Release 8.0 (supports Windows Server 2016 or later. The ITSM 2025.2 is the last version to support IIS Release 8.0) • Microsoft IIS Release 7.5 (supports Windows Server 2016 or later. The ITSM 2025.2 is the last version to support IIS Release 7.5)

Client Computer Requirements

This section explains the requirements that your computer (client) must meet to successfully run a specific application, connect to a server, or access a system.

Hardware

Component	Recommended
CPU	Single core
RAM	4 GB

Mobile

Neurons for ITSM offers native apps for mobile devices for the Android and iOS platforms. The Neurons for ITSM app can be downloaded from their respective platform's app stores.

Platform	Supported Version
Android	12.0 or later
iOS	17.0 or later

Software

See the **Browser Compatibility** for browser compatibility information.

There is a Known Issue during installation of the 2017.2 Discovery Client Agent.

Installation pops up Windows Warning for digital signature requirement from Microsoft.

Windows cannot verify the digital signature for this file. A recent hardware or software change might have installed a file that is signed incorrectly or damaged, or that might be malicious software from an unknown source.

To correct this, digital signature certificates have been updated. However, Ivanti recommends to apply updates to support Code Signing issues in Windows 7 and Windows Server 2008.

For windows 7, please make sure that System has SP1 installed and install KB from **https://technet.microsoft.com/en-us/library/security/3033929**.

As per MSDN driver signing has changed in Windows 10, version 1607.

Exceptions / Are cross-signed drivers still valid?

- Enforcement only happens on fresh installations, with Secure Boot on, and only applies to new kernel mode drivers:
  - PCs upgrading from a release of Windows prior to Windows 10 Version 1607 will still permit installation of cross-signed drivers.
  - PCs with Secure Boot OFF will still permit installation of cross-signed drivers.
  - Drivers signed with an end-entity certificate issued prior to July 29th, 2015 that chains to a supported cross-signed CA will continue to be allowed.

To prevent systems from failing to boot properly, boot drivers will not be blocked, but they will be removed by the Program Compatibility Assistant. Future versions of Windows will block boot driver.

To summarize, on non-upgraded fresh installations of Windows 10, version 1607 with Secure Boot ON, drivers must be signed by Microsoft or with an end-entity certificate issued prior to July 29th, 2015 that chains to a supported cross-signed CA.

Reference: **https://blogs.msdn.microsoft.com/windows_hardware_certification/2016/07/26/driver-signing-changes-in-windows-10-version-1607/**.

If the certificate uses SHA-2 or has SHA-2 certificates in its chain of trust and it is being used to sign kernel modules, then please be aware of KB3033929, an update for Windows 7 distributed through Windows Update. On versions of Windows 7 without this update, the kernel will reject signatures made with certificates that use SHA-2, so they cannot be used to get a kernel module to load.

Reference: **https://support.comodo.com/index.php?/Knowledgebase/Article/View/1103/0/signing-microsoft-windows-64-bit-kernel-mode-drivers**.

Remote Control Feature

For customers using the legacy Remote Control solution, this feature requires that your browsers enable either TLS 1.0 or TLS 1.1 to support the HTTPS connection between the clients and the remote control server.

Bandwidth and Latency

For optimal application performance, we recommend a latency of 110 ms or below with a minimum of 1.5 MB/sec in bandwidth between the Neurons for ITSM application server and the remote location client machines.

Chat Service Requirements

Software requirements

Component Name	Description
Redis	The Redis server should be running on a Linux container.
Microsoft SQL server	The database server should be available to store chat history and other statistics.
Node js	The node software should be installed where Neurons for ITSM installer is being installed. The node software is required to enable the server to run chat as a back-end service.

Points to Note:

- In the on-premise machine, Chat service is installed as a part of the appserver module installation while Neurons for ITSM installer is installed as software and Chat service runs on IIS.
- Currently installed on-prem node version: 23.2.0.
- Currently installed on-prem Redis server version : 6.2.5 or higher.

Redis server installation guide

For more details on Redis server installation, refer **Setting up Redis**.

3.2. About Third-Party Software

Neurons for ITSM supports the functionality provided by any third-party applications included in the Neurons for ITSM distribution, regardless of the vendor's support status. Some of the versions mentioned in the documentation for these third-party applications will be updated to the latest versions in subsequent releases.

Third-Party Software Components

The Neurons for ITSM installation package provides the following required third-party software components:

- Microsoft SQL Server 2014 Shared Management Objects
- Microsoft System CLR Types for Microsoft SQL Server 2014
- Microsoft Exchange Web Services Managed API 2.1
- IISNode v0.2.26
- Microsoft Access Database Engine 2016 Redistributable (x64)
- Microsoft .NET Core 8.0 Runtime
- URL Rewrite Module 2.1 (English)
- VC++ 2022 Redistributable (x64)
- VC++ 2013 Redistributable (x64)
- Node.js v24.4.1
- Microsoft Windows Server Roles and Features

> Ensure these Roles and Features are part of the operating system or install them manually.

- Redis 6.2.5 or later

> You have to install Redis manually. For more details, refer **Setting up Redis**.

- Microsoft .NET Framework 4.8 Full
- Server Manager - Windows Server Common (windows feature)
- Server Manager - Windows Server Common_WCF (windows feature)
- Server Manager - Windows Server Common_MSMQ (windows feature)

If the installer does not detect these components on the host system, they are installed automatically.

> If you create your own installation packages for remote deployment, include the software components that are listed in the system requirements that follow.

Third-Party Software Support

Ivanti relies on, and integrates with, many different third-party applications.

Ivanti Software regularly performs quality assurance testing in the interactions between Neurons for ITSM and these applications. Our policy is to support any tested version of a third-party application (as certified) and newer versions (as compliant), so long as the third-party vendor claims and maintains backward compatibility.

If Ivanti Software becomes aware that a third-party vendor has not maintained backward compatibility to the most recent certified version, Ivanti Software notes this in the compatibility matrix as an exception.

Neurons for ITSM does not support third-party applications that are older than a certified version. Ivanti Software often certifies more than one version of third-party software when more than one version is in common use; otherwise, the current generally available version is used during each Ivanti Software version quality assurance cycle.

3.3. Port Requirements

Ensure that ports are configured correctly so that all Neurons for ITSM components open correctly.

For All Deployments

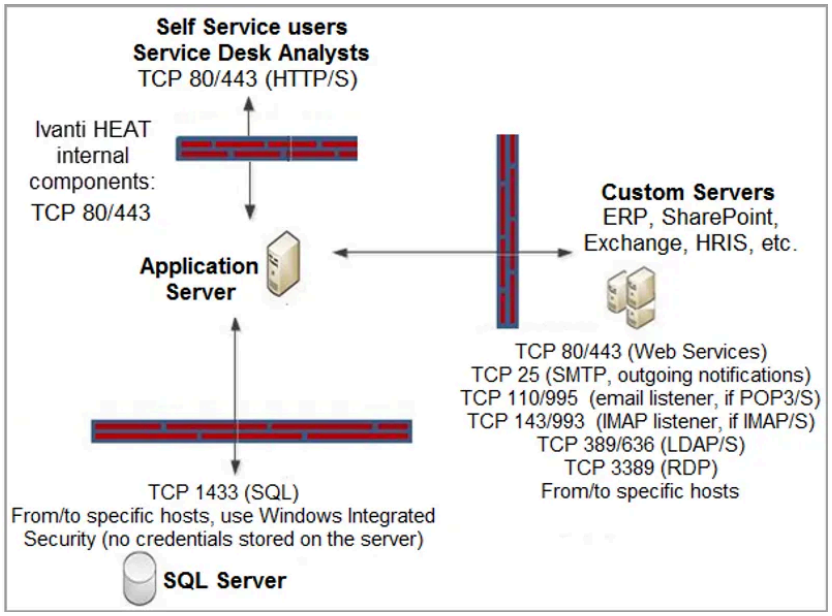
For all deployments, we recommend that you open the following ports:

Port	Service
53	For Active Directory. Select the UDP (for DNS) or TCP protocol.
139 or 445	For deploying agents to other computers. Select the UDP (for DNS) or TCP protocol.
443	For sending surveys. However, port HTTPS 7075 is supported until further notice.
1433	For the SQL server.
2323	Neurons for ITSM Voice web services port: Set to match the web services port set in the Ivanti Service Manager Voice management portal.
5743	For the Neurons for ITSM Voice server.
5986	For WinRM to receive remote commands.
8097	The default proxy port. Used unless defined in the endpoint address. For example, http://company/users:8113.
7200	For the message queue.
61000	The public TCP port of the Neurons for ITSM Voice message server service.
62000	The public TCP port of the Neurons for ITSM Voice reset password service.
6379	For the Redis Enterprise Software.
1213	For Integration Service.

For Deployments With All Components Installed on One Host

Port Configuration shows the port requirements for a deployment where all Neurons for ITSM components are installed on one host.

Fig.1. Port Configuration



The ports to open are as follows:

- Self Service users and Service Desk Analysts:
  - TCP 80/443 (HTTP/HTTPS)
- Neurons for ITSM application server (Neurons for ITSM internal components):
  - TCP 80/443
- Custom servers (ERP, SharePoint, Exchange, HRIS, and so on):
  - TCP 80/443 (web services)
  - TCP 25 (SMTP, outgoing notifications)
  - TCP 110/995 (Email listener if POP3/S is used)
  - TCP 143/993 (IMAP listener, if IMAP/S is used)
  - TCP 389/636 (LDAP/S)
  - TCP 3389 (RDP)
- Microsoft SQL Server:
  - TCP 1433 (SQL)

To and from specific hosts, use Windows Integrated Security. No credentials are stored on the server.

> The outbound port may not be the same as the inbound port. The outbound is determined by your environment.

For Deployments with the Web Server Installed on a Different Host

If your deployment has the Neurons for ITSM web server outside of the firewall, ensure that the following ports are also open:

- TCP 80/443
- TCP 25 (SMTP, outgoing notifications)
- TCP 1433 (SQL)

- TCP 54327 (License)
- TCP 389/636 (LDAP/S)

For Deployments that Include Discovery

> For additional information about the Discovery requirements, see the Neurons for ITSM online help and look for the topic called "Standard Gateway System Prerequisites". See "Related Documentation" for information on accessing the Neurons for ITSM documentation.

- About the Discovery Servers
- Port Information

About the Discovery Servers

The Discovery components reside on two servers: the Discovery web server and the Discovery application server. You can view the components that are deployed to each server from Microsoft IIS.

- The AgentTaskWs and ClientTransportProcessor components reside on the Discovery web server.
- The MessageSender, AssetProcessor, DiscoProcessor, and TaskProcessor components reside on the Discovery application server.

All communication from the gateway or client agent goes to the AgentTaskWs and ClientTransportProcessor components (on the Discovery web server), and those components then relay the messages or requests to the corresponding components on the Discovery application server.

Commands sent to the AgentTaskWs component are relayed to the TaskProcessor component via Net.TCP on port 7100.

Asset and Discovery data sent to the ClientTransportProcessor component is forwarded to the message sender via Net.TCP on port 5000. The message sender component then queues the message on one of the processor queues depending on the message type; you do not need to open a particular port for this.

Port Information

If your deployment includes Discovery, ensure that the following ports are also open for the

Discovery application server:

- TCP 8080
- TCP 5000
- TCP 8382
- TCP 7100

For Deployments with Remote Control

To use the remote control feature without the plugin (using HTML5), port 11438 must be opened. For more information about the remote control feature, see the Neurons for ITSM online help. (See "Related Documentation" for information about accessing the Neurons for ITSM documentation, including the online help.)

3.4. Compatibility Matrix

This compatibility matrix provides a comprehensive overview of the system components that required for the optimal deployment and operation of Neurons for ITSM.

See also **About Third-Party-Software**.

Legend

C	Has been fully tested by Quality Assurance (QA) and indicates compatibility (supported).
W	Has not been fully tested by QA for compatibility and/or integration, but works (supported).
N	Has been tested by QA for compatibility and/or integration but is NOT supported due to test results.
NT	Has not been tested by QA; is currently NOT supported due to lack of testing.
TOP	Has been fully tested by Quality Assurance (QA) for On-Premises only.
NTOP	Has not been fully tested by Quality Assurance (QA) for compatibility and/or integration, but works (supported) for On-Premises only.
--	Not applicable/not available (not supported).

Server Compatibility

Operating System	Neurons for ITSM Release 2025
Microsoft Windows Server 2016 English - EN-US*	NT
Microsoft Windows Server 2019 English - EN-US	C
Microsoft Windows Server 2022 English - EN-US	C
Microsoft Windows Server 2025 English - EN-US	C

* ITSM 2025.3 is the last version to support Microsoft Windows Server 2016 English.

Microsoft .NET Framework

	Neurons for ITSM Release 2025
Microsoft .NET Framework Release 4.6	NT
Microsoft .NET Framework Release 4.7	NT
Microsoft .NET Framework Release 4.8	C
Microsoft .NET Framework Release 4.8.1	C

Web Servers

	Neurons for ITSM Release 2025
Microsoft IIS Release 7.5	NT
Microsoft IIS Release 8.0	NT
Microsoft IIS Release 8.5	NT
Microsoft IIS Release 10	C

Database Management Software

	Neurons for ITSM Release 2025
Microsoft SQL Server 2016 *	NT
Microsoft SQL Server 2017	C
Microsoft SQL Server 2019	C
Microsoft SQL Server 2022 **	C

* ITSM 2025.2 is the last version to support SQL Server 2016.

** Customers using SQL 2022 who are considering migrating to a Cloud deployment of Ivanti Neurons for ITSM, may have to downgrade their SQL 2022 database to SQL 2019 prior to migrating.

- ITSM application does not support case-sensitive SQL Collation.
- ITSM application only supports the SQL Collation of SQL_Latin1_General_CP1_CI_AS.

Configuring a SQL Server Always On High Availability Group

ITSM supports the **SQL Server Always On High Availability Group** feature. To configure, refer the **Microsoft documentation**.

Database Management Software	Neurons for ITSM Release 2025
Microsoft SQL Server 2017	W
Microsoft SQL Server 2019	C
Microsoft SQL Server 2022	C

- After configuring the **SQL Server Always On High Availability Group**, make sure to rerun the System Configuration Wizard by providing availability group listener name in the **Configuration Database Server Name** section.
- The Ops Console doesn't work when the **SQL Server Always On High Availability Group** is enabled.
- When upgrading, make sure to remove the databases from the availability group. Once the upgrade is complete, you can add the databases back to the availability group.

Email Servers

	Neurons for ITSM Release 2025
Lotus Domino	NT
Novell GroupWise	NT
Microsoft Exchange 2010 (Email Server products that support POP3, SMTP, IMAP4)	C
Microsoft Exchange Web Service	C
Google Service Account	C
Microsoft Graph	C

Other Integrations

	Neurons for ITSM Release 2025
Ivanti Voice	C
Ivanti Service Manager DSM Release 2015.2	NT
Ivanti Service Manager DSM Release 2016.x *	W
Citrix	NT
VMWare ESX 4	W
Ivanti Service Manager Mobile	C

* DSM EoL (End of Life) is Dec 31, 2026

Browser Compatibility

> Neurons for ITSM versions 2022.1 and newer do not support Internet Explorer.

Browser	Supported Version
Google Chrome	Ivanti recommends and supports using the latest production version of Google Chrome (not including beta versions) with Neurons for ITSM. If new versions of Google Chrome break compatibility with Neurons for ITSM, we will document those exceptions here and work to support the new browser version with the next release. At time of release, Ivanti has successfully tested Ivanti Neurons for ITSM with Google Chrome version 137.0.7151.41. Download and install the **ClickOnce extension** to ensure that the Ivanti Neurons for ITSM reporting feature works correctly.
Mozilla Firefox	Ivanti recommends and supports using the latest production version of Mozilla Firefox (not including beta versions) with Neurons for ITSM. If new versions of Mozilla Firefox break compatibility with Neurons for ITSM, we will document those exceptions here and work to support the new browser version with the next release.
Microsoft Edge	Ivanti recommends and supports using the latest production version of Microsoft Edge (not including beta versions) with Neurons for ITSM. If new versions of Microsoft Edge break compatibility with Neurons for ITSM, we will document those exceptions here and work to support the new browser version with the next release.
Safari	Ivanti recommends and supports using the latest production version of Safari (not including beta versions) with Neurons for ITSM. If new versions of Safari break compatibility with Neurons for ITSM, we will document those exceptions here and work to support the new browser version with the next release.

Identity Providers

The following table lists the identity providers that have been tested, along with the supported authentication protocols.

Identity Provider	Authentication Protocol	Neurons for ITSM Release 2025
Google	OpenID Connect	C
Yahoo	OpenID Connect	C
Microsoft Entra ID	OpenID Connect	C
Microsoft Entra ID	ADFS / SAML	C
OKTA	ADFS / SAML	C
Microsoft ADFS	ADFS / SAML	C
On-Premises	Windows Integration Service (WIS)	TOP
On-Premises	OneLogin	NTOP

3.5. Troubleshooting - System Requirements

If you have problems with your installation or deployment, check this section first.

1

Problem: The Neurons for ITSM web servers do not work correctly.

Possible Cause: In multi-server environments, each Neurons for ITSM web server must meet the system requirements.

Solution: See "Web Server Requirements".

2

Problem: I am unable to use integrated components, such as Neurons for ITSM Voice.

Possible Cause: You must use the Microsoft Windows operating system when you use integrated components.

Solution: Change to the Microsoft Windows operating system. See "Client Computer Requirements".

3

Problem: I am using Google Chrome and cannot download or edit reports.

Possible Cause: You have not installed the ClickOnce extension yet.

Solution: Install the ClickOnce extension for Google Chrome. Navigate to **https://chrome.google.com/webstore/detail/meta4-clickonce- launcher/jkncabbipkgbconhaajbapbhokpbgkdc?hl=en** to download the extension. See "Client Computer Requirements".

4. Installation and Deployment

This document is intended for system administrators who are migrating from ITSM Release 6.x, ITSM Release 7.x, ITSM Release 8.x, and Ivanti Classic to Neurons for ITSM - Version 2025.3.

Document Organization

This guide contains the following sections:

- **Neurons for ITSM Deployment Overview**: Describes recommended and optional deployment architectures, including the Neurons for ITSM reporting feature and Ivanti Voice.
- **Neurons for ITSM Installation Prerequisites**: Provides information about confirming your role, the different accounts and passwords used in Neurons for ITSM, using a virtual machine, enabling file streaming and full-text search in Microsoft SQL Server, and verifying server roles and features.
- **Installing the Neurons for ITSM System**: Describes how to install Neurons for ITSM, including the Neurons for ITSM Operations Console, the reporting feature, ISM Discovery, and ISM Knowledge.
- **Initially Configuring the System**: Describes how to use the System Configuration Wizard to configure Neurons for ITSM, including its servers and databases; and how to use the Neurons for ITSM Operations Console to configure the deployment.
- **Upgrading Neurons for ITSM from an Earlier Release**: Contains information and instructions for upgrading Neurons for ITSM from an earlier release to the current release.
- **Using the License Manager**: Explains how to install and use the Neurons for ITSM License Manager.
- **Troubleshooting - Installation and Deployment**: Contains solutions to common problems.

4.1. Neurons for ITSM Deployment Overview

- Hardware Considerations
- Demonstration or Proof-of-Concept Deployment
- Minimum Production Deployment
- Enterprise Production Deployment
- Security Enterprise Production Deployment
- Database Deployment Options
- Operations Console Deployment Options
- Reporting Services Deployment Options
- Voice Deployment Options

Hardware Considerations

The following factors drive the hardware requirements for a new Neurons for ITSM deployment:

Neurons for ITSM supports separate landscapes: development (also called staging), test (also called UAT), and production. You can run these landscapes on the same server or isolate them on separate servers.
If you choose to run them on separate servers, you must provide additional hardware. Except for the demonstration or proof-of-concept deployment, we do not recommend putting all of the landscapes on one server.

- You can adjust the amount of hardware used after you deploy your system, after you have collected usage data. The Neurons for ITSM software is horizontally and vertically scalable, allowing you to add more hardware to increase performance without license changes.

4.1.1. Demonstration or Proof-of-Concept Deployment

- About the Demonstration or Proof-of-Concept Deployment
- Hardware Requirements for the Demonstration or Proof-of-Concept Deployment
- About Installing the Neurons for ITSM Components for the Demonstration or Proof-of-Concept Deployment

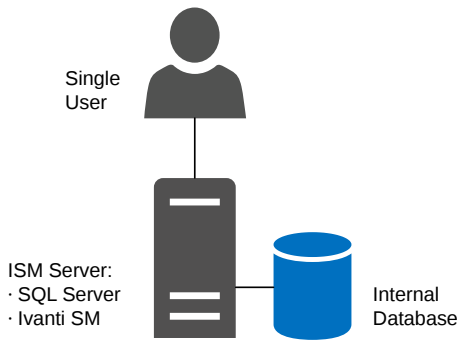
About the Demonstration or Proof-of-Concept Deployment

> This deployment is only recommended for demonstration purposes. Do not use this deployment for any production environment.

You can run the entire Neurons for ITSM system on a single server for the purpose of a demonstration or a proof-of-concept system.

Install Neurons for ITSM, with all options enabled, onto a single server that has Microsoft SQL Server loaded locally. Most users install this deployment in a virtual environment.

Example of Neurons for ITSM *and Databases on a Single Server*



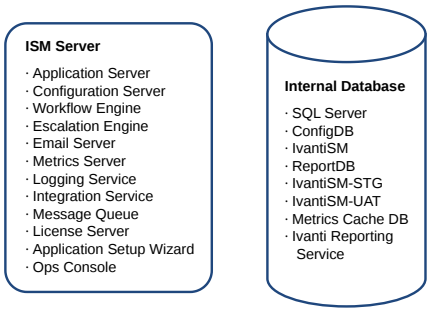
Hardware Requirements for the Demonstration or Proof-of-Concept Deployment

- 1 server, either virtual or physical
- 1 CPU
- 4 GB memory
- 40 GB hard drive

About Installing the Neurons for ITSM Components for the Demonstration or Proof-of-Concept Deployment

The below diagram shows the recommended locations of the various Neurons for ITSM software components in this deployment.

Neurons for ITSM *Components and Databases on a Single Server*

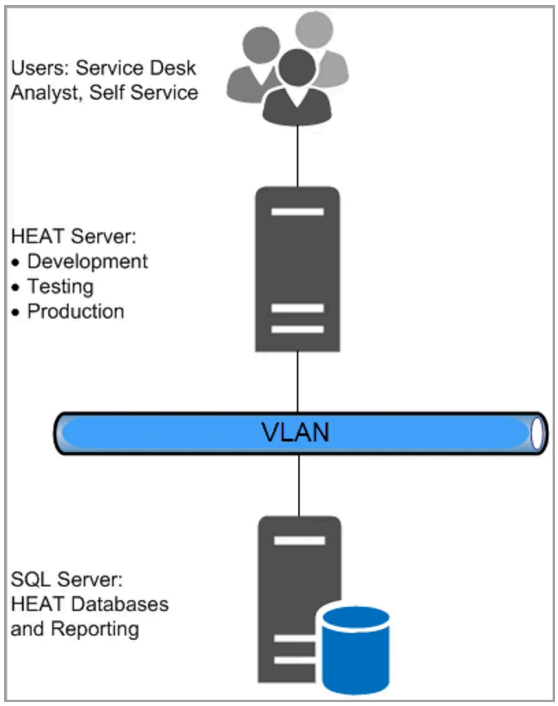


If you plan to include the Neurons for ITSM reporting feature in your deployment, you must install the Microsoft SQL Server Reporting Services (SSRS) component of Microsoft SQL Server. See **Installing Neurons for ITSM Reporting Feature**.

4.1.2. Minimum Production Deployment

Customers with 10 or fewer users often want to minimize their hardware investment. With the minimum production deployment architecture, you install all Neurons for ITSM components, except for the Neurons for ITSM databases, onto a single server. We recommend that you install the Neurons for ITSM databases onto a separate Microsoft SQL server.

Example of Neurons for ITSM *and Databases on Separate Servers*



In this deployment, all three landscapes (development [staging], testing [UAT], and production) run on the same server. As a result:

- Any issues created during the development cycle might affect production users.
- You cannot perform pre-upgrade testing.

Hardware Requirements for the Minimum Production Deployment

The following are the hardware requirements for the Neurons for ITSM server:

- One server, either physical or virtual
- 2 CPU
- 4 GB memory
- 200 GB hard drive

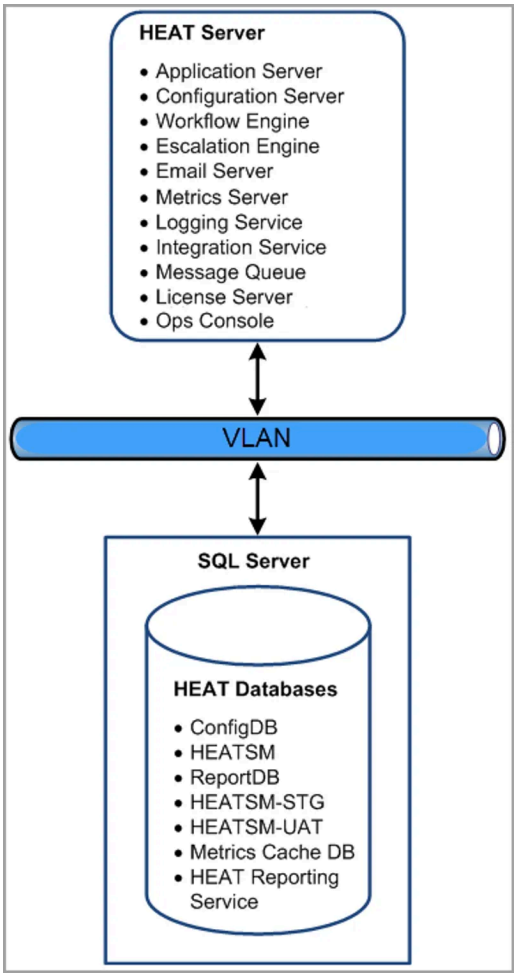
The following are the hardware requirements for the Microsoft SQL server where the databases reside:

- One server (we recommend a physical server)
- 2 CPU
- 8 GB memory
- 1 TB hard drive

About Installing the Neurons for ITSM Components for the Minimum Production Deployment

Neurons for ITSM Components and Databases on Separate Servers shows a diagram of the recommended locations of the various Neurons for ITSM software components in this deployment.

Neurons for ITSM *Components and Databases on Separate Servers*



4.1.3. Enterprise Production Deployment

Customers supporting large user populations of 100 or more are primarily concerned with availability and load management rather than with minimizing hardware expense. Customers with more than 100 users should use this setup as a baseline and then add the equivalent amount of hardware for each additional 100 users. The enterprise production deployment architecture splits Neurons for ITSM into:

- Front-end web servers
- Back-end processing servers

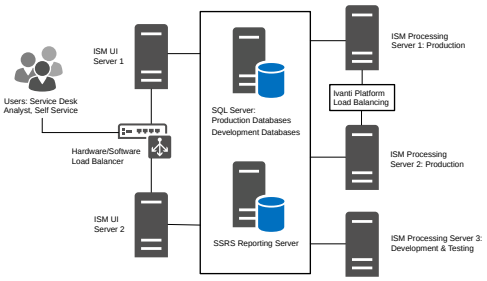
The back-end servers handle workflow, email processing, license allocation, inventory management, and so on. This architecture also separates the production environment from development and testing.

There are two key advantages of separating development and testing from production:

- Any issues created during the development cycle do not affect production users.
- You can perform upgrade testing on the development and testing servers before upgrading the production servers.

Optionally, you can install your development and test landscapes onto separate physical servers.

Example of Neurons for ITSM *in an Enterprise Deployment*



Hardware Requirements for the Enterprise Production Deployment

Web Servers

The following are the hardware requirements for the Neurons for ITSM web servers:

- Two virtual servers
- Outside the firewall
- Load balanced
- 4 CPU
- 16 GB memory
- 200 GB hard drive

Processing Servers (Production)

The following are the hardware requirements for the Neurons for ITSM processing servers used for the production landscape:

- Two virtual servers

- Inside the firewall
- Load balanced
- 4 CPU
- 16 GB memory
- 200 GB hard drive

Processing Servers (Development and Testing)

The following are the hardware requirements for the Neurons for ITSM processing server used for the development (staging) and testing (UAT) landscapes:

- One physical or virtual server
- 2 CPU
- 4 GB memory
- 200 GB hard drive

Microsoft SQL Server Requirements

Microsoft SQL Server is the database server for the deployment. Depending on your needs, you may have multiple database servers. The following are the requirements for the Microsoft SQL server where the databases reside:

- One physical server
- Microsoft SQL Server -
  - Microsoft SQL Server 2012 including Management Studio
  - Microsoft SQL Server 2012 SP2 including Management Studio
  - Microsoft SQL Server 2014
  - Microsoft SQL Server 2016
  - Microsoft SQL Server 2017
  - Microsoft SQL Server 2019
- Inside the firewall
- 8 CPU
- 24 GB memory
- 1 TB hard drive

> Customers may experience an error when Always On Availability is configured on the database server. Ivanti Service Manager currently does not support Always On Availability. For more details, see **Ivanti Community**.

Microsoft SSRS Reporting Server

The reporting feature requires that Microsoft SQL Server Reporting Services (SSRS) be installed and running on the database server that hosts the Neurons for ITSM reporting feature.

Microsoft SSRS can be installed on the Microsoft SQL server or on a separate Microsoft SSRS server, depending on your preference. We recommend a separate installation in the following situations:

- Multi-tenant environments, including managed service providers and enterprises with multiple Ivanti Service Manager tenants.
- The Microsoft SQL database is used also for applications other than Neurons for ITSM.
- Installations where the database server must remain online, because upgrading Microsoft SSRS often requires a reboot.

The following are the requirements for the Microsoft SQL Server Reporting Services (SSRS) reporting server:

- One physical server
- Microsoft SQL Server Release 2016
- Inside the firewall
- 8 CPU
- 24 GB memory
- 1 TB hard drive

Hardware Requirements when Adding Additional Users

For each additional 100 users, add this to architecture:

- Two additional Neurons for ITSM web servers.
- Two additional Neurons for ITSM processing servers.

System load determines the actual number of servers to add. You can add the web and processing servers on demand.

In most cases, you do not need to add more development (staging) or testing (UAT) servers.

High Availability and Load Balancing

You can achieve high availability through physical load balancing or by load balancing handled within Neurons for ITSM. For example, in the configuration shown in **Example of Neurons for ITSM in an Enterprise Deployment**. The Ivanti Service Manager components are connected to one or more load-balanced servers. In this example:

- The Neurons for ITSM web servers, which contain Neurons for ITSM components that are user-facing, all connect to a physical load-balancing server.
- The Neurons for ITSM process servers, which contain Neurons for ITSM components that are not user-facing, do not connect to a physical load-balancing server, but instead use a built-in load balancer. The escalation engine, workflow engine, and email server have built-in load-balancing mechanisms.
- The Neurons for ITSM web servers and process servers all connect to the same Ivanti Service Manager application database and configuration database. High availability for the Neurons for ITSM databases is achieved through a cluster.

Communication within the system, such as between an Neurons for ITSM process server and an Neurons for ITSM web server, does not require SSL. However, external communication can use SSL.

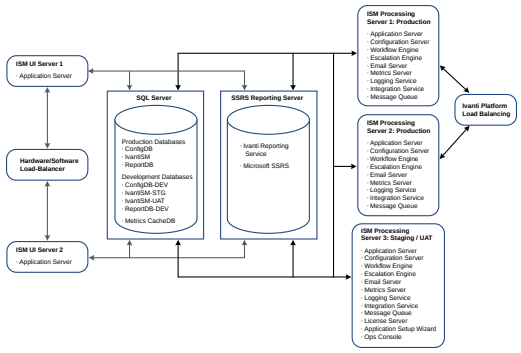
> By default, the **Use SSL** checkbox in all instances will be selected for new customers. New customers must provide the fully qualified domain name for the server location in the **SSL certificate** field. However, the **Use SSL** checkbox is not selected by default for the existing customer, and they have to select **Use SSL** if needed and provide the domain name. For more details on the **SSL Certificate**, see **Optional SSL Configuration**.

When using a hardware load balancer, turn on session persistence so that connections from the web browsers always get directed to the same web server.

About Installing the Neurons for ITSM for the Enterprise Production Deployment

Neurons for ITSM Components and Databases in an Enterprise Deployment shows a diagram of the recommended locations of the various Neurons for ITSM components.

Neurons for ITSM *Components and Databases in an Enterprise Deployment*



4.1.4. Security Enterprise Production Deployment

This deployment is based on the **Enterprise Production Deployment** with a DMZ added to provide security where users log in from outside of the company network.

We recommend the following DMZ configurations:

- DMZ with Neurons for ITSM web servers
- DMZ with reverse proxy servers

The DMZ is configured for authenticated access. When web servers are in the DMZ, each user must enter his user name and password to log into Neurons for ITSM. This architecture involves the additional cost of setting up and maintaining two firewalls.

Where you implement reverse proxy servers, you can add another layer of access authentication. This architecture involves the additional cost of setting up and maintaining two reverse proxy servers and a load balancer.

About Using a DMZ with Web Servers

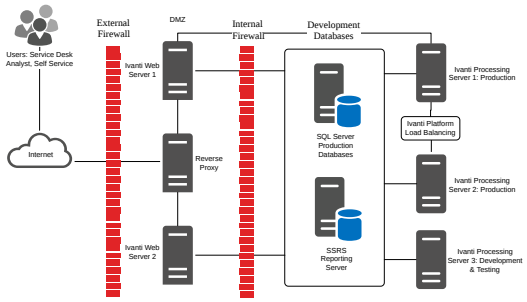
This option offers a greater level of security than placing your Neurons for ITSM web servers outside of a single company firewall. Placing a second firewall between the Internet and the web servers forms a semi-trusted network that prevents external access to your Neurons for ITSM process servers and databases.

We recommend that you harden the Neurons for ITSM web servers by taking the following actions:

- Disabling all unnecessary services
- Running necessary services with the lowest possible privileges
- Requiring strong passwords
- Locking an account after a certain number of login failures
- Deleting or disabling unnecessary user accounts, such as the guest user account
- Renaming or changing the description of the administrator account
- Installing the latest security updates and patches on the server
- Enabling security logging and checking the logs frequently

The same Neurons for ITSM components are installed on the web servers when they are located in the DMZ as when they are located outside of the company firewall. See **About Installing the Neurons for ITSM for the Enterprise Production Deployment**.

Example of an Enterprise Deployment with Neurons for ITSM *Web Servers in the DMZ*



About Using a DMZ with Reverse Proxy Servers

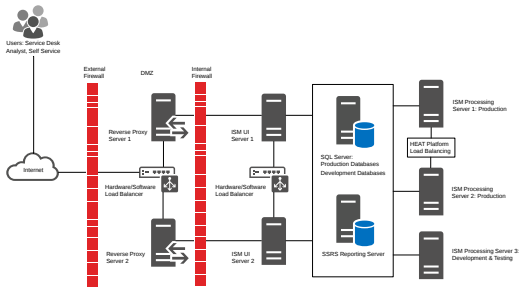
This option offers the greatest level of security by placing your web servers inside the company firewall. By placing reverse proxy servers in the DMZ, you prevent direct user login to the Neurons for ITSM web servers.

In addition, by locating the web servers on the same network as the process servers and databases, you achieve a true three-tier architecture.

The same Neurons for ITSM components are installed on the web servers when they are located inside the firewalls as when the servers are located outside.

This architecture involves the additional costs of servers to host the reverse-proxy service, as well as setting up and maintaining a second firewall. See **About Installing the Neurons for ITSM for the Enterprise Production Deployment**.

Example of an Enterprise Deployment with Neurons for ITSM *Reverse Proxy Servers in the DMZ*



4.1.5. Database Deployment Options

This section describes the different options available for the Neurons for ITSM databases.

- About the Databases Used in Neurons for ITSM
- Diagram Conventions
- Option D1: Separate Database Servers (Recommended and Best Practice)
- Option D2a: Single Database Server (Best Practice)
- Option D2b: Single Database Server
- Option D3: Single Configuration Database on a Single Database Server
- Option D4: Multiple Configuration Databases on a Single Server
- Option D5: Multiple Configuration Databases on Multiple Servers
- Option D6: Single Configuration Database on Separate Database Servers (for MSPs)
- Option D7: Multiple Configuration Databases on Separate Database Servers (for MSPs)

About the Databases Used in Neurons for ITSM

The following are the databases used in Neurons for ITSM:

- configuration database (ConfigDB)
- Neurons for ITSM application database (IvantiSM)
- Neurons for ITSM reporting database (ReportDB). This database is used for the reporting feature. See Reporting Services Deployment Options for more information.
- Metrics cache database (Metrics Cache DB). This database is used to cache the run-time state of the schedule jobs. Only one metrics cache database is needed in each isolated landscape. For example, if the configuration database and the Neurons for ITSM application database are hosted on the same database server, your deployment only needs one metrics cache database.

In addition, there are generally three instances for each tenant:

- Staging
- UAT
- Production

Each tenant instance, Staging (development), UAT (Test) and Production, can have its own database instance or server. These are called:

Database	Description
IvantiSM	Application database for production
IvantiSM-STG	Application database for staging (development)
IvantiSM-UAT	Application database for UAT (testing)
IvantiSM-Dev	Application database for staging (development) and UAT (testing)
ConfigDB	Configuration database for production
ConfigDB-STG	Configuration database for staging (development)
ConfigDB-UAT	Configuration database for UAT (testing)
ConfigDB-Dev	Configuration database for staging (development) and UAT (testing)
ReportServer	Report database for production
ReportServer-STG	Report database for staging (development)
ReportServer-UAT	Report database for UAT (testing)
ReportServer-Dev	Report database for staging (development) and UAT (testing)

The advantage of hosting a database on its own server is that it ensures security by isolating the production landscape. It also enhances performance. When you make multiple changes or upgrades to the staging or UAT landscapes, you do not want that to affect the performance of the production landscape. Separating the different databases makes it easier to upgrade the systems, because when you upgrade the systems you have to shut them down and that affects users in the production landscape.

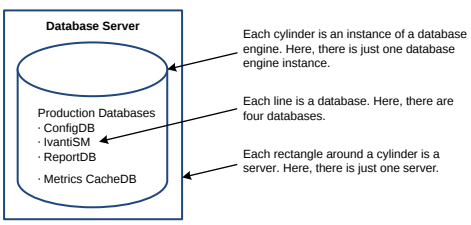
The advantage of hosting everything on one server is that you save money by only having to purchase and administer one server and one Microsoft SQL Server license.

Diagram Conventions

The following conventions are used in the diagrams:

- A line item is considered a database. There can be many databases within one database engine instance.
- A barrel is considered a database engine instance. It can contain multiple database instances.
- A square or rectangle around something means it is on one server (machine). Each server can potentially host multiple database engine instances.

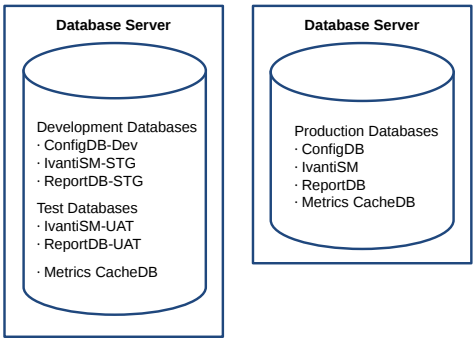
Diagram Conventions



Option D1: Separate Database Servers (Recommended and Best Practice)

In this deployment, there are two separate database servers. One is for the production landscape and the other is for the staging and UAT landscape. The staging and UAT servers share a configuration database.

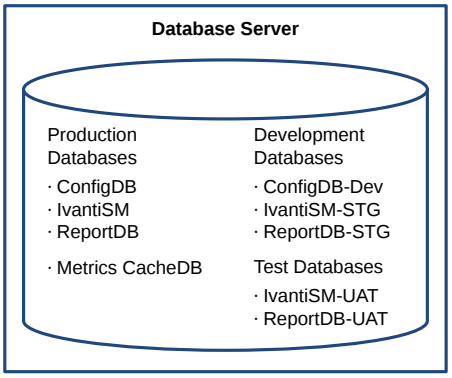
Example of Separate Database Servers



Option D2a: Single Database Server (Best Practice)

In this deployment, there is a single database server and one configuration database (ConfigDB-Dev).

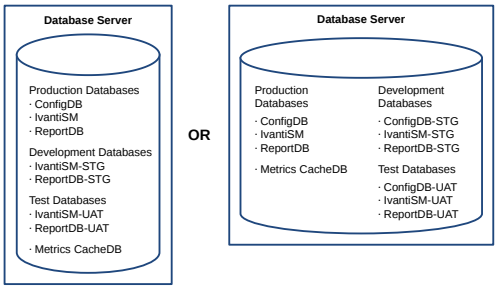
Example of Recommended Single Database Server



Option D2b: Single Database Server

In this deployment, there is a single database server and either no configuration database or two configuration databases (ConfigDB-STG and ConfigDB-UAT).

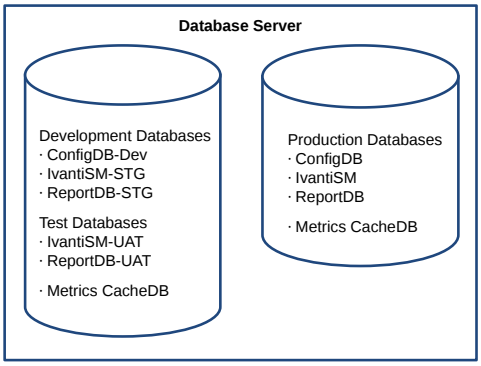
Example of Alternative Single Database Servers



Option D3: Single Configuration Database on a Single Database Server

In this deployment, there is one configuration database for the production landscape and another configuration database shared between the staging and UAT landscapes.

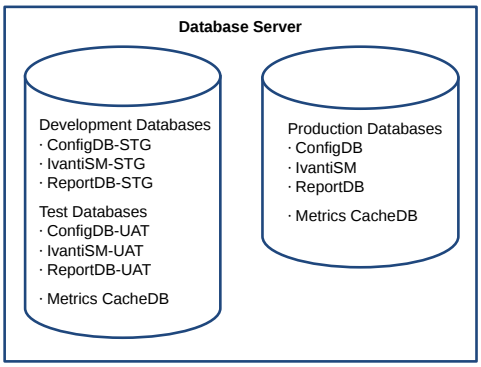
Example of Separate Database Engine Instances on a Single Database Server



Option D4: Multiple Configuration Databases on a Single Server

In this deployment, there are separate configuration databases for the production landscape, the staging landscape, and the UAT landscape.

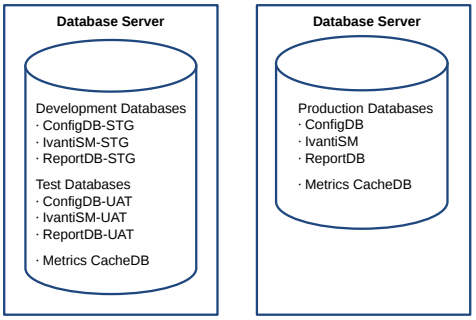
Example of Multiple Configuration Databases on a Single Server



Option D5: Multiple Configuration Databases on Multiple Servers

In this deployment, there are separate configuration databases for the production landscape, the staging landscape, and the UAT landscape and separate instances.

Example of Multiple Configuration Databases on Multiple Servers

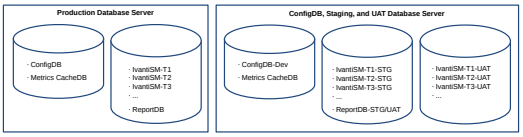


Option D6: Single Configuration Database on Separate Database Servers (for MSPs)

A Managed Service Provider (MSP) is a service partner who provides Neurons for ITSM for multiple tenants or outside customers.

In this deployment, there is a single configuration database shared between landscapes.

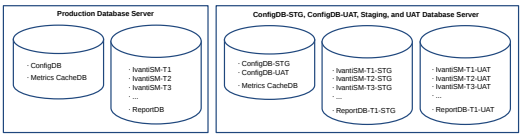
Example of a Single Configuration Database on Separate Database Servers



Option D7: Multiple Configuration Databases on Separate Database Servers (for MSPs)

In this deployment, there are separate configuration databases for the production landscape, the staging landscape, and the UAT landscape and separate instances.

Example of Multiple Configuration Databases on Separate Database Servers



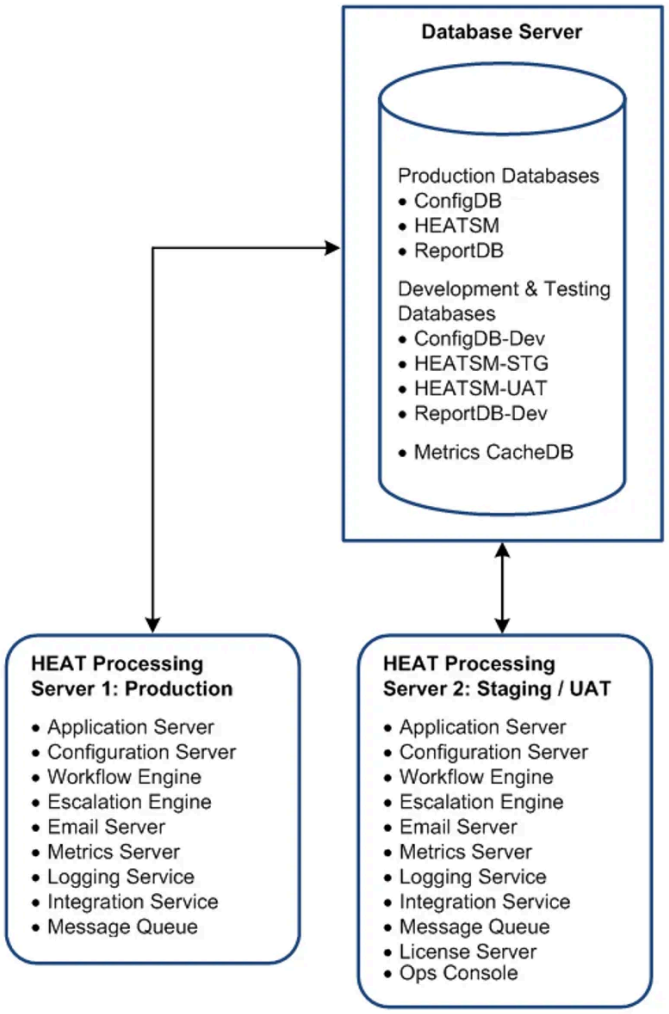
4.1.6. Operations Console Deployment Options

- Determine your database deployment options first. In all of the Operations Console deployment options below, you can substitute any of the database deployments described in **Database Deployment Options** for the database deployments in the figure. So for the database engine instance shown in Option OC1 below, you could substitute Option D1, Option D2a, Option D2b, Option D3, and so on for that database engine instance.
- To use the Operations Console, the production, staging, and UAT application servers must all use the same version of Neurons for ITSM. When you test new versions of Neurons for ITSM, either install the new version in a separate environment or plan the Operations Console pushes before testing the new version.
- We recommend using a deployment that has either two or three configuration databases, such as Option OC1, Option OC2, or Option OC3.

Option OC1: Separate Configuration Databases and Application Servers for Production and Development (Best Practice)

You only install the Operations Console on the staging/UAT server. Do not install it on any other servers.

Example of Separate Configuration Databases and Application Servers for Production and Development



Option OC2: Two Configuration Databases and Separate Application Servers for Each Landscape (Best Practice)

You only install the Operations Console on the staging server. Do not install it on any other servers.

Example of Two Configuration Databases and Separate Application Servers for Each Landscape

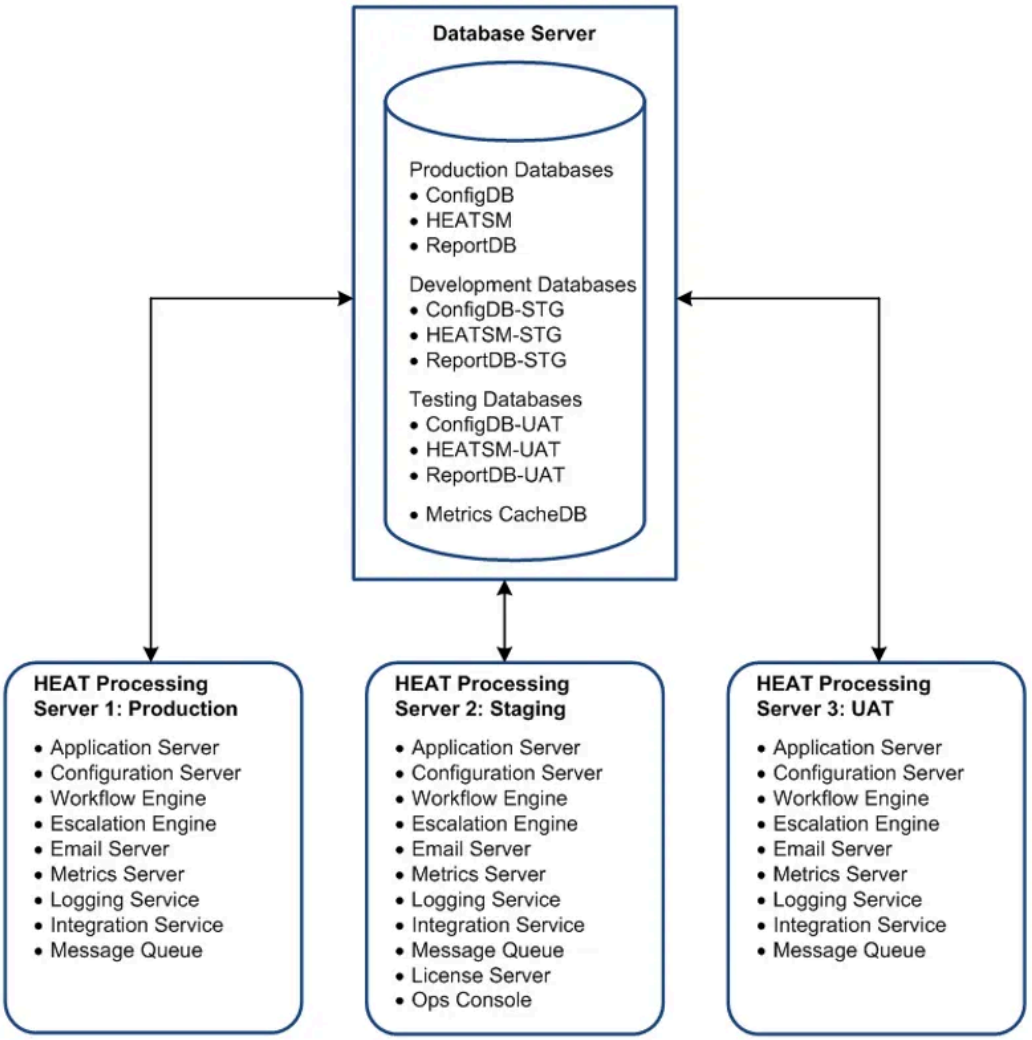


Option OC3: Separate Configuration Databases for Each Landscape (Best Practice)

> The only use case for this option is to upgrade the staging or UAT landscape at different times.

You only install the Operations Console on the staging server. Do not install it on any other servers.

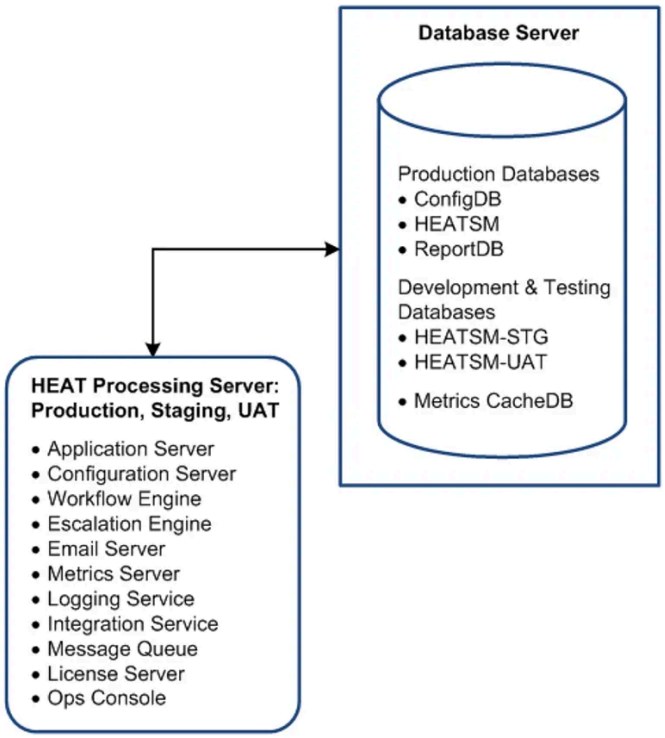
Example of Separate Configuration Databases for Each Landscape



Option OC4: One Configuration Database and One Application Server

We do not recommend using this or any similar deployment that has only one configuration database that is used for IvantiSM, IvantiSM-STG, and IvantiSM-UAT, unless you are setting up a demo environment. The reason for this only being viable for demo environments is that this deployment prevents you from being able to test future upgrades on development tenants without forcing production to upgrade at the same time.

Example of One Configuration Database and One Application Server

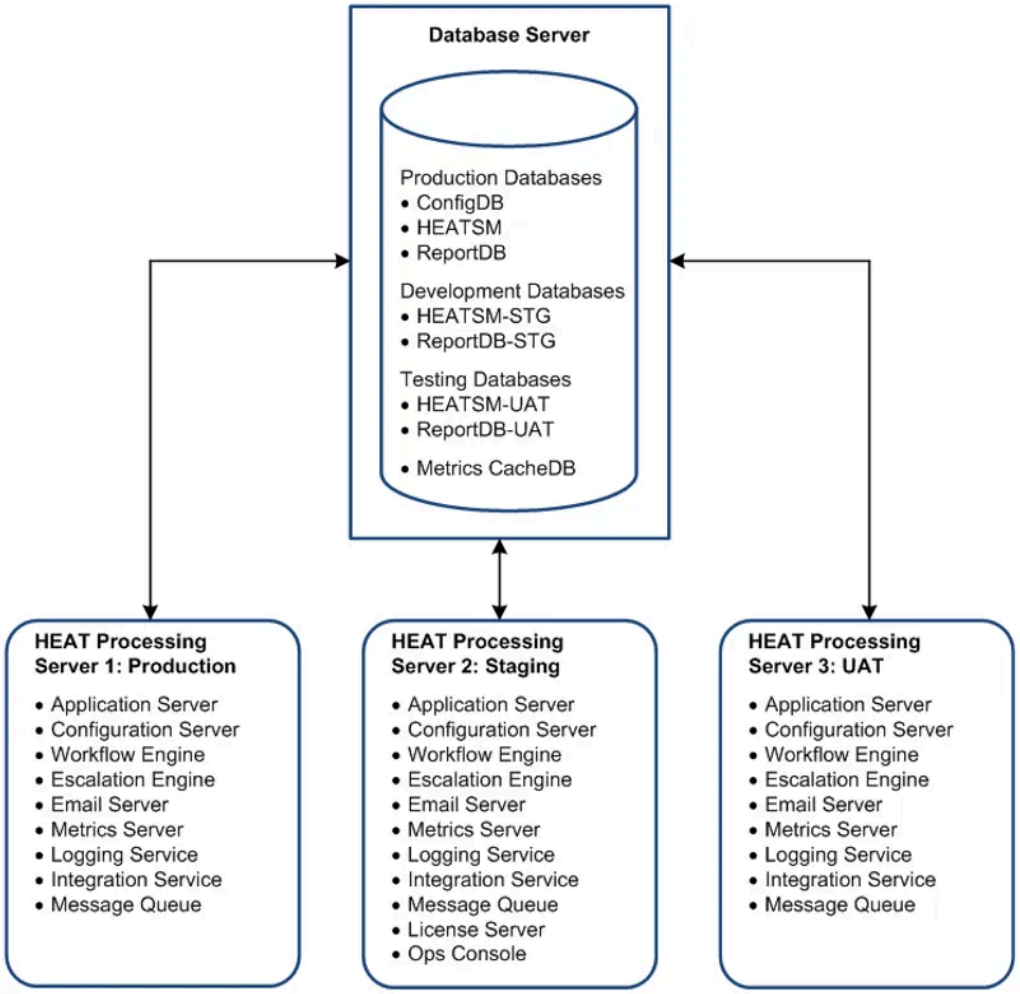


Option OC5: One Configuration Database and Separate Application Servers for Each Landscape

> We do not recommend using this or any similar deployment that has only one configuration database that is used for IvantiSM, IvantiSM-STG, and IvantiSM-UAT, unless you are setting up a demo environment. The reason for this only being viable for demo environments is that this deployment prevents you from being able to test future upgrades on development tenants without forcing production to upgrade at the same time.

You only install the Operations Console on the staging server. Do not install it on any other servers.

Example of One Configuration Database and Separate Application Servers for Each Landscape


Database Server
Production Databases
• ConfigDB
• HEATSM
• ReportDB
Development Databases
• HEATSM-STG
• ReportDB-STG
Testing Databases
• HEATSM-UAT
• ReportDB-UAT
• Metrics CacheDB
HEAT Processing Server 1: Production
• Application Server
• Configuration Server
• Workflow Engine
• Escalation Engine
• Email Server
• Metrics Server
• Logging Service
• Integration Service
• Message Queue
HEAT Processing Server 2: Staging
• Application Server
• Configuration Server
• Workflow Engine
• Escalation Engine
• Email Server
• Metrics Server
• Logging Service
• Integration Service
• Message Queue
• License Server
• Ops Console
HEAT Processing Server 3: UAT
• Application Server
• Configuration Server
• Workflow Engine
• Escalation Engine
• Email Server
• Metrics Server
• Logging Service
• Integration Service
• Message Queue

4.1.7. Reporting Services Deployment Options

Determine your database deployment options first (see **Database Deployment Options**), and then your Operations Console deployment options (see Neurons for ITSM **Operations Console Deployment Options**). Then determine your reporting services deployment options.

In all of the reporting services deployment options below, note the following:

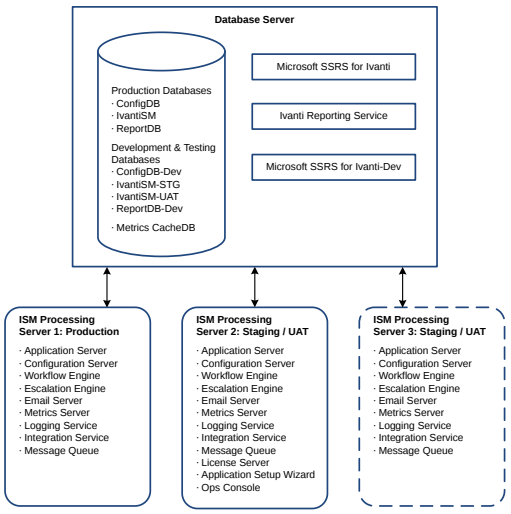
- You can substitute any of the database deployments described in **Database Deployment Options** for the database deployments in the figure. For example, for the database engine instance shown in Option R5 below, you could substitute Option D1, Option D2a/b, Option D3, and so on for that database engine instance.
- You can substitute any of the Operations Console deployments described in Neurons for ITSM **Operations Console Deployment Options** for the Operations Console deployments in the figure. For example, for the single application server shown in Option R5 below, you could substitute Option OC3 for that application server.

- The Neurons for ITSM report database can reside on a separate database instance, or on a database instance on a separate server.
- The Microsoft SSRS instance can run on the Neurons for ITSM database server, application server, or on a separate server.
- The reporting feature should be installed on the Microsoft SSRS Server.
- We recommend using either Option R1 or Option R2.

Option R1: Multiple Microsoft SSRS Instances on the Same Database Server (Best Practice)

This deployment is the most common and recommended for simplicity and security. The advantage of this deployment is that you have a reporting feature for production that is separate from the development environments.

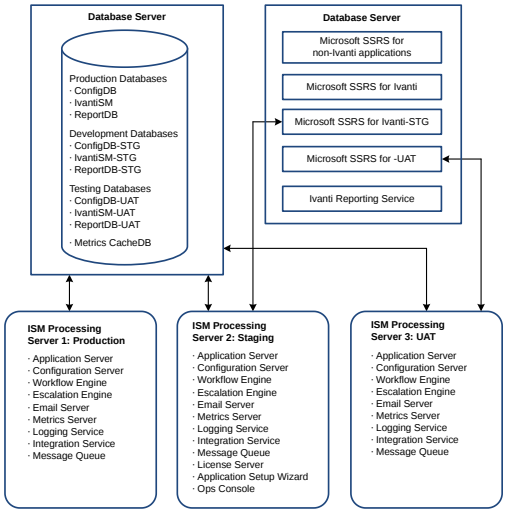
Example of Multiple Microsoft SSRS Instance on One Database Server



Option R2: Separate Microsoft SSRS Instances and a Separate Database Server (Best Practice)

Use this deployment to keep the reporting separate between the production and development environments.

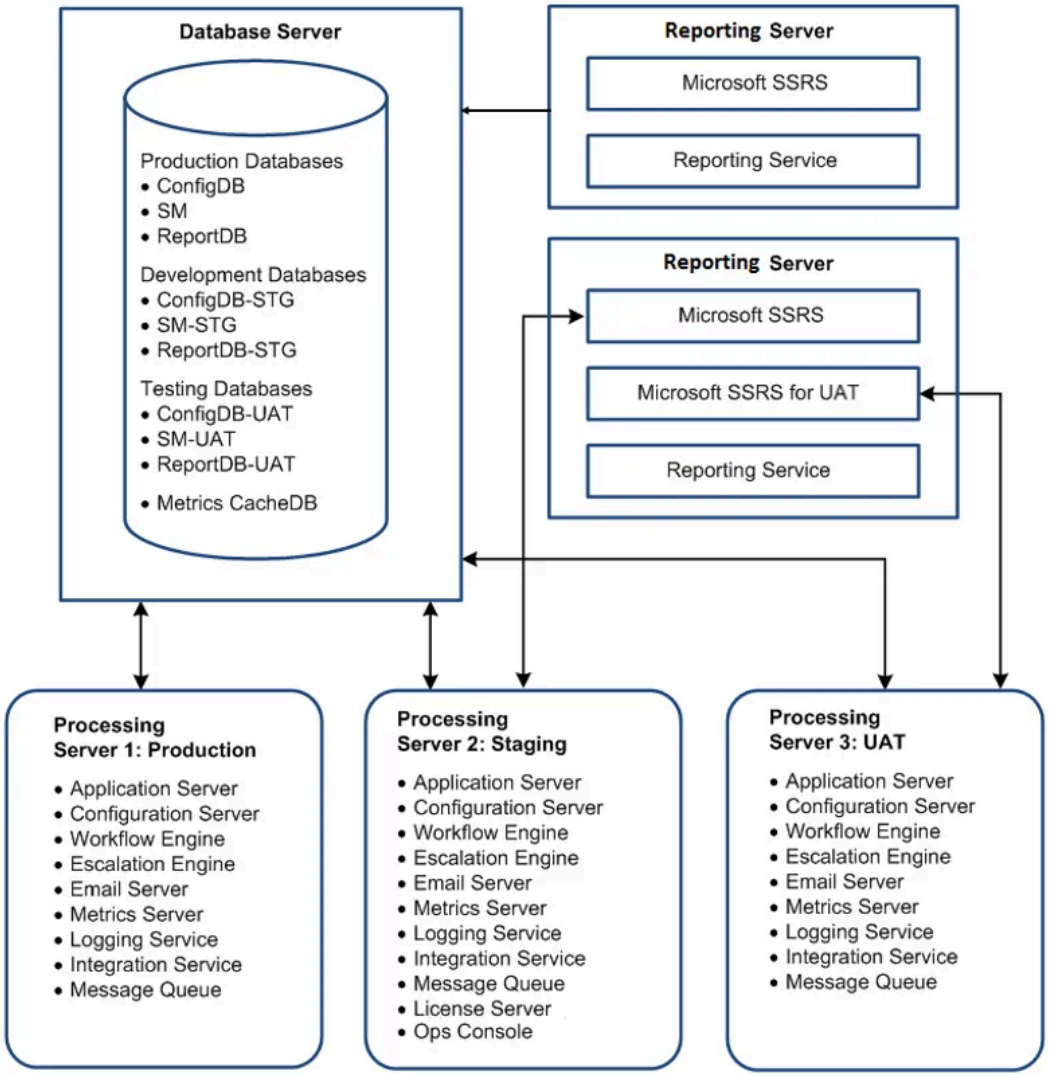
Example of Separate Microsoft SSRS Instances and a Separate Database Server



Option R3: Separate Microsoft SSRS Instances

Use this deployment to keep the reporting separate between the production and staging/UAT environments.

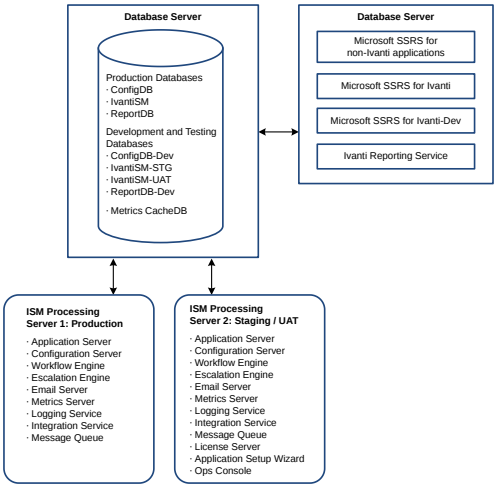
Example of Separate Microsoft SSRS Instances



Option R4: Multiple Microsoft SSRS Instances on a Separate Report Server

This deployment is recommended if you need to maintain your database in a locked down environment and keep it separate from the development environment. We also recommend this deployment if you already have a server with a Microsoft SSRS instance for other applications.

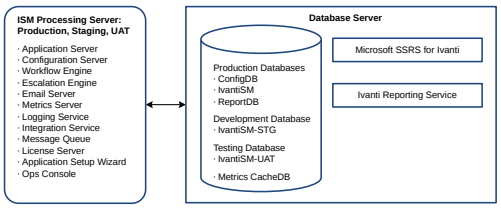
Example of a Multiple Microsoft SSRS Instances on a Separate Report Server



Option R5: Single Microsoft SSRS Instance on One Database Server

> We do not recommend using this deployment unless you are setting up a demo environment.

Example of a Single Microsoft SSRS Instance on One Database Server



4.1.8. Voice Deployment Options

- About Ivanti Voice
- About this Deployment
- Installing and Configuring Ivanti Voice
- Integrating Ivanti Voice with Neurons for ITSM

About Ivanti Voice

Ivanti Voice is the telephony application that integrates automated call routing and management and computer telephone integration to Ivanti Software applications such as Neurons for ITSM. Ivanti Voice uses the next-generation, standards-based IP communication transport, called Session Initiation Protocol (SIP). Neurons for ITSM and Voice are typically installed and maintained on separate servers..

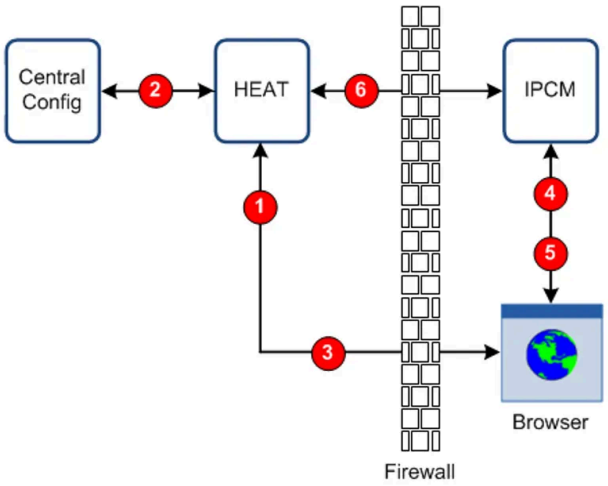
About this Deployment

- Agent Login Sequence
- Incoming Call Sequence

Agent Login Sequence

Agent Login Sequence shows the Voice agent login sequence.

Agent Login Sequence

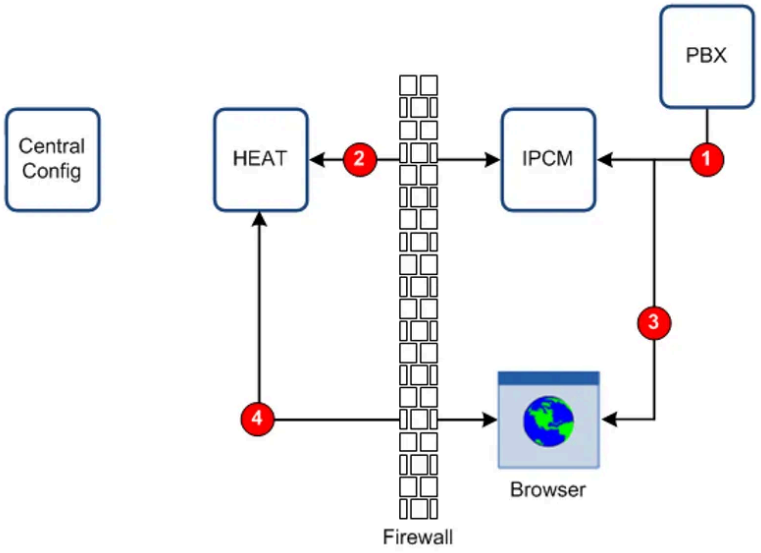


1. The agent logs into a secure (HTTPS) connection.
2. The system verifies the Voice server IP address or domain name.
3. The system obtains the session key from the Ivanti Voice server IP address or domain name.
4. The system verifies the session key (TCP 5743).
5. The system makes the phone connection (SIP UDP 5060).
6. The system validates the session key (HTTPS).

Incoming Call Sequence

Incoming Call Sequence shows the sequence of events for Ivanti Voice incoming calls.

Incoming Call Sequence



1. The system receives a call (SIP UDP 500).
2. The IVR searches and updates the record (HTTPS).
3. The call is forwarded to the agent (SIP UDP 5060, RTP UDP, TCP 5743).
4. The record appears to the agent (HTTPS).

Installing and Configuring Ivanti Voice

For information about installing and configuring Ivanti Voice, see the *IP Communications Management Administrator Guide*.

Integrating Ivanti Voice with Neurons for ITSM

For complete information about integrating Voice with Neurons for ITSM, see *Working with Voice* in the Neurons for ITSM online help.

4.2. Neurons for ITSM Installation Prerequisites

Before you install Neurons for ITSM, do the following:

- Confirm your role. See **About Roles**.
- Create the needed accounts. See **About the Different Accounts Used in Neurons for ITSM**.
- Meet the system, hardware, and software prerequisites. This information is contained in the *System Requirements and Compatibility Matrix for* Neurons for ITSM.
- Configure the ports needed for your deployment. This information is contained in the *System Requirements and Compatibility Matrix for* Neurons for ITSM.
- If you are going to deploy Neurons for ITSM as a virtual image, review the requirements at **About Using a Virtual Machine with Neurons for ITSM**.
- Ensure that the FILESTREAM feature is enabled on the Microsoft SQL Server. See **Enabling Attachment File Streaming**.
- If you are going to deploy the reporting feature, ensure that Microsoft SQL Server Reporting Services (SSRS) is enabled. See **Installing Neurons for ITSM Reporting Feature**.
- Enable full-text search for Microsoft SQL Server. See **Full Text Search** .
- (Optional) Verify the server roles and features described in **Verifying Server Roles and Features**.
- Determine the directories in which to install the Neurons for ITSM components on the host that you are logged into.
- Install the following from the installer package:
  - Prerequisite Installer
  - System Configuration Wizard (SCW)
  - Additional App Domain shared in the package - do this post the SCW installation.

4.2.1. About Roles

- Administrator Account Permissions
- Database Access Rights Needed for Neurons for ITSM
- Database Access Rights Needed for the Reporting Feature

Administrator Account Permissions

Use your account that has local administrator permissions to install Neurons for ITSM, including all optional components. This administrator account must have permission to create and modify folders, files, and registry keys.

Database Access Rights Needed for Neurons for ITSM

The following are the minimum access rights required for the account of the user running the System Configuration Wizard:

Role	Access Right
Server	public
	dbcreator
	securityadmin NOTE: This role is only needed temporarily, to assign the db_owner role to the configuration database and the Neurons for ITSM databases.

If you do not want to give this permission to the user running the System Configuration Wizard, you must go into the Microsoft SQL Server Management Studio and manually assign the service account as the db_owner for the configuration database and the Neurons for ITSM databases.

Database Access Rights Needed for the Reporting Feature

The following are the minimum access rights used in Microsoft SQL Server Management Studio for Neurons for ITSM when also using the reporting feature:

Role	Access Right
Server	public
	dbcreator
	securityadmin *
Database	db_owner of master
	db_owner of msdb
* The securityadmin role, and the db_owner of the master Microsoft database, are only needed if you set up Microsoft SSRS and create a new report server database. If you have already configured Microsoft SSRS and do not make any changes to it, you do not need this role. The securityadmin role creates the following Microsoft SSRS-related roles in the master, msdb, report server, and report server temporary databases: • RSExecRole • SQLAgentOperatorRole • SQLAgentReaderRole • SQLAgentUserRole	

The service account for Microsoft SSRS requires:

Database	Role
MSDB	RSExecRole
	SQLAgentOperatorRole
	SQLAgentReaderRole
	SQLAgentUserRole
Master	RSExecRole
Report server	RSExecRole
	db_owner
Report server temporary	RSExecRole
	db_owner

4.2.2. About the Different Accounts Used in Neurons for ITSM

Before you install and configure Neurons for ITSM, you must create and know the credentials for the following accounts:

- IT Account
- Database Account
- Service Account

During configuration of Neurons for ITSM, you create the following accounts:

- Administrator Account for the Configuration Database
- Administrator Account for Neurons for ITSM
- Administrator Account for the Reporting Service
- Administrator Account for the Operations Console

IT Account

This is the account that you are logged into when installing Neurons for ITSM and running the System Configuration Wizard.

You must create this account before you start the installation. This account must have administrator privileges on the servers on which you install Neurons for ITSM. Your IT department sets up this account for you.

Database Account

You must create this account before you start the installation.

You enter your credentials for the Neurons for ITSM database account on the following System Configuration Wizard pages:

- **Configuration Application**
- **Ivanti Service Manager Application**
- **Metrics Server**
- **Ivanti Discovery Application Server**

If you plan on installing the reporting feature, you also enter your credentials for the Neurons for ITSM database account on the following System Configuration Wizard pages:

- **Microsoft SSRS Configuration**
- **Reporting Service Configuration**

Service Account

You must create this account before you start the installation and configuration and it must have administrator privileges on the Neurons for ITSM application server.

A Neurons for ITSM service account is only required when you:

- Do not use the local system account for the Microsoft IIS application pool identity and Windows service.
- Install the reporting feature.

You enter your credentials for the Neurons for ITSM service account on the following System Configuration Wizard pages:

- **Ivanti Application Server Settings**
- **Microsoft SSRS Configuration**

Administrator Account for the Configuration Database

This account enables you to log into the configuration database.

You create this account during system configuration on the **Configuration Application** page of the System Configuration Wizard.

There will probably be different accounts used for different tenants.

Administrator Account for Neurons for ITSM

This account enables you to log into Neurons for ITSM, with access to the Service Desk Console and Configuration Console.

You create this account during system configuration on the **Ivanti Service Manager Application** page of the System Configuration Wizard.

There will probably be different accounts used for different tenants.

Administrator Account for the Reporting Service

This account enables you to log into the reporting feature.

You create this account during system configuration on the **Reporting Service Configuration** page of the System Configuration Wizard.

Administrator Account for the Operations Console

This account enables you to log into the Operations Console.

You create this account during system configuration when you check Use this host for Operations Console on the **Ivanti Application Server Settings** page of the System Configuration Wizard.

4.2.3. About Passwords Used in Neurons for ITSM

We recommend that you follow industry standards for creating strong passwords.

For more information, refer to **Microsoft - Strong Passwords** and **Microsoft - Password Policy.**

4.2.4. About Using a Virtual Machine with Neurons for ITSM

You can also deploy Neurons for ITSM as a virtual image. If you use a virtual machine, always keep the same universally unique identifier (UUID).

Refer to the documentation specific to the virtual machine you are using for more information.

> Any virtual machine platform that meets the **System Requirements** is compatible with Neurons for ITSM.

4.2.5. Enabling Attachment File Streaming

- About Enabling Attachment File Streaming
- Task 1: Enabling the FILESTREAM Feature
- Task 2: Setting up Neurons for ITSM
- Task 3: Running the File Stream Attachment Script

About Enabling Attachment File Streaming

Microsoft SQL Server 2022 and later versions come with enhanced support for unstructured data in the form of the FILESTREAM data type, which stores unstructured data files in the NTFS file system while storing the metadata about the unstructured data files in the Microsoft SQL Server database. For more information, see https://technet.microsoft.com/en-us/library/bb933993(v=sql.105).aspx.

Beginning with Release 2016.1.1, Neurons for ITSM supports the FILESTREAM attachment type when you perform the tasks below.

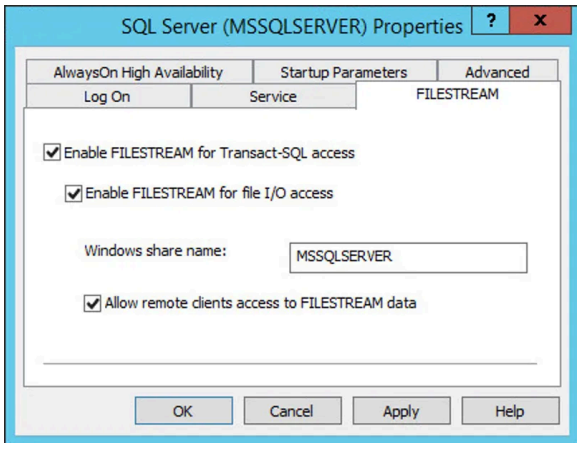
> This procedure is only required for deployments that store FILESTREAM attachments.

Task 1: Enabling the FILESTREAM Feature

You must enable the FILESTREAM feature on the Microsoft SQL Server instances where you plan to install the configuration database and the Neurons for ITSM application database. The database administrator normally performs this task.

1. On the database server, go to the Windows apps menu and click SQL Server Configuration Manager.
2. Navigate to the **SQL Server Services** node and double-click the Microsoft SQL Server instance to modify. For example, select SQL Server (MSSQLSERVER).

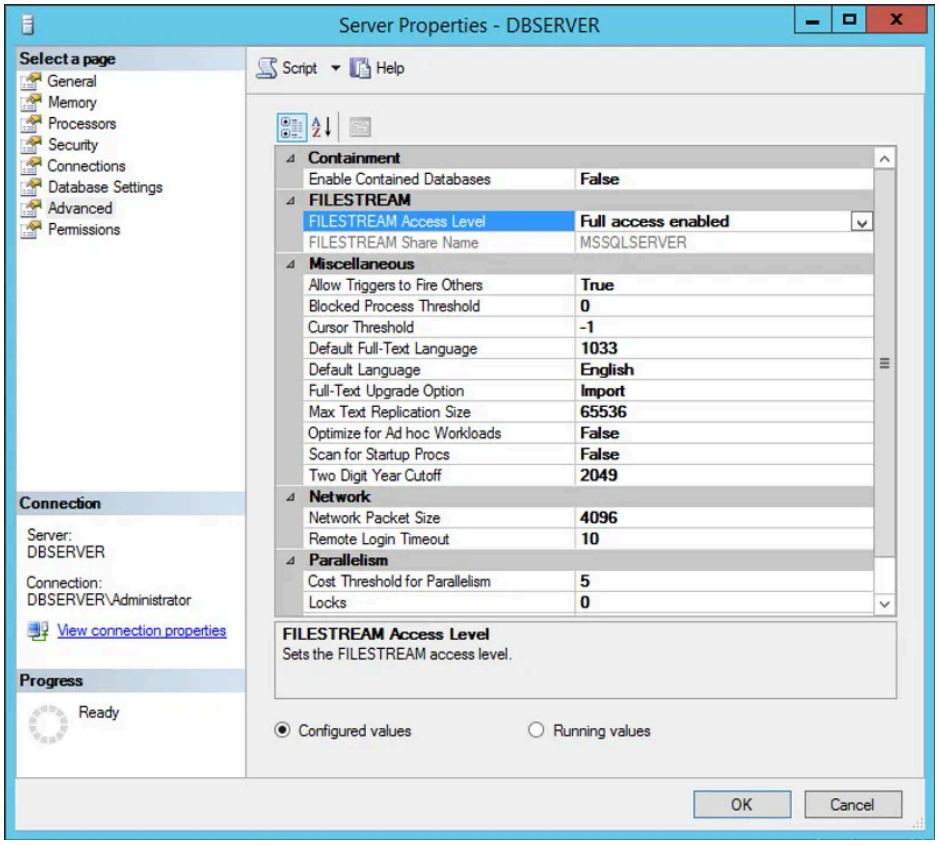
   Microsoft SQL Server Properties Dialog Box

3. Click the **FILESTREAM** tab.
4. Check Enable FILESTREAM for Transact-SQL access and Enable FILESTREAM for file I/O access.
5. Enter a Windows share name for the files.
6. Check Allow remote clients access to FILESTREAM data.
7. Click OK.

8. On the database server, go to the Windows apps menu and click SQL Server Management Studio.
9. In Microsoft SQL Server Management Studio, connect to the database server.
10. Right-click the server and choose Properties from the pop-up menu.
11. Under **Select a page**, click Advanced.
12. Double-click the entry called FILESTREAM Access Level until it displays Full access enabled.
13. Click OK.
14. (Optional) If you see a restart required message, restart the Microsoft SQL Server.

Microsoft SQL Server Properties



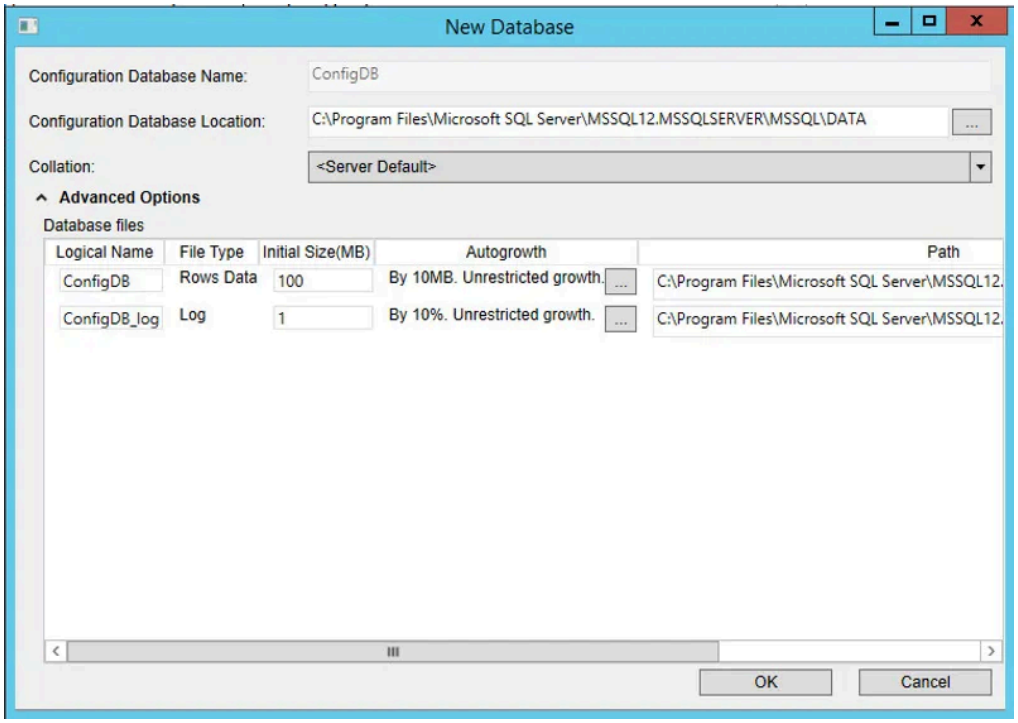
Task 2: Setting up Neurons for ITSM

The Neurons for ITSM application database and the configuration database are Microsoft SQL databases. On Microsoft SQL databases, user accounts with the db_owner fixed database role can perform all configuration and maintenance activities on the database.

These steps are in addition to those described under **Configuring Neurons for ITSM** .

1. On the database server, go to the Windows apps menu and click System Configuration Wizard.
2. On the **Configuration Application** page, specify a configuration database user account that has a db_owner role on Microsoft SQL Server. This is for the **User Name** field. See **Configuring the Configuration Database**.
3. When you create a new configuration database, specify the directory path to the MDF and LDF database files.

Creating a New Configuration Database



4. On the Neurons for ITSM **Application** page, specify a Neurons for ITSM application database user account that has a db_owner role on Microsoft SQL Server. This is for the **User Name** field. See **Configuring the Neurons for ITSM Application**.
5. When you create a new Neurons for ITSM application database, specify an additional path to the server.

Task 3: Running the File Stream Attachment Script

The FileStreamAttachment.sql script is included on the Software product CD. The script creates an attachment filestream database file group and a filestream.

Before you run the script, open it in Microsoft SQL Server Management Studio and read the script notes carefully, so that you understand what the script does.

1. On the Microsoft SQL Server hosting the configuration database, run the FileStreamAttachment.sql script.
2. On the Microsoft SQL Server hosting the Neurons for ITSM application database, run the FileStreamAttachment.sql script.

When you create your Neurons for ITSM application database, you can now specify the FILESTREAM attachment type.

4.2.6. SSL Configuration

This section explains how to configure Secure Sockets Layer (SSL) on Internet Information Services (IIS) for first-time users.

The configuration of SSL for a site includes the following steps:

- Create an SSL binding on a site.
- Verify the SSL binding by making a request to the site.
- Configure SSL options in IIS by making SSL a requirement for the site.

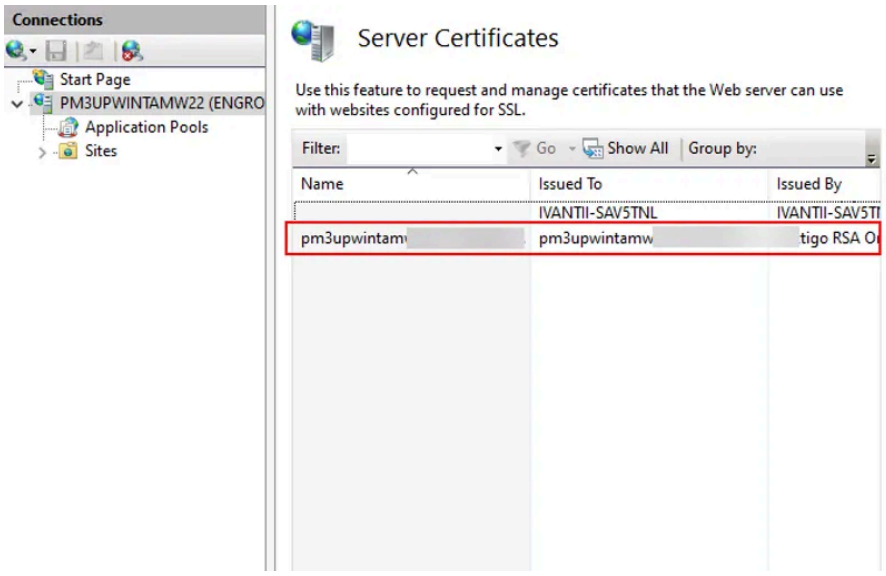
Prerequisites

- Install Microsoft IIS in your virtual machine or computer.
- Obtain appropriate certificate for SSL encryption and for authenticating the identity of the server.

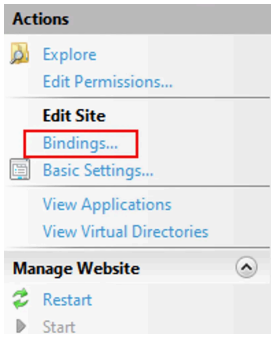
Create an SSL binding

1. Open **Internet Information Services (IIS) Manager**, and open **Server Certificates**.
2. Click **Import**, and upload the certificate.

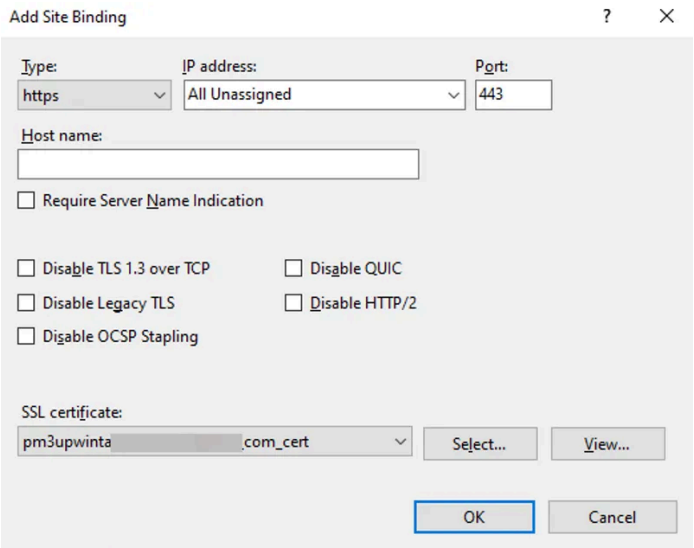
   After the certificate is imported, it appears on the **Server Certificates** page.



3. In the **Connections** pane, go to **Sites** > **Default Web Site**.
4. In the **Actions** pane, click **Bindings** to create SSL binding. The Site Bindings window opens.



5. Click **Add**, and enter the required information:



Field	Description
Type	Select **https** from the dropdown menu.
IP address	Select **All Unassigned** from the dropdown menu.
Port	Enter *443*.
Host name	Enter the host name of your virtual machine or computer.
SSL certificate	From the dropdown menu, select the certificate that you uploaded in Server Certificates.

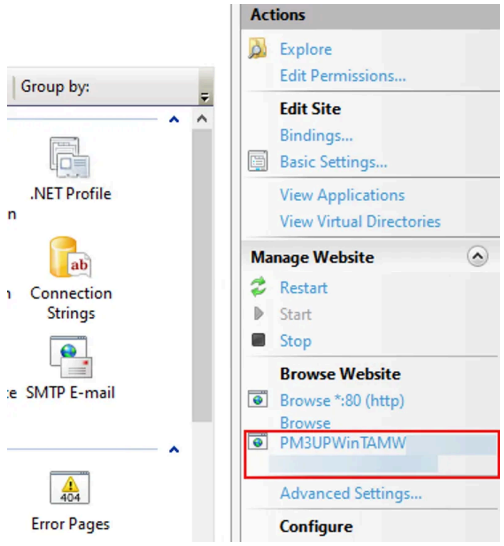
6. Click **OK**.

   IIS creates a new SSL binding for your site.

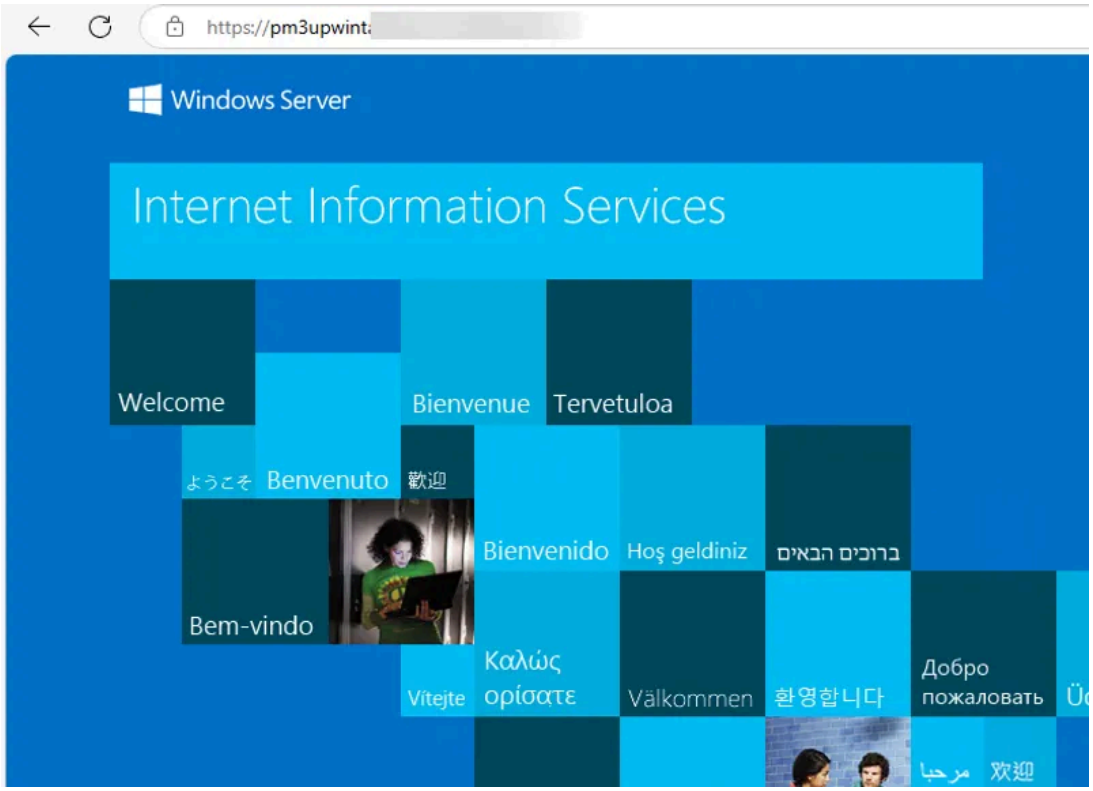
Next, you can verify whether the SSL binding works.

Verify the SSL binding

1. In the **Connections** pane, go to> **Sites** > **Default Web Site**.
2. In the **Actions** pane, under **Browser Website**, click the site you configured.

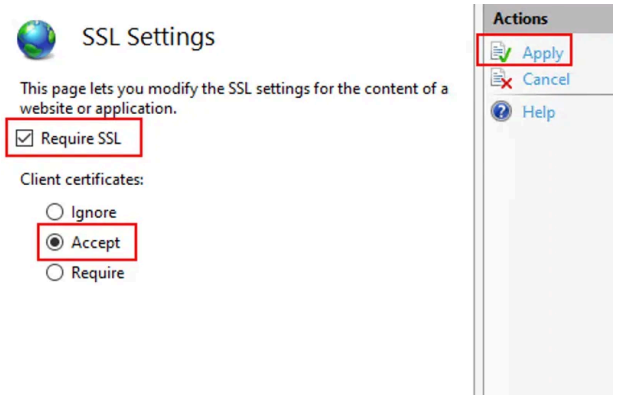


The site opens in your default browser. If SSL is configured correctly, the URL begins with *https*, indicating a secured connection.



Configure SSL options in IIS

1. Go to **Connections** pane > **Sites** > **Default Web Site**.
2. On the **Default Web Site Home** page, open **SSL Settings**.
3. Select the **Require SSL** checkbox.
4. Select the **Accept** button for Client certificates.



5. Click **Apply**.

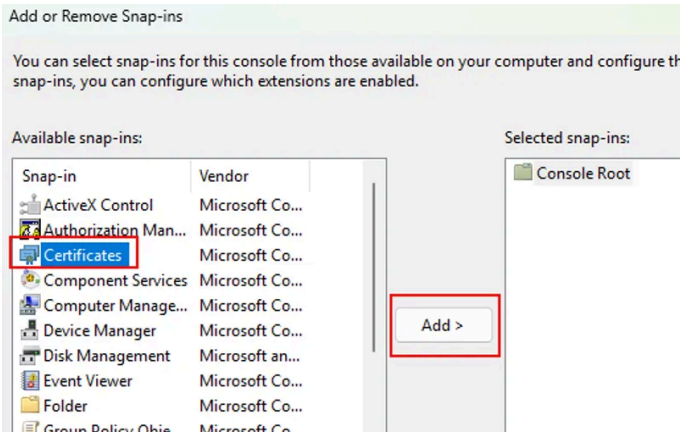
Once you enable this setting, your host device is secured and you can access the application securely with *https*.

Manage trusted root certificates for a local computer

To prevent security warnings in the browser, you need to add the certificate to the **Trusted Root Certification Authorities** store on your local computer so that the browser recognizes it as a trusted source.

To add the certificate to the **Trusted Root Certification Authorities**, follow these steps:

1. Click **Start** and search for **mmc**.
2. On the **File** menu, click **Add/Remove Snap-ins**.
3. Under **Available snap-ins**, select **Certificates** and click **Add**.



4. Select **Computer account**, and click **Next**.
5. Click **Finish**, and then click **OK**.
6. In the console tree, go to **Certificates (Local Computer)** > **Trusted Root Certification Authorities**.
7. Right-click **Certificates**, and go to **All Tasks** > **Import**.
8. Click **Next**, and upload the same certificate you used for SSL binding.

The local computer now trusts your certificate, and any connections or applications that rely on this certificate will recognize it as secure and valid.

If you want to configure both your Neurons for ITSM application database and configuration database to use SSL, refer to **Optional SSL Configuration**.

> SSL offloading is enabled in IIS Reverse Proxy, so we recommend deploying it only inside trusted network zones.

4.2.7. Installing Neurons for ITSM Reporting Feature

- About Microsoft SSRS
- Checking if Microsoft SSRS is Installed
- Installing Microsoft SSRS

About Microsoft SSRS

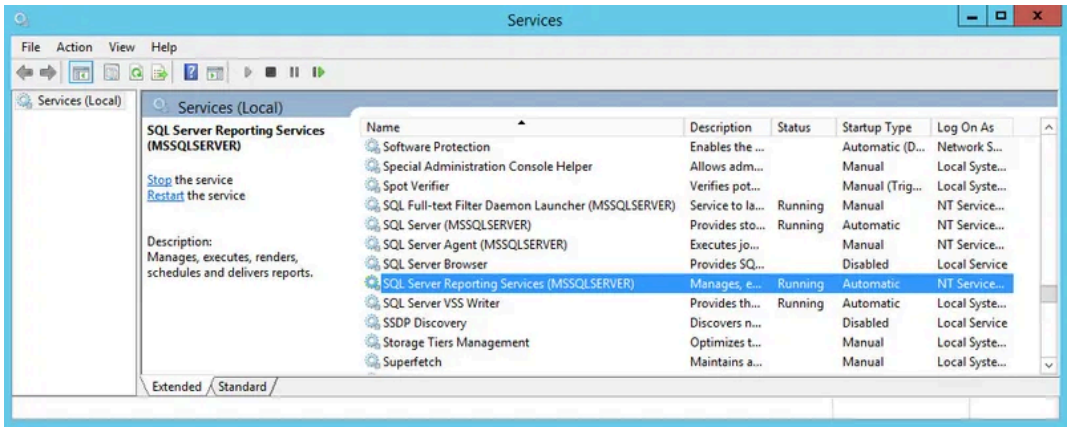
The Neurons for ITSM reporting feature requires that the 64-bit version of Microsoft SQL Server and Microsoft SQL Server Reporting Services (SSRS) are installed and running on the Neurons for ITSM database server. The reporting feature does not support the 32-bit version of Microsoft SQL Server and Microsoft SQL Server Reporting Services (SSRS).

Checking if Microsoft SSRS is Installed

To verify that Microsoft SSRS is installed on your system, do the following:

1. Open the Microsoft Windows **Services** dialog box and look for an entry called SQL Server Reporting Services.

   Windows Services Dialog Box

   The name in parentheses is the Microsoft SSRS instance name. See **Windows Services Dialog Box**. During the reporting feature configuration, you must enter this instance name.

   > The instance name cannot contain any special characters, such as \ , : ; ... and so on. The instance names can contain $ and _. See https://support.microsoft.com/en-us/help/2282743/fix-the-special-characters-are-displayed-incorrectly-in-a-textbox-cont.
   >
   > If the names of the current Microsoft SSRS instances have special characters, you must install an additional Microsoft SSRS instance with an instance name free of special characters.

2. Choose the appropriate action:
   - If Microsoft SSRS is already installed and the instance name (the name in parentheses) has no special characters (-, _, #, and so on), make sure that:
     - The value in the **Status** column is set to Running.
     - The value in the **Startup Type** column is set to either Automatic or Automatic (Delayed Start).

- If Microsoft SSRS is already installed but the instance name has special characters (-, _, #, and so on), use the Microsoft SQL Server Installation Center to add a new Microsoft SSRS instance with an instance name free of special characters.
- If Microsoft SSRS is not installed, use the Microsoft SQL Server Installation Center to add a Microsoft SSRS component to an existing Microsoft SQL installation.

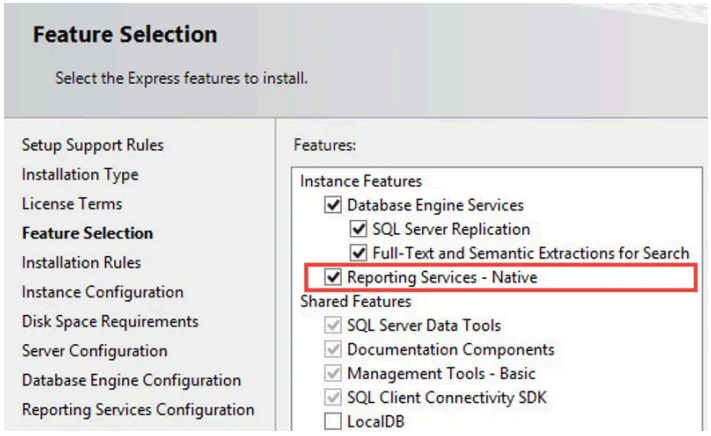
Installing Microsoft SSRS

Microsoft SQL Server Reporting Services (SSRS) is an optional installation feature of Microsoft SQL Server. It is a server-based, report-generating software system.

The Microsoft SSRS feature is disabled by default. This section describes how to enable and configure the Microsoft SSRS feature of Microsoft SQL Server.

1. During Microsoft SQL Server installation, in the **Feature Selection** page of the Microsoft SQL Server setup wizard, check **Reporting Services - Native**.

   Checking Reporting Services - Native

2. As you work your way through the Microsoft SQL Server setup wizard, verify that **Reporting Services - Native** is part of the server configuration.
3. When the installation is complete, go back to Windows Services and verify that Microsoft SQL Server Reporting Services is present. See **Checking Reporting Services - Native**.

With Microsoft SSRS present and running, the database server is ready for you to install and configure Neurons for ITSM.

4.2.8. Full Text Search

The full-text search feature of Microsoft SQL Server is a specialized indexing and querying service for character-based data in Microsoft SQL Server database tables.

By default, the application database is configured to index these incident fields:

- Owner
- ProfileFullName
- Resolution
- Subject
- Symptom

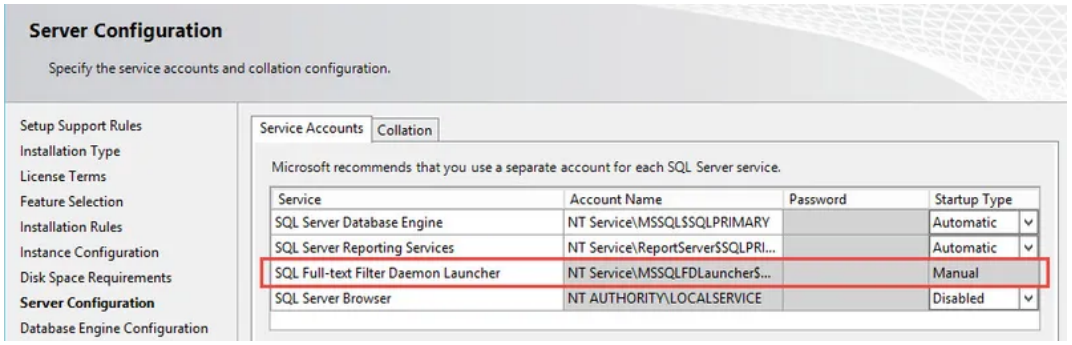
The full-text search feature is disabled by default.

Enabling the Full-Text Search Feature

This section describes how to enable and configure the full-text search feature of Microsoft SQL Server.

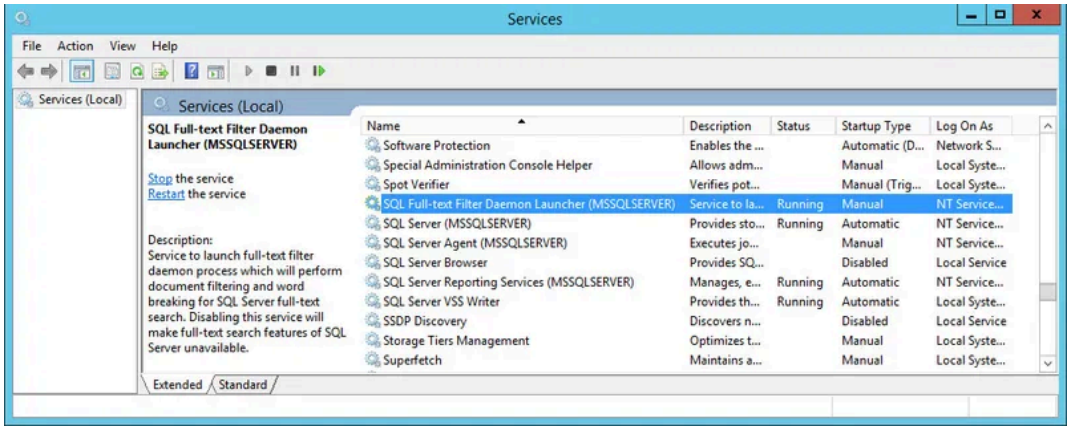
1. During Microsoft SQL Server installation, in the **Feature Selection** page of the Microsoft SQL Server setup wizard, check **Full-Text and Semantic Extractions for Search**.
2. In the **Server Configuration** page of the Microsoft SQL Server setup wizard, ensure that the Microsoft SQL Full-text Filter Daemon Launcher is configured with the local service account.

   Server Configuration



3. In the **Services** panel for the system, ensure that the status for **SQL Full-text Filter Daemon Launcher** says Running.

   Services Dialog Box



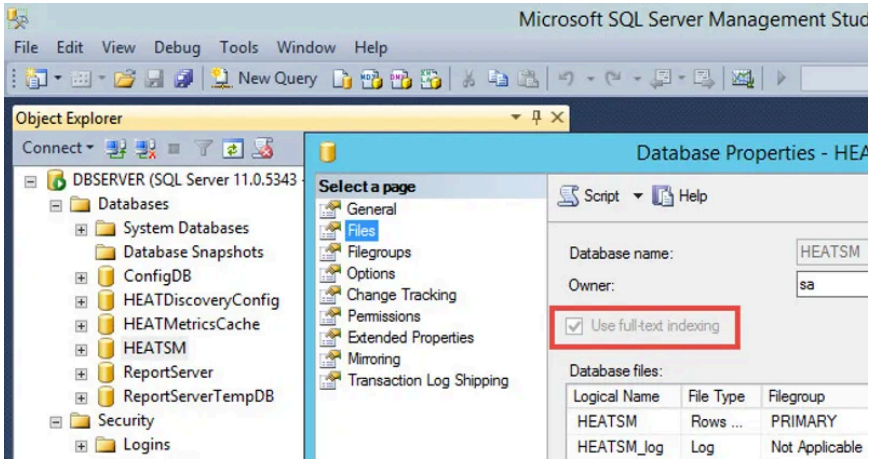
4. Verify that full-text search is configured in Microsoft SQL Server Management Studio by doing the following:

   a. Open the **Files** page from the **Database Properties** dialog box.

   b. Ensure that Use full-text indexing is checked.

   c. If it is not checked, verify that the full-text search services are running as described in **In the Services panel for the system, ensure that the status for SQL Full-text Filter Daemon Launcher says Running.**

   Checking the Use Full-Text Indexing Option



4.2.9. Verifying Server Roles and Features

You can optionally verify which server roles and features are installed by the Neurons for ITSM installer.

If you install Neurons for ITSM on Windows Server 2012 or Windows Server 2016, verify the Microsoft IIS Release 8.0 configuration. See Verifying Roles and Features for Windows Server 2012 and 2016.

Verifying Roles and Features for Windows Server 2012 and 2016

1. Go to the Windows apps menu and click **Server Manager**.
2. Under **configure this local server**, click **Add roles and features**.
3. On the **Roles** page, verify that these server roles are installed under **Web Server (IIS)** > **Web Server**:
    - Common HTTP features
        - Default Document
        - Directory Browsing
        - HTTP Errors
        - Static Content
    - Health and Diagnostics
        - HTTP Logging
    - Performance
        - Static Content Compression
        - Dynamic Content Compression
    - Security
        - Request Filtering
    - Application Development
        - .NET Extensibility
        - ASP.NET
        - ISAPI Extensions
        - ISAPI Filters
        - Under **Web Server (IIS)** > **Management Tools:**
        - IIS Management Console
        - Check the box of any listed server role that is not installed.
4. On the **Features** page, verify that these features are selected:
    - NET Framework 4.5
        - ASP.NET 4.5
        - WCF Services > HTTP Activation
    - Windows Process Activation Service
        - Process Model
        - Configuration APIs

- Check the box of any listed feature that is not installed.

5. Click **Next** and then click **Install**.

4.3. Installing the Neurons for ITSM System

This section describes how to install Neurons for ITSM for a new, complete system using the recommended deployment as described in **Neurons for ITSM Deployment Overview**.

- **Installing Neurons for ITSM in the Production Environment**
- **Installing Neurons for ITSM in the Staging and UAT Environments**
- **Installing the Reporting Feature**
- **Installing the Neurons for ITSM Operations Console**

Installing Neurons for ITSM in the Production Environment

To install Neurons for ITSM in the production environment, do the following:

- Install all Neurons for ITSM features except for the Operations Console. See **Installing Neurons for ITSM**.
- Use the System Configuration Wizard to configure your production environment. See **Configuring Neurons for ITSM** .

Installing Neurons for ITSM in the Staging and UAT Environments

- If you are hosting the staging and UAT environments on one server, follow the steps below.
- If you are hosting the staging and UAT environments on separate servers, follow the steps below once for the staging environment and once for the UAT environment.

To install Neurons for ITSM in the staging and UAT environments, do the following:

- Install all Neurons for ITSM features including the Operations Console.

  > If you are hosting the staging and UAT environments on separate servers, we recommend only installing the Operations Console on the staging server and not on both.

- Use the System Configuration Wizard to configure your staging and UAT environments. See **Configuring Neurons for ITSM** .

Installing the Reporting Feature

To install the reporting feature, do the following:

- Install the reporting feature on the Neurons for ITSM database server. See **Installing the Reporting Feature**.
- Use the System Configuration Wizard to configure the reporting feature. See **Configuring the Reporting Feature**.

Installing the Neurons for ITSM Operations Console

You use the Neurons for ITSM Operations Console to create the staging and UAT instances of the tenant. (The system automatically created the production instance of the tenant.)

If you have any problems with the installation, you can review the installation log file that resides with the other system temporary files in the system temporary folder at %tmp%.

4.3.1. Installing Neurons for ITSM

This chapter provides an overview of product deployment and associated installation steps.

Where to Install Neurons for ITSM

Before you begin installation, be clear on your deployment plan for Neurons for ITSM. See Neurons for ITSM**Neurons for ITSM Deployment Overview**.

- For the **Demonstration or Proof-of-Concept Deployment**, you install Neurons for ITSM onto the Neurons for ITSM server. With this deployment, you must install Microsoft SQL Server and Microsoft SQL Server Reporting Services (SSRS) on your server before you install Neurons for ITSM.
- For the **Minimum Production Deployment** , you install Neurons for ITSM onto the Neurons for ITSM server.
- For the **Enterprise Production Deployment**, you install Neurons for ITSM onto each of your Neurons for ITSM processing servers and web servers, as described below.

Installing Neurons for ITSM

Prerequisite:

- Without fail take a backup of the entire folder in all the machines where Neurons for ITSM is installed. This step is a mandatory preventive measure, do not miss it.
- Install the prerequisite installer using **Ivanti Service & Asset Manager Prerequisite.exe**.

  The **Ivanti Service & Asset Manager.exe** installs the essential software that is required for running the System Configuration Wizard (SCW) tool.

  For ITSM version 2025.4 and above installation, it places the additional App Domains payload at *C:|Program Files (x86)|HEAT Software|Prerequisite*.

  > The list of AppDomains for 2025.4 version is same as 2025.3 version.

- Install the **Additional Ap Domain** provided with the package. Follow instructions given in Additional App Domains Deployment. (Supports up to ITSM 2025.3 version)

  > Installing the Additional App Domains is mandatory.

Installation is basically the same for all hosts in your deployment, except for the features to be installed, which happens in **Your selection of features from the Setup Type dialog box depends on the role of the individual host in your deployment plan.**

> After the Additional App Domains powershell script is successfully run, the cache re-build is automatically executed. If the Additional App Domains powershell script is not run in same machine, then the Cache Rebuild should be executed manually. For instructions on how to executed Cache Rebuild manually, refer to **Neurons for ITSM Cache Rebuild**.

Do the following on each production server in your deployment:

1. Access the installation folder on the Software product CD or zip file and run **IvantiServiceManager.exe**. Right-click and select Run as Administrator to ensure proper installation.

The installer checks for the prerequisite software components. If any of those components is not installed, the system prompts you to install them now.

2. Click **Install** at the prompt. Installation of the prerequisite software can take several minutes. If you are prompted to restart the system, click **Yes**. The system displays the **Welcome** dialog box. The installer checks for space and other requirements before displaying the **Next** button.
3. Click **Next**. The system displays the **License Agreement** dialog box.
4. Choose **I accept the terms in the license agreement** and click **Next**. The system displays the **Destination Folder** dialog box.
5. Click **Next** to accept the default installation folder, or click **Change** and select a different folder. The system displays the **Setup Type** dialog box.
6. Your selection of features from the **Setup Type** dialog box depends on the role of the individual host in your deployment plan.
    - For the **Demonstration or Proof-of-Concept Deployment**, choose Complete, click Next, and install all components.
    - For the **Minimum Production Deployment** , choose Complete, click Next, and install all components.
    - For the **Enterprise Production Deployment**, your choice depends on the role of the host:
        - Production processing server: Choose Custom, click Next, and install all components except for the Neurons for ITSM Operations Console. You must install this server first before you install the staging, UAT, or web servers.
        - Staging or UAT processing servers: Choose Complete, click Next, and install all components.
        - Web servers: Choose Custom, click Next, and only install the Neurons for ITSM application server.

    For the **Enterprise Production Deployment**, note the following:

        - If your deployment includes different processing servers for production, staging, and UAT, you must install the License Manager in every one of those landscapes.
        - You can choose Complete and install all Neurons for ITSM features on all of your servers, but doing that takes up more disk space than necessary.

    When installing components, note the following:

        - To not install a component, click the down arrow next to the server icon next to the category name, highlight the component, right click, and select This feature will not be available.
        - If your deployment includes DSM, you do not need to install the DSM Integration service because that component has been incorporated into Neurons for ITSM since Release 2015.2.
7. Click Next. The system displays the **Ready to Install the Program** dialog box.
8. Click **Install**. The system begins installing Neurons for ITSM and displays a status dialog box, showing the installation progress of each module over the next few minutes.
9. Click OK if the system displays a dialog box saying that reboot is required after installation.
10. If the system displays a dialog box saying that some of the files that need to be updated are currently in use, choose Automatically close and attempt to restart applications and click OK.

    When the installation is completed, the system displays the System Configuration Wizard.
11. Go to **Configuring Neurons for ITSM** for instructions about using the System Configuration Wizard.

Installing Additional App Domains (Up to version 2025.3)

You will be prompted to install the App Domain after the SCW installation is complete. Proceed with the installation without skipping.

- Additional components zip file is provided along with the installer.
- All the components in the zip files needs a working setup of Ivanti Service & Asset Manager.
- Powershell script should be run with Administrator privilege.
- Ensure you have atleast 500 MB of free space before running the script.

List of App Domains:

1. Survey
    - SurveyUI
    - SurveyProxy
2. ActionService
3. TeamsBot

Steps to run the script:

1. Extract the zip file provided along with the Installer.

   The setup will have one powershell script named **install.ps1**.

2. Open a powershell window with Administrator privilege.
3. Run the following command from the extracted location:

   ```Bash
   Set-ExecutionPolicy Bypass -Scope Process -Force; .\install.ps1
   ```

4. Teamsbot is optional component, please type yes to install. If Installing Teamsbot there are two URLs you need to pass where components are installed:
    - AppServer URL
    - CentralConfig URL

   The value can be machine name (if above components are installed on a different server) or localhost (if the components are installed on same server).

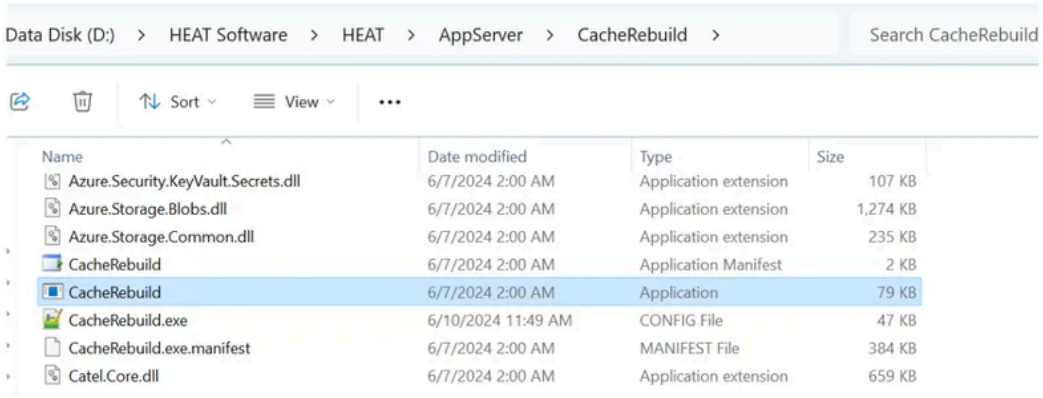
ActionService and Survey will be installed by default with the script as they are necessary components for the application to function properly.

Neurons for ITSM Cache Rebuild

Follow these steps to run the cache rebuild manually:

1. Open the command prompt as an Administrator.
2. Go to "HEAT Software\HEAT" folder on the machine where you have run the installer.

   For example, D:\HEAT Software\HEAT\AppServer\CacheRebuild.

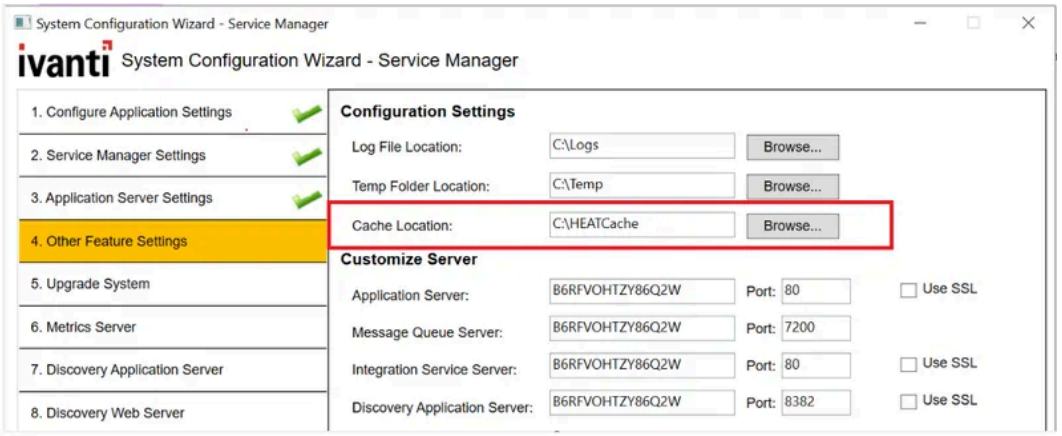


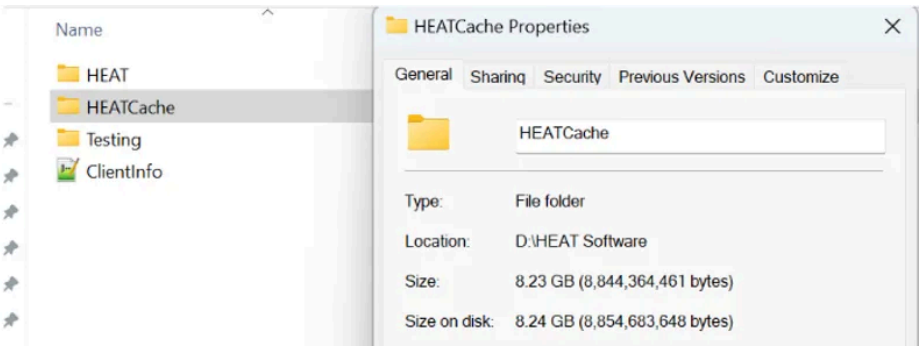
3. In the command prompt once the path is correct, run the **CacheRebuild.exe**.

   For example, D:\HEAT Software\HEAT\AppServer\CacheRebuild>CacheRebuild.exe.

> Do not close the command prompt window while the cache rebuild is running; it will close automatically after it's done.

The CacheRebuild folder will be saved in the path provided in the SCW.





4.3.2. Installing the Reporting Feature

Neurons for ITSM deployments can have multiple reporting tenant instances. To use the reporting feature in a multi-tenant environment, we highly recommend that you install a separate Microsoft SSRS instance for the reporting feature only. For example, a deployment may have one reporting server for the production instance of the tenant and another reporting server for the staging and UAT instances of the tenant. See **Enterprise Production Deployment**.

Where to Install the Reporting Feature

Where you install the reporting feature depends on which deployment you have chosen. See Neurons for ITSM **Neurons for ITSM Deployment Overview**.

- For the **Demonstration or Proof-of-Concept Deployment**, you install the reporting feature onto the Neurons for ITSM database server, after you have installed Neurons for ITSM.
- For the **Minimum Production Deployment** , you install the reporting feature on the Neurons for ITSM database server.
- For the **Enterprise Production Deployment**, you can install the reporting feature on the Neurons for ITSM database servers or on a dedicated reporting server. See **Reporting Services Deployment Options**.

Installing the Reporting Feature

Before you begin, ensure that you have met all of the prerequisites for installing the Neurons for ITSM reporting feature. See **Installing Neurons for ITSM Reporting Feature**. In particular, be sure that Microsoft SSRS is installed and running. See **Checking if Microsoft SSRS is Installed**.

> If you are using SQL 2016 or higher, ensure that you install Microsoft Report Builder manually.

Prerequisite:

- Install the prerequisite installer using **Ivanti Service & Asset Manager Prerequisite.exe**.

  The **Ivanti Service & Asset Manager Prerequisite.exe** installs the essential software that is required for running the system Configuration Wizard tool.

To install the reporting feature, do the following:

1. Access the installation folder on the Software product CD or zip file and run **ReportingServices.exe**. Right-click and select Run as Administrator to ensure proper installation.

   The installer checks for the prerequisite software components. If any of those components is not installed, the system prompts you to install them now.

2. Select **Install** at the prompt. Installation of the prerequisite software can take several minutes. If you are prompted to restart the system, click **Yes**. The system displays the **Welcome** dialog box. The installer checks for space and other requirements before displaying the **Next** button.
3. Click **Next**. The system displays the **License Agreement** dialog box.
4. Select **I accept the terms in the license agreement** and click **Next**. The system displays the **Destination Folder** dialog box.
5. Click **Next** to accept the default installation folder, or click **Change...** and select a different folder.
6. Click Next. The system displays the **Ready to Install the Program** dialog box.

7. Click **Install**. The system begins installing the Neurons for ITSM reporting feature and displays a status dialog box showing the installation progress over the next few minutes.

   When the components are installed, the system automatically launches the System Configuration Wizard.

8. Go to **Configuring the Reporting Feature** for instructions about using the System Configuration Wizard.

4.3.3. Installing Discovery on a Dedicated Server

In small deployments, you install Discovery with the other Neurons for ITSM components on the Neurons for ITSM production servers. For an **Enterprise Production Deployment**, you can either install Discovery on the production servers or, to reduce the load on the production servers, you can configure a dedicated Discovery server. If you install Discovery on your production servers and then experience a significant load increase, we recommend configuring a dedicated Discovery server.

This topic describes how to install Discovery on a dedicated server.

1. Install Neurons for ITSM on your production servers. See **Installing Neurons for ITSM**.
2. On the server dedicated to Discovery, access the installation folder on the Software product CD or zip file and run IvantiServiceManager.exe. Right-click and select Run as Administrator to ensure proper installation.

   The installer checks for the prerequisite software components. If any of those components is not installed, the system prompts you to install them now.
3. Click **Install** at the prompt. Installation of the prerequisite software can take several minutes. If you are prompted to restart the system, click **Yes**. The system displays the **Welcome** dialog box. The installer checks for space and other requirements before displaying the **Next** button.
4. Click **Next**. The system displays the **License Agreement** dialog box.
5. Select **I accept the terms in the license agreement** and click **Next**. The system displays the **Destination Folder** dialog box.
6. Click **Next** to accept the default installation folder or click **Change...** and select a different folder. The system displays the **Setup Type** dialog box.
7. Select the following components for the **Custom Setup** dialog box. Under Discovery, choose:
   - Web Application Components
   - Discovery Web Server
8. Click Next. The system displays the **Ready to Install the Program** dialog box.
9. Click **Install**. The system begins installing Discovery and displays a status dialog box showing the installation progress of each component over the next few minutes.

   When the components are installed, the system automatically displays the System Configuration Wizard.
10. Go to **Configuring the Discovery Application Server** for instructions about using the System Configuration Wizard.

4.3.4. Installing the Knowledge Uploader (Knowledge Import Tool)

- About the Knowledge Uploader (Knowledge Import Tool)
- Installing the Knowledge Uploader (Knowledge Import Tool)
- Initially Configuring and Logging Into the NXT Knowledge Import Tool
- Using the NXT Knowledge Import Tool
- Initially Configuring and Logging Into the Knowledge Import Tool
- Using the Knowledge Import Tool

About the Knowledge Uploader (Knowledge Import Tool)

You can import knowledge articles from HEAT Classic, ITSM, or a local or networked file system into Neurons for ITSM. Neurons for ITSM currently has two knowledge import tools: the NXT Knowledge Server Tool (for NXT formats such as ITSM or HEAT Classic) and the Knowledge Server Tool (to use with file systems). These tools are collectively called the Knowledge Uploader and are also called the Knowledge Import Tool.

For information about using these tools after you have installed and configured them, see **Using the Knowledge Uploader** in the Neurons for ITSM online help.

Installing the Knowledge Uploader (Knowledge Import Tool)

Most installations of Neurons for ITSM automatically include the Knowledge Uploader (also called the Knowledge Import Tool).

The following procedure is necessary only if you did not install the Knowledge Uploader when you installed Neurons for ITSM.

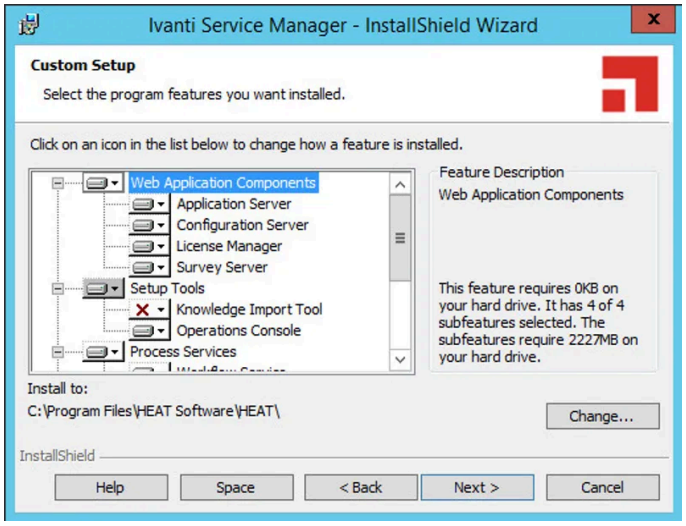
On the server or servers that you are going to use as the production environment, do the following:

1. Access the installation folder on the Software product CD or zip file and run IvantiServiceManager.exe. Right-click and select Run as Administrator to ensure proper installation.

   The installer checks for the prerequisite software components. If any of those components is not installed, the system prompts you to install them now.

2. Click **Install** at the prompt. Installation of the prerequisite software can take several minutes. If you are prompted to restart the system, click **Yes**. The system displays the **Welcome** dialog box. The installer checks for space and other requirements before displaying the **Next** button.
3. Click **Next**. The system displays the **License Agreement** dialog box.
4. Choose **I accept the terms in the license agreement** and click **Next**. The system displays the **Destination Folder** dialog box.
5. Click **Next** to accept the default installation folder or click **Change...** and select a different folder. The system displays the **Setup Type** dialog box.
6. Select Custom in the **Setup Type** dialog box and click the Knowledge Import Tool under Setup Tools. See Custom Setup and the Knowledge Import Tool.

   Custom Setup and the Knowledge Import Tool



7. Click Next. The system displays the **Ready to Install the Program** dialog box.
8. Click **Install**. The system begins installing Neurons for ITSM and displays a status dialog box showing the installation progress of each component over the next few minutes.

   When the installation is finished, the system displays the **InstallShield Wizard Completed** dialog box.
9. Click Finish. When the file installation is finished, the system displays the System Configuration Wizard.
10. Close the System Configuration Wizard, as there is no specific configuration for the Knowledge Uploader in the System Configuration Wizard.

The system now has two new applications on it. They are called Knowledge Import Tool and Knowledge Import Tool (NXT).

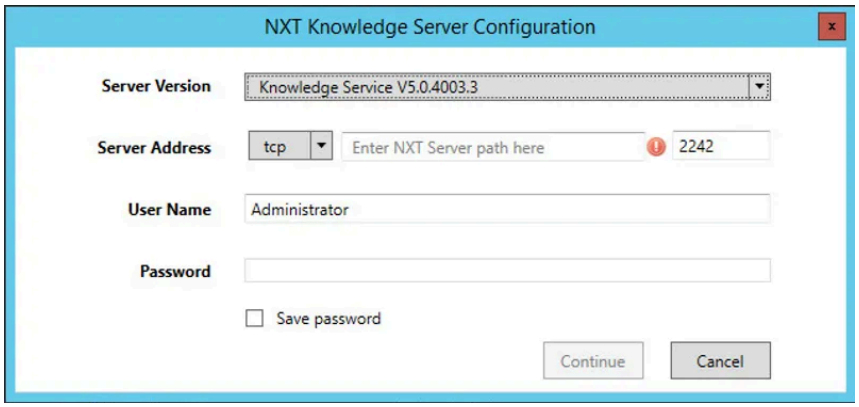
Initially Configuring and Logging Into the NXT Knowledge Import Tool

> There are two knowledge tools. This topic is about the NXT Knowledge Import Tool and NOT the Knowledge Import Tool.

After you have installed Neurons for ITSM, including the Knowledge Uploader (also called the Knowledge Import Tool), do the following to initially configure and log into the NXT Knowledge Import Tool:

1. On the system where you installed Neurons for ITSM, open the application called **Knowledge Import Tool NXT**. The system displays the **NXT Knowledge Server Configuration** dialog box. See NXT Knowledge Server Configuration Dialog Box.

   NXT Knowledge Server Configuration Dialog Box



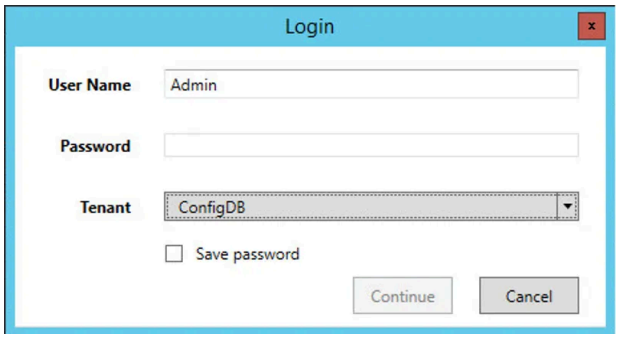
2. Enter information into the fields.

Parameter	Description
Server Version	The version of the NXT server that you are taking the knowledge articles from. To determine the server version, open the program and find the server information. You can choose from any of the following: • Knowledge Service V5.0.4003.3 • Knowledge Service Plus V9.6.1.13 • Knowledge Service Plus V9.6.1.14
Server Address	Enter the following: • The protocol type (uses TCP by default) • The server path of the NXT server that you are taking the knowledge articles from • The port (uses 2242 by default)
User Name	The user name of a person who can access the NXT server that you are taking the knowledge articles from.
Password	The password associated with the user name.
Save password	Saves the password.

> To change these settings after the initial configuration, from the menu bar select Configuration > Configure NXT server.

3. Click Continue.
4. The system displays the **Login** dialog box. See **Knowledge Server Tool Login**.

Knowledge Server Tool Login



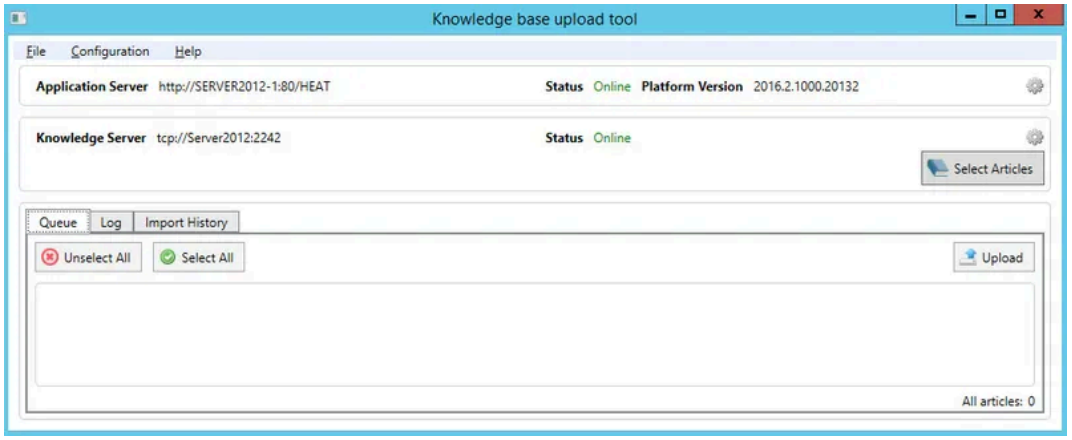
5. Enter information into the fields.

Parameter	Description
User Name	The user name of a person who can access Neurons for ITSM.
Password	The password associated with the user.
Tenant	The tenant to log into. You can select either ConfigDB to log into the configuration database or IvantiSM to log into Neurons for ITSM.
Save password	Saves the password.

> To change these settings after the initial configuration, from the menu bar select Configuration > Configure application server.

6. Click Continue. The system displays the **Select role** dialog box.
7. Select the role to log in as and click Continue. The system displays the **Knowledge base upload tool** page.

Knowledge Base Upload Tool



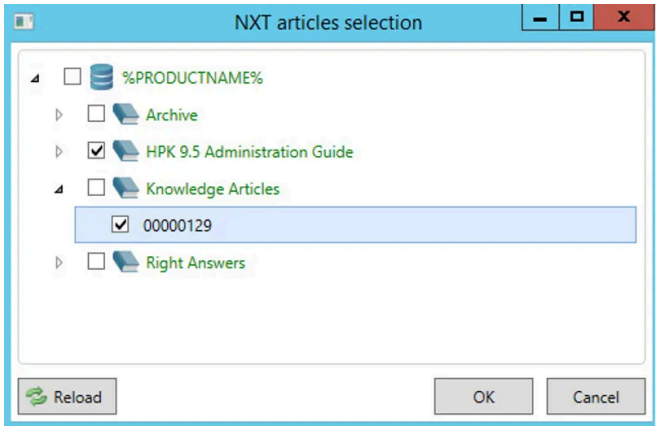
Using the NXT Knowledge Import Tool

> There are two knowledge tools. This topic is about the NXT Knowledge Import Tool and NOT the Knowledge Import Tool.

Follow these steps to import knowledge into Neurons for ITSM.

1. Click Select Articles to select the knowledge articles to import. The system displays the **NXT articles selection** dialog box.
2. Expand the list to see where knowledge articles are located, and then highlight one or more articles.

Select the Knowledge Articles to Import



3. Click OK. The system lists the selected articles on the bottom part of the page.
4. Select the articles to upload and click Upload to launch the **Article Creation** wizard and start the upload process.

   The **Article Creation wizard** page contains two main areas: a **Selected Documents** area on the left containing the list of files to import, and a **Document Properties** area on the right that contains two tabs. One tab is called **Default Values** and the other is called **Selected Documents**.

5. To remove a document from the **Selected Documents** section, check it and click **Remove selected**. By default, the knowledge uploader imports all of the documents in the **Selected Documents**, even if you do not select them.
6. Optionally, use the **Document Properties** section to specify the properties that the knowledge uploader displays together with the files in Neurons for ITSM. You can specify properties for all files (using the **Default values** tab) or for individual files (using the **Selected Document** tab after highlighting the file in the **Selected Documents** section).

   You can specify these properties:

   - **Article Type**: The article type. Select from Document, Error Message, Issue Resolution, Patch, Q&A, or Reference.
   - **Article Status**: The status of the article. Select from Approved, Draft, Expired, Failed Review, Pending Review, Published, or Submitted.
   - **Target collection**: The target. Select from Company Policy & Procedure, Customer Knowledge, Internal Knowledge, IT Knowledge, or Service Desk.
   - **Category**: The category for this knowledge article.
   - **Subject**: The subject of the article.
   - **Resolution**: The resolution details if the knowledge article contains this information.
   - **Description**: A description for the knowledge article.
7. Click **Import**. The system starts importing the selected knowledge articles.

   If you select a file that has already been imported into the knowledge base, the system displays the **Collision warning** dialog box.

8. Depending on what you want to do with the file that has already been imported, click one of the following:
   - **Apply for all**: Applies your choice to all previously imported files that are found during the import process.
   - **Update**: Overwrites the file that already exists in the Knowledge Base with the file that is being imported.
   - **Create New**: Does not overwrite the file being imported with the file that already exists in the Knowledge Base. After importing, there are two knowledge articles with the same name in the Knowledge Base.

- **Cancel**: Cancels the import process.

The system displays a dialog box with the status of the import.

9. Click **OK**.
10. Optionally, do the following:
    - Review the import logs by clicking the **Log** tab. You can clear or save the log.
    - Review the import history by clicking the **Import History** tab. Click a number in the **Imported** column to display more information about an import session.

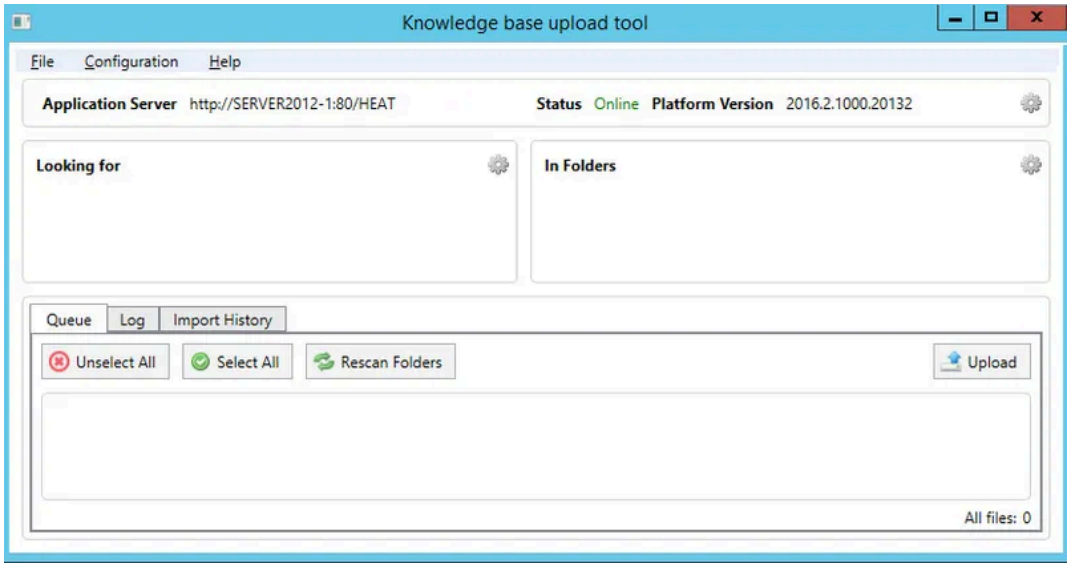
Initially Configuring and Logging Into the Knowledge Import Tool

> There are two knowledge tools. This topic is about the Knowledge Import Tool and NOT the NXT Knowledge Import Tool.

After you have installed Neurons for ITSM, including the Knowledge Uploader, do the following to initially configure and log into the Knowledge Import Tool:

1. On the system where you installed Neurons for ITSM, open the application called **Knowledge Import Tool**. The system displays the **Select role** dialog box.
2. Select the role to log in as and click Continue. The system displays the **Knowledge base upload tool** page.

Knowledge Base Upload Tool



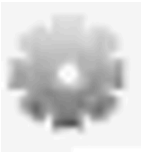
Using the Knowledge Import Tool

> There are two knowledge tools. This topic is about the Knowledge Import Tool and NOT the NXT Knowledge Import Tool.

Follow these steps to import knowledge into Neurons for ITSM.

1. Specify the article file type to search for, by doing the following:

a. Click the **Settings** icon in the **Looking for** section. The system displays the **File search filters** dialog box.



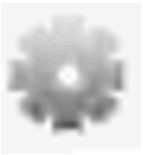
b. In the **Add New** drop-down list, select one or more file types:

- **Microsoft office documents**: Files that have .doc, .docx, .xls, and .xlsx extensions.
- **PDF files**: Files that have a .pdf extension.
- **Images**: Files that have .bmp, .jpg, and .gif extensions.

c. Optionally, you can add more file types by clicking **Custom**. The system adds a line item to the filter list called Custom. Click the line item to open it for editing. You can change the filter name and specify one or more extensions. Click **OK**. The system adds the file types that you specified to the **Looking for** section.

2. Specify which local or network folders to search for knowledge articles and then perform the search:

a. Click the **Settings** icon in the **In Folders** section. The system displays the **Import folders** dialog box.



b. Click **Add New**. The system displays the **Browse For Folder navigation** dialog box.

c. Navigate to a folder and highlight it. (To optionally add a new subfolder to the folder, click **Make New Folder**.)

d. Click **OK**. The system adds the folder to the list of folders to search. If you created a new folder, the system adds it to the file system as an empty folder.

e. Optionally, check **Recursive** for a folder to specify that the system should search the subfolders within the folder.

f. Optionally, check **Monitor for changes** to configure the knowledge uploader to monitor all of the folders on the list for changes to the file types that you specified in step 1b.

g. Repeat steps b-f to continue to add folders to the list as necessary.

h. Click **OK**.

The knowledge uploader performs the search and displays all of the files that meet the search criteria.

3. Select the files to upload:

a. If you think the contents of the scanned folders might have changed, click **Rescan Folders** to regenerate the file list.

b. In the file list, check the files to upload, or click **Select All** to upload all of the files.

c. Click **Upload** to launch the **Article Creation** wizard and start the upload process.

The **Article Creation wizard** page contains two main areas: a **Selected Documents** area on the left containing the list of files to import, and a **Document Properties** area on the right that contains two tabs. One tab is called **Default Values** and the other is called **Selected Documents**.

4. To remove a document from the **Selected Documents** section, check it and click **Remove selected**. By default, the knowledge uploader imports all of the documents in the **Selected Documents**, even if you do not select them.
5. Optionally, use the **Document Properties** section to specify the properties that the knowledge uploader displays together with the files in Neurons for ITSM. You can specify properties for all files (using the **Default values** tab) or for individual files (using the **Selected Document** tab after highlighting the file in the **Selected Documents** section).

   You can specify these properties:

   - **Article Type**: The article type. Select from Document, Error Message, Issue Resolution, Patch, Q&A, or Reference.
   - **Article Status**: The status of the article. Select from Approved, Draft, Expired, Failed Review, Pending Review, Published, or Submitted.
   - **Target collection**: The target. Select from Company Policy & Procedure, Customer Knowledge, Internal Knowledge, IT Knowledge, or Service Desk.
   - **Category**: The category for this knowledge article.
   - **Use file name as Subject**: Check to use the article file name to fill in the **Subject** field.
   - **Subject**: Only use this field if you did not check **Use file name as Subject**. Enter a subject for the article.
   - **Resolution**: Enter the resolution if the knowledge article contains one.
   - **Description**: Enter a description for the knowledge article.
6. Click **Import**. The import process begins.

   If you select a file that has already been imported into the knowledge base, the system displays the **Collision warning** dialog box.
7. Depending on what you want to do with the file that has already been imported, click one of the following:
   - **Apply for all**: Applies your choice to all previously imported files that are found during the import process.
   - **Update**: Overwrites the file that already exists in the knowledge base with the file that is being imported.
   - **Create New**: Does not overwrite the file being imported with the file that already exists in the knowledge base. After importing, there are two knowledge articles with the same name in the knowledge base.
   - **Cancel**: Cancels the import process.

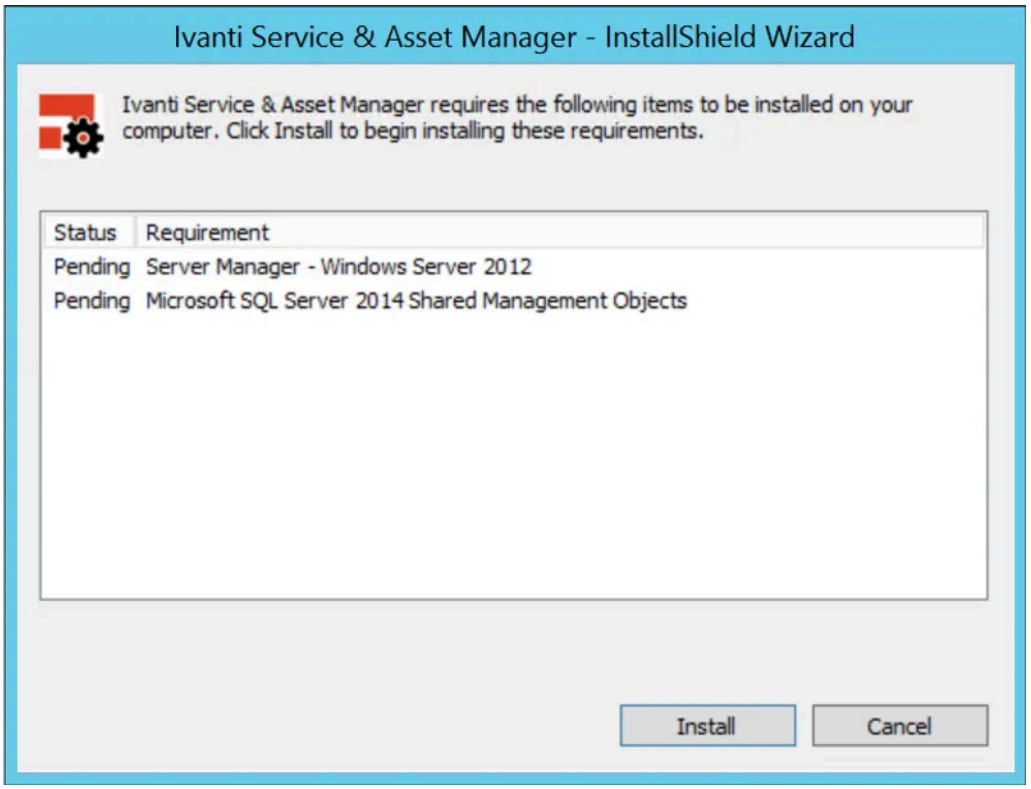
   The system displays an Import dialog box with the status of the import.
8. Click **OK**.
9. Optionally, do the following:
   - Review the import logs by clicking the **Log** tab. You can clear or save the log.
   - Review the import history by clicking the **Import History** tab. Click a number in the **Imported** column to display more information about an import session.

4.4. Initially Configuring the System

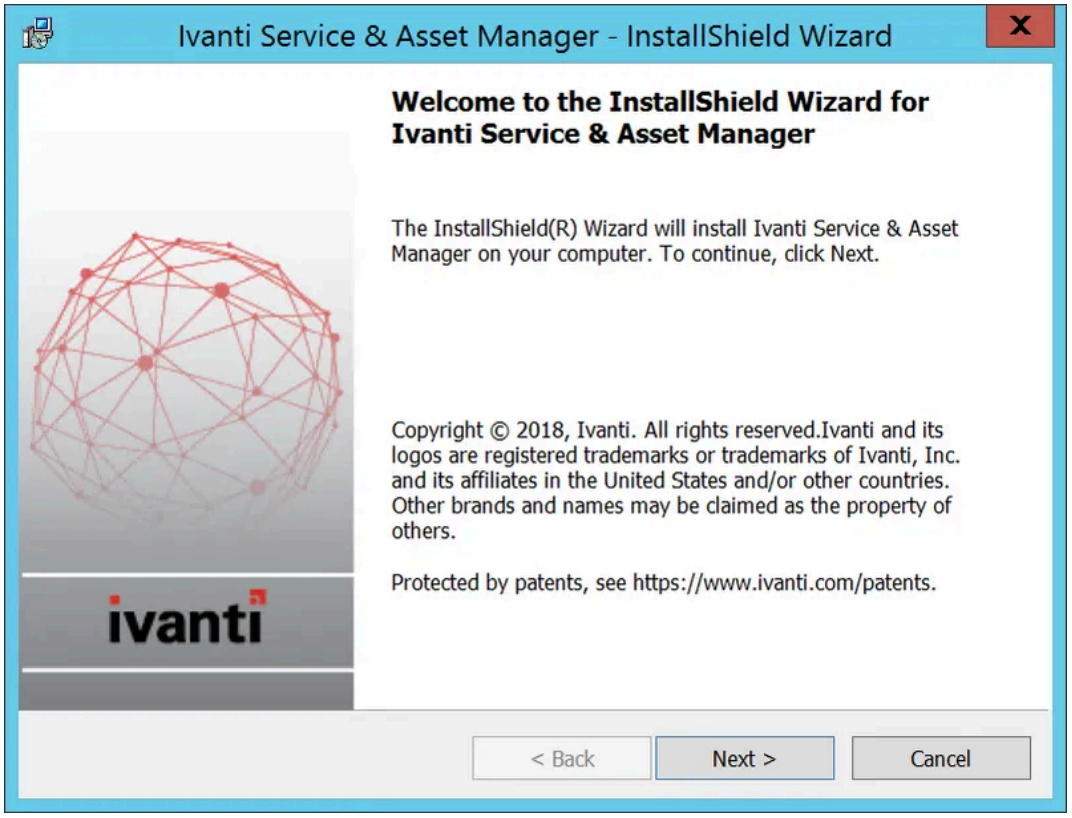
As part of the Neurons for ITSM installation, you must configure the system using the System Configuration Wizard.

The wizard walks you through the first-time set up and configuration as shown in the screen shots below.

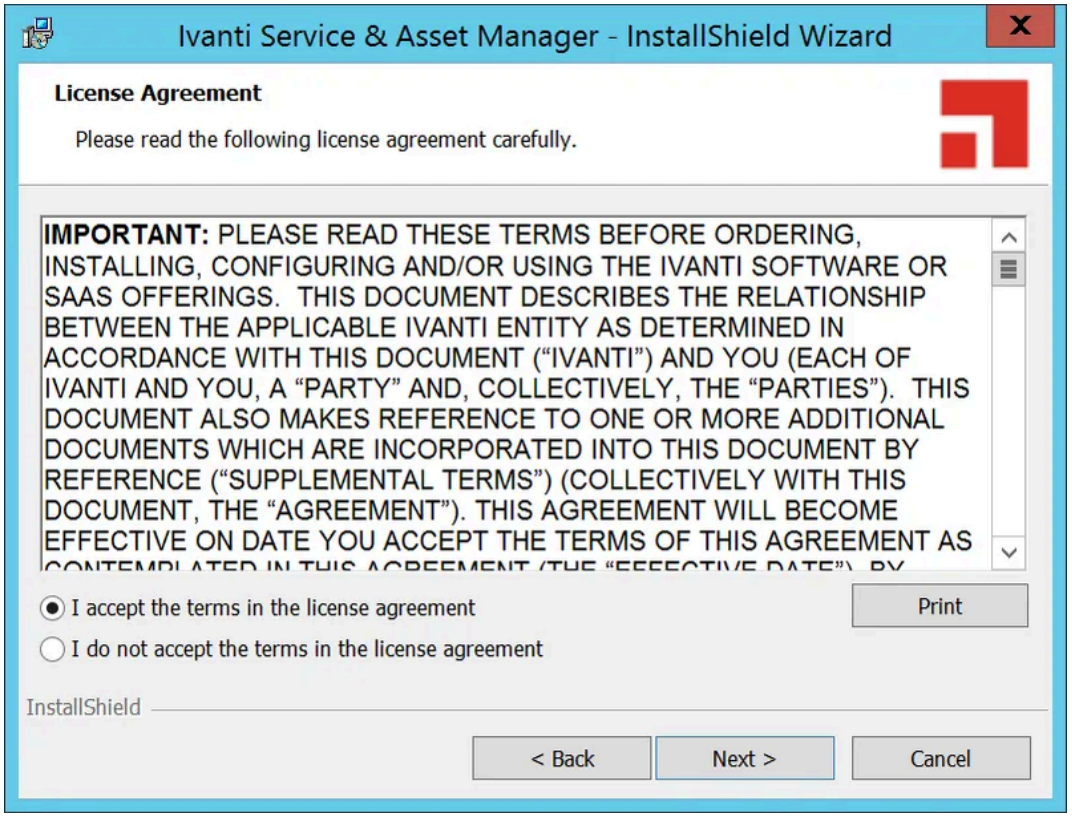
1. Click **Install** to begin the installation.



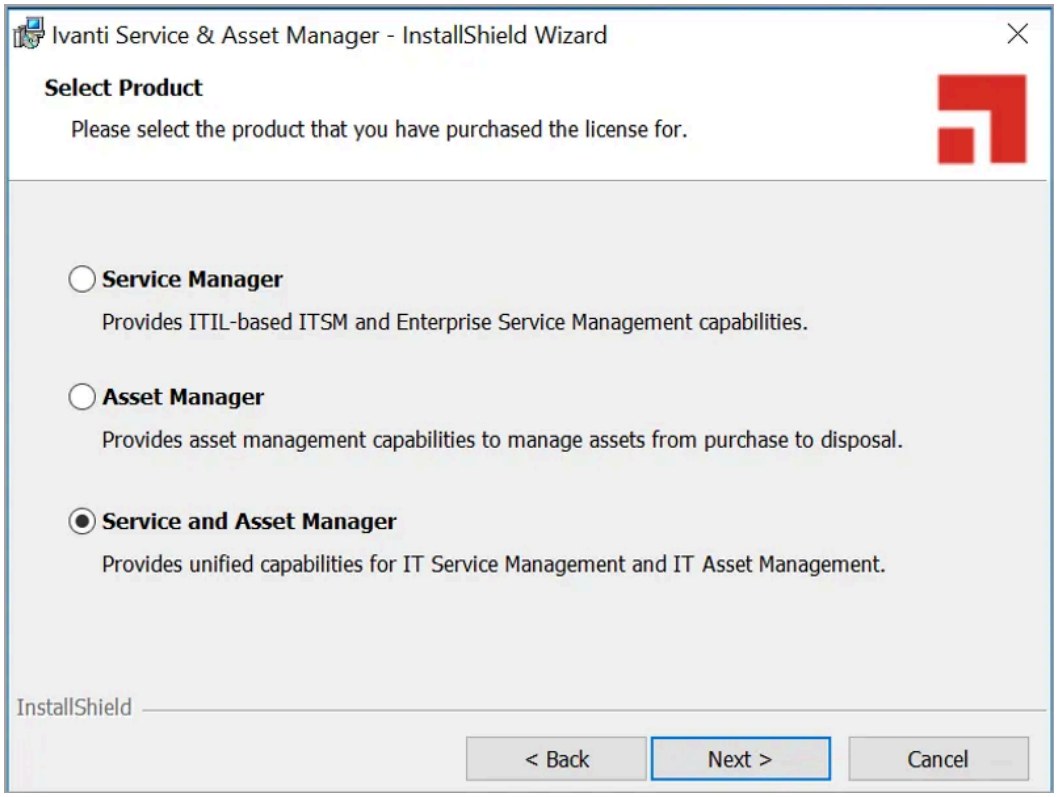
2. Click **Next**.



2. Click **Next** to continue the installation process.

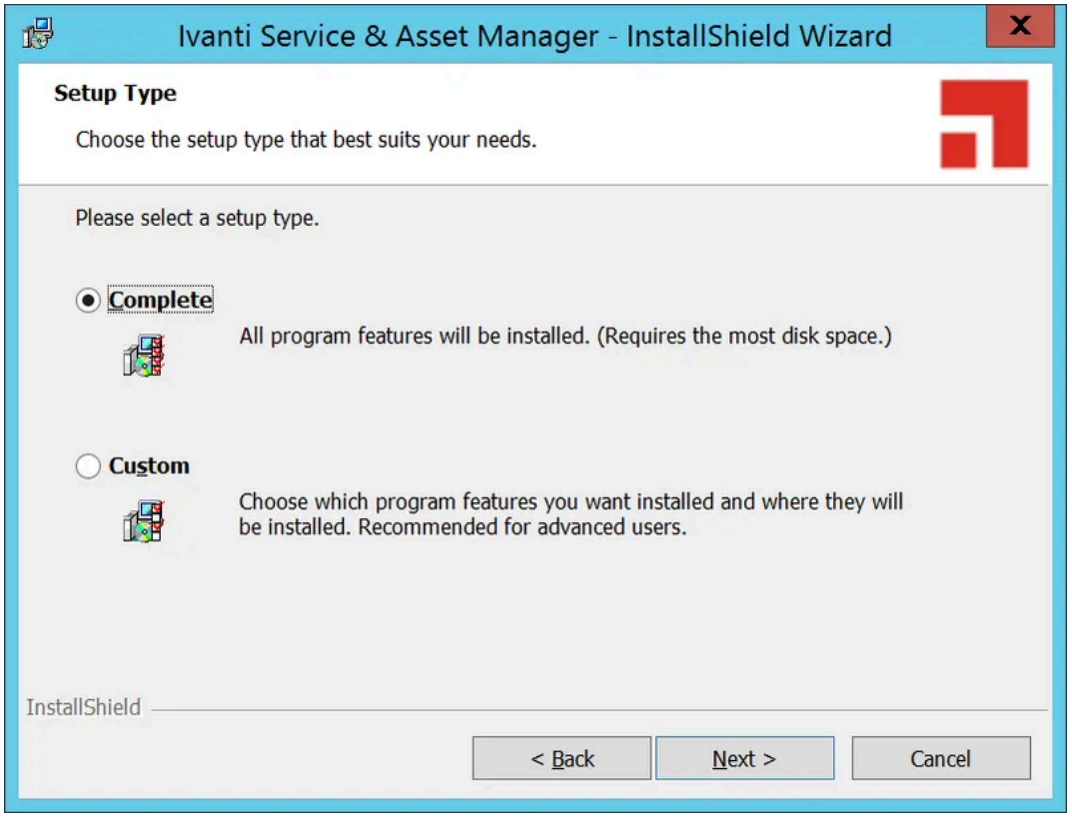


2. Select one of the options, **Service Manager**, **Asset Manager**, or **Service and Asset Manager**.

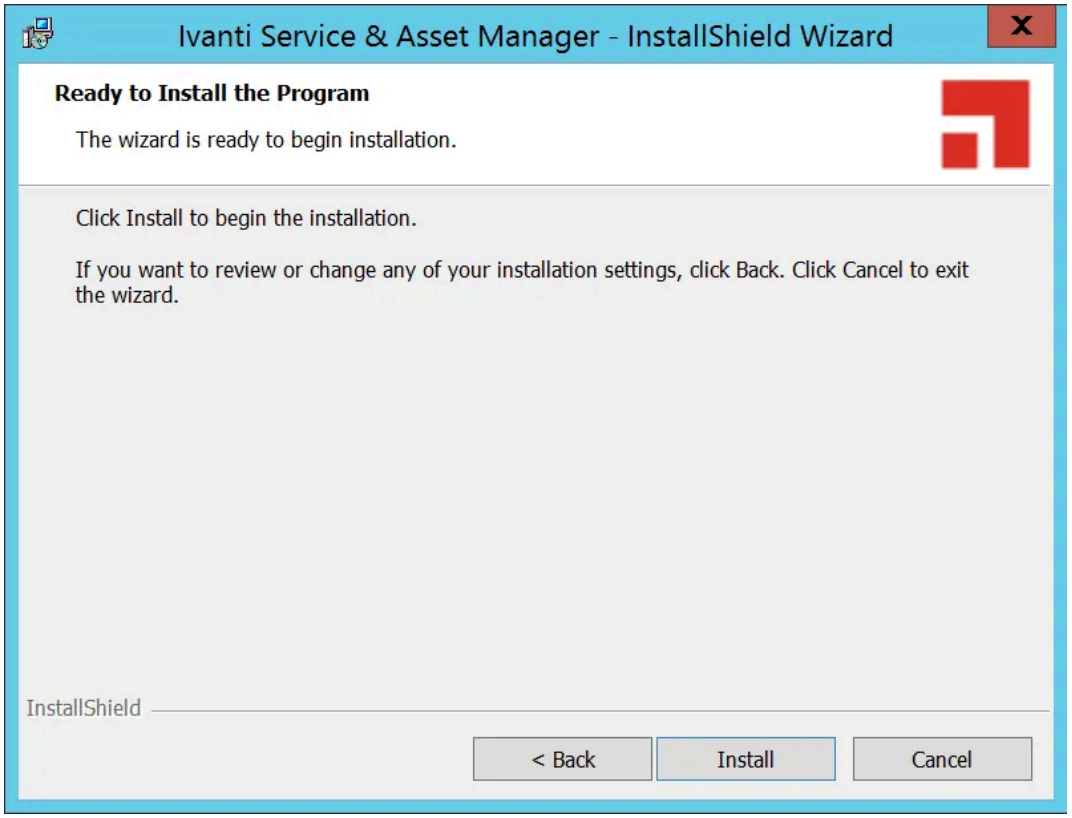


For more information on Asset Manager, see **Ivanti Asset Manager**.

5. Select the **Custom** or **Complete** options.



2. Click **Next**.



The tabs that you see in the wizard depend on which components that you installed. You may see all of the tabs or you may just see a subset of them.

See the following sections:

- Using the System Configuration Wizard
- Configuring Neurons for ITSM
- Configuring the Reporting Feature
- Configuring Discovery
- Configuring the Deployment on the Neurons for ITSM Operations Console
- Optional SSL Configuration
- Optional LDAP Configuration
- About Configuring with ADFS
- About Configuring Throttling Settings

4.4.1. Using the System Configuration Wizard

The System Configuration Wizard provides an interface in which you can perform various configuration tasks for the initial setup of Neurons for ITSM.

- Starting the System Configuration Wizard Manually
- Navigating the System Configuration Wizard
- Actions to Perform in Each Page

Starting the System Configuration Wizard Manually

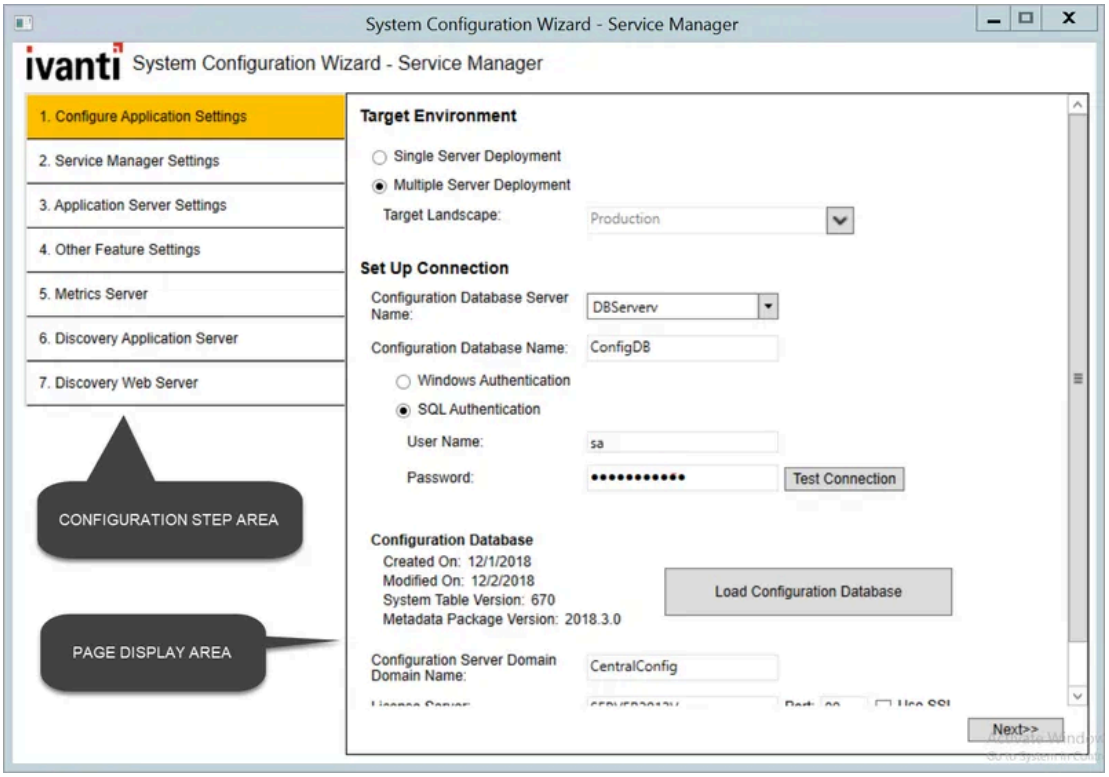
The System Configuration Wizard starts automatically after you finish installing Neurons for ITSM.

You can also start the System Configuration Wizard manually after your system is configured. See Changing Feature Settings by Running the System Configuration Wizard.

Navigating the System Configuration Wizard

The System Configuration Wizard contains the areas shown in System Configuration Wizard Areas.

System Configuration Wizard Areas



- In the configuration step area, the page that you are currently working on is highlighted in yellow.
- After you complete a page and click Next, the system places a green check mark next to the completed page name to show that it has been completed.

> The configuration step area is not a navigation pane. You cannot go to a page by clicking it in the configuration step area. Instead, click the **Next** and **Previous** buttons to move through the pages sequentially.

- The page display area shows the wizard page itself, including the fields and controls that you set the various parameters.

> - Chat Configuration installs Redis version 6.2 re-distributable which requires internet access while configuring the System Configuration Wizard.
> - The default behavior of the System Configuration Wizard with respect to **SSL** check boxes have changed from **non-SSL** to **SSL**. For non-SSL based installation these check boxes need to be cleared.
> - As part of the security fix, the System Configuration Wizard requires host name with domain (Fully Qualified Domain Name). For more information, refer to the **Configuring the Application Server Settings section** in the Installation and Deployment guide.

Actions to Perform in Each Page

The System Configuration Wizard page numbers can change depending on the features you choose to install.

Unless otherwise noted, the settings that you specify in each wizard page are stored in the Configuration Database.

Configuration Application Settings page:

- Specify the name of the configuration database and the authentication method that is used by all supported web applications when connecting to the configuration database.
- Create or recreate the configuration database.
- If you use Windows Integrated Security for the configuration database, specify the Windows domain credentials for the service account.

See **Configuring the Configuration Database**.

Service Manager Settings page:

- Specify the name and authentication method for the server where the Neurons for ITSM application database resides.
- Load the demo database on a server other than the server where the Neurons for ITSM application database resides. This option allows you to review the demo database while the Neurons for ITSM application database continues to run in its current configuration.
- Specify the application name, attachment location, client authorization key, and the type of name to use when accessing the application.

See **Configuring the Neurons for ITSM Application**.

Application Server Settings page:

- Specify the location of the configuration server and the host name.
- Specify if you will use this server for surveys or the reporting feature.
- If you use Windows Integrated Security for the Neurons for ITSM application server, specify the Windows domain credentials for the service account.

See **Configuring the Application Server Settings**.

Other Feature Settings page:

- Specify the log file and temporary file locations on the configuration server.
- Specify different server hosts and whether to use SSL for the Neurons for ITSM application server.
- Specify the inbound web server information.

See **Configuring Other Feature Settings**.

Metrics Server page:

- Specify the locations for the log file and the Neurons for ITSM configuration server.
- Map applications to the metrics server.

See **Configuring the Metrics Server**.

Microsoft SSRS Configuration page:

- Appears when you install the reporting feature.
- Specify information about the Microsoft SQL Server database for the reporting feature.

See **Configuring the Reporting Feature**.

Reporting Service Configuration page:

- Appears when you install the reporting feature.
- Specify information about the Neurons for ITSM report database for the reporting feature.

See **Configuring the Reporting Feature**.

Discovery Application Server page:

- Specify information about the configuration of the Discovery application server.

See **Configuring the Discovery Application Server**.

Discovery Web Server page:

- Specify information about the configuration of the Discovery web server.
- Upgrade the configuration database and the Neurons for ITSM application database.

See **Configuring the Discovery Web Server**.

Upgrade System page:

- Appears when you upgrade to a newer release of Neurons for ITSM.
- Upgrade the configuration database and the Neurons for ITSM application database.

See **Upgrading Neurons for ITSM from an Earlier Release**.

4.4.2. Configuring Neurons for ITSM

The following sections contain step-by-step instructions for the procedures that you can perform through the System Configuration Wizard.

- About Entering Server Location Names in the System Configuration Wizard
- Configuring the Configuration Database
- Configuring the Neurons for ITSM Application
- Configuring the Application Server Settings
- Configuring Other Feature Settings
- Configuring the Metrics Server
- Configuring the Discovery Application Server
- Configuring the Discovery Web Server
- Finishing the System Configuration

About Entering Server Location Names in the System Configuration Wizard

For all of the fields in the System Configuration Wizard that ask for a server location, unless noted otherwise, you can enter one of three things:

- An IP address
- A machine name
- A fully-qualified domain name

We recommend that you use the same naming format in all server location fields, if possible.

- If you are using SSL, you must enter a fully-qualified domain name.
- If you have a load-balanced system, we recommend using a fully-qualified domain name.

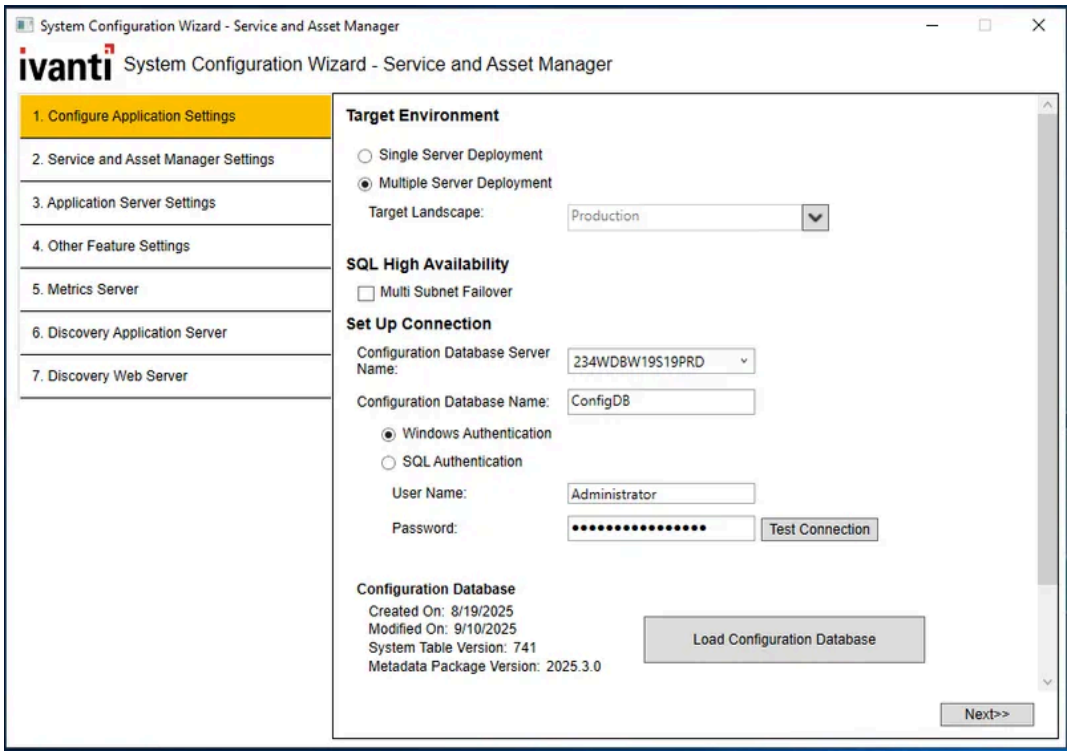
4.4.2.1. Configuring the Configuration Database

After you have installed Neurons for ITSM, the System Configuration Wizard starts automatically and displays the **Configuration Application** page.

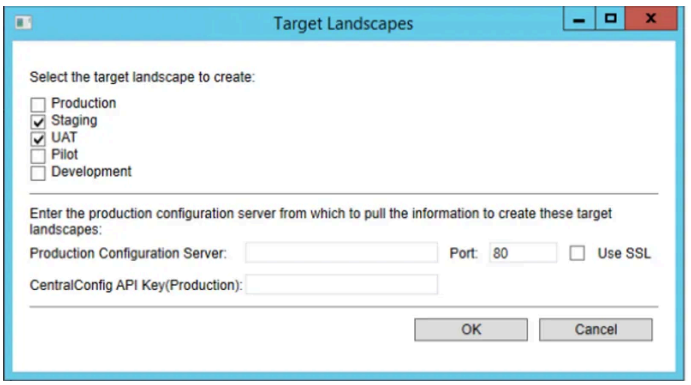
On this page, you will do the following:

- Specify the name of the configuration database and the authentication method that is used by all supported web applications when connecting to the configuration database.
- Create or recreate the configuration database.
- If you use Windows Integrated Security for the configuration database, specify the Windows domain credentials for the service account.

Configuration Application Page



1. Enter values into the fields:

Parameter	Description
Target Environment	
Deployment	Choose one of the following: • Single Server Deployment: If you are setting up the **Minimum Production Deployment**. • Multiple Server Deployment: If you are setting up the **Enterprise Production Deployment**.
Target Landscape	(Only if you selected Multiple Server Deployment) Click the down arrow. The system displays the **Target Landscapes** dialog box. • On the first server that you configure, check Production and click OK. • On subsequent servers, check any combination of landscapes other than production and enter values in the subsequent fields. See below. *Target Landscapes Dialog Box* 
Production Configuration Server	(Only if you selected Multiple Server Deployment and selected anything other than Production in the **Target Landscape** field) The host or domain name of the configuration server for the production instance. Do not enter the information for any other instance; you must enter the information for the production instance.
Port	(Only if you selected Multiple Server Deployment and selected anything other than Production in the **Target Landscape** field) The port number of the production server. The default is 80, or 443 if you check Use SSL.
Use SSL	(Only if you selected Multiple Server Deployment and selected anything other than Production in the **Target Landscape** field) Check to enable SSL encryption to the server.
CentralConfig API Key (Production)	CentralConfig API key of the production environment. To retrieve the CentralConfig API Key,- navigate to the Configuration console > **Configure** > **Security Controls** > **API Keys** > **CentralConfigApiKey (Key groups)** > **CentralConfigApiKey)**.

2. Click OK to close the **Target Landscapes** dialog box.
3. Under **SQL High Availability**, select the **Multi Subnet Failover** checkbox if the SQL Servers fall under separate subnets.

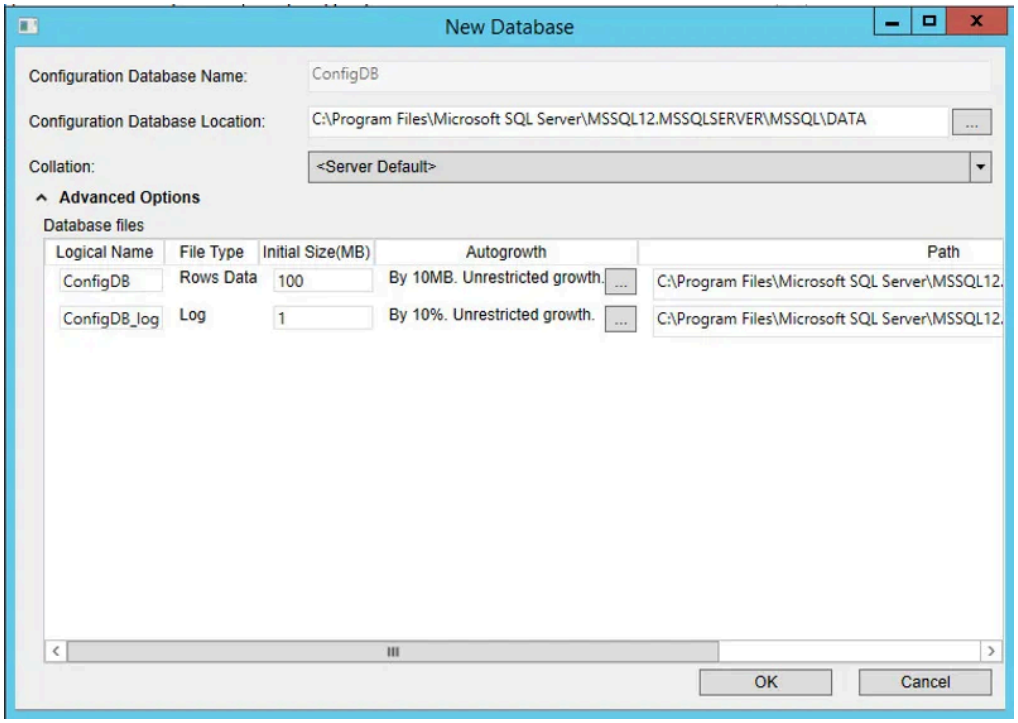
Multi-Subnet Failover in SQL Server is a feature designed to enhance high availability and disaster recovery in environments where Failover Cluster Instances (FCIs) or Always On Availability Groups span multiple subnets. This configuration ensures that SQL Server can efficiently handle failovers across geographically dispersed or separate subnets.

4. Enter values into the fields:

Parameter	Description
Configuration Database Server Name	The server where the configuration database is located. Do one of the following: • Accept the default value. • Enter the name of the server of the configuration database. • Browse for a server name by clicking the down arrow and selecting <Browse for more...>. The system displays any Microsoft SQL Server instances in your network. Choose a server and click OK. NOTE: Do not choose localhost unless you are setting up a **Demonstration or Proof-of-Concept Deployment**.
Configuration Database Name	The name of the configuration database. The default name is ConfigDB and you cannot change the name.
Authentication Method	Select one of the following: • Windows Authentication: Uses Microsoft Windows authentication. • SQL Authentication: Uses Microsoft SQL authentication. If you select this option, enter the user name and password below.
User Name	(Only if you selected SQL Authentication) The user name associated with the Microsoft SQL authentication. NOTE: If your deployment store FILESTREAM attachments, specify a configuration database user account that has a db_owner role on Microsoft SQL Server. See **Task 2: Setting up Neurons for ITSM**.
Password	(Only if you selected SQL Authentication) The password associated with the user name for the Microsoft SQL authentication.

5. Click Test Connection to test the connection to the configuration database server. The system displays *Connection successful!* if the connection is good. You must have a working connection to create a database. You cannot continue with the configuration if this connection does not work.
6. Depending if there is already a configuration database, click Load Configuration Database or Re-Create & Load Configuration DB. If the configuration database already exists, the system asks if you want to recreate the database. Click Yes. The system displays the **New Database** dialog box.

   New Database Dialog Box



7. Enter values into the fields:

Parameter	Description
Configuration Database Name	The name of the configuration database. The default name is ConfigDB and cannot be changed.
Configuration Database Location	The location for the configuration database. NOTE: If you have customized the existing configuration database, specify a different location; otherwise, the system overwrites your customization when you load the recreated database.
Collation	The set of rules that describe how to compare and sort strings, such as the order in which letters are sorted and whether case matters. NOTE: ISM application does not support case sensitive SQL Collation.
Advanced Options	
Logical Name	The logical name of the configuration database. This name is stored as a file name, and is a separate entity from the read-only database name that is displayed in Microsoft SQL Server Management Studio.
File Type	The file type, either database records or database logs. These values are read-only.
Initial Size (MB)	The initial file size, in MB, when creating the configuration database.
Autogrowth	Specifies how the database will expand when it reaches its maximum file size. We recommend having at least 1 GB or 10% to start with. Click ... to update the values.
Path	The location of the file. Click ... to update the location.
File Name	The name of the file.

8. Click OK. The system creates the configuration database and installs it on the configuration database server that you specified. Several progress dialog boxes appear as various Microsoft SQL scripts execute to create and load the configuration database. After the system loads the configuration database, the **Configuration Application** page displays *Configuration database loaded*.
9. Click Next. The system refreshes the **Configuration Application** page. (If you end up on the next page of the System Configuration Wizard, click Previous to return to the **Configuration Application** page.)
10. Enter values into the fields:

Parameter	Description
Configuration Server Domain Name	The fully-qualified domain name of the configuration server in this format: config.servername.com.
License Server	The name of the license server. We recommend using a fully-qualified domain name.
Port	The port number of the license server. The default is 80, or 443 if you check Use SSL.
Use SSL	Check if the license server uses SSL.
IvantiSM License File-Production	The name of the license file for the Neurons for ITSM production instance. Click Browse..., navigate to the license file, and click Open. License files have a .lic suffix. NOTE: The system displays either one or both of the license fields, depending on your landscape environment. At this point, you only need to enter one license, which allows the License Manager to start and run Neurons for ITSM. You can import additional licenses, if needed, at a later time from within the License Manager. See **Using the License Manager**.
IvantiSM License File-Non-Production	The name of the license file for the Neurons for ITSM non- production instance. Click Browse..., navigate to the license file, and click Open. License files have a .lic suffix. NOTE: The system displays either one or both of the license fields, depending on your landscape environment. At this point, you only need to enter one license, which allows the License Manager to start and run Neurons for ITSM. You can import additional licenses, if needed, at a later time from within the License Manager. See **Using the License Manager**.

11. Create the administrator account for the configuration database. You only need to create this account once. If you return to this page in the System Configuration Wizard later, these fields are not displayed.

    Enter values into the fields:

Parameter	Description
Login ID	The system automatically displays the login ID of HSWAdmin. You cannot change this.
Password	The password associated with the administrator. Record this password for future reference. If you lose the password, you must recreate and load the configuration database.
Confirm Password	Re-enter the password associated with the administrator.
First Name	The first name of the administrator.
Last Name	The last name of the administrator.
Email Address	The email address of the administrator.

12. Do one of the following:
    - If you chose Single Server Deployment, click Next. The system displays the Neurons for ITSM **Application** page.
    - If you chose Multiple Server Deployment, repeat all of the steps in this procedure to create a configuration database for each of your tenants. Then click Next. The system displays the Neurons for ITSM **Application** page.

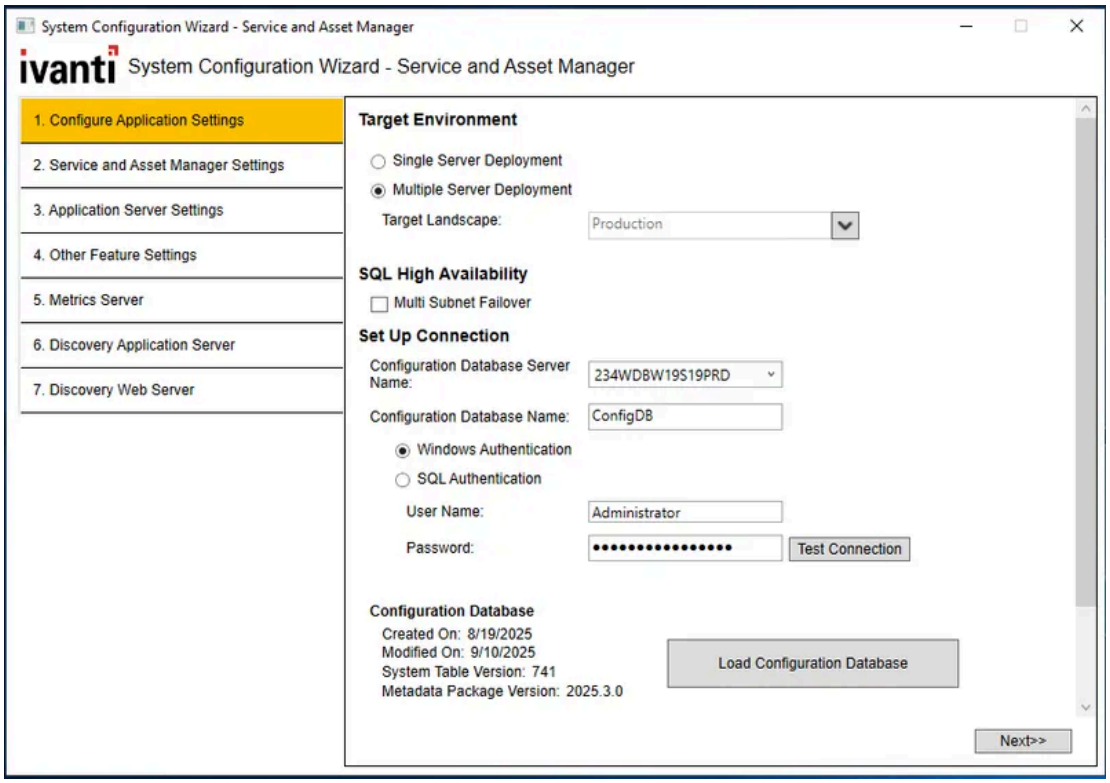
4.4.2.2. Configuring the Neurons for ITSM Application

On this page you will do the following:

- Specify the name and authentication method for the server where the application database resides.
- Load the demo database on a server other than the server where the application database resides. This option allows you to review the demo database while the application database continues to run in its current configuration.
- Specify the application name, attachment location, client authorization key, and the type of name to use when accessing the application.

The first time that you see this page, there is no application database, as shown in **Service Manager Application Page: No Database**.

Service Manager Application Page: No Database

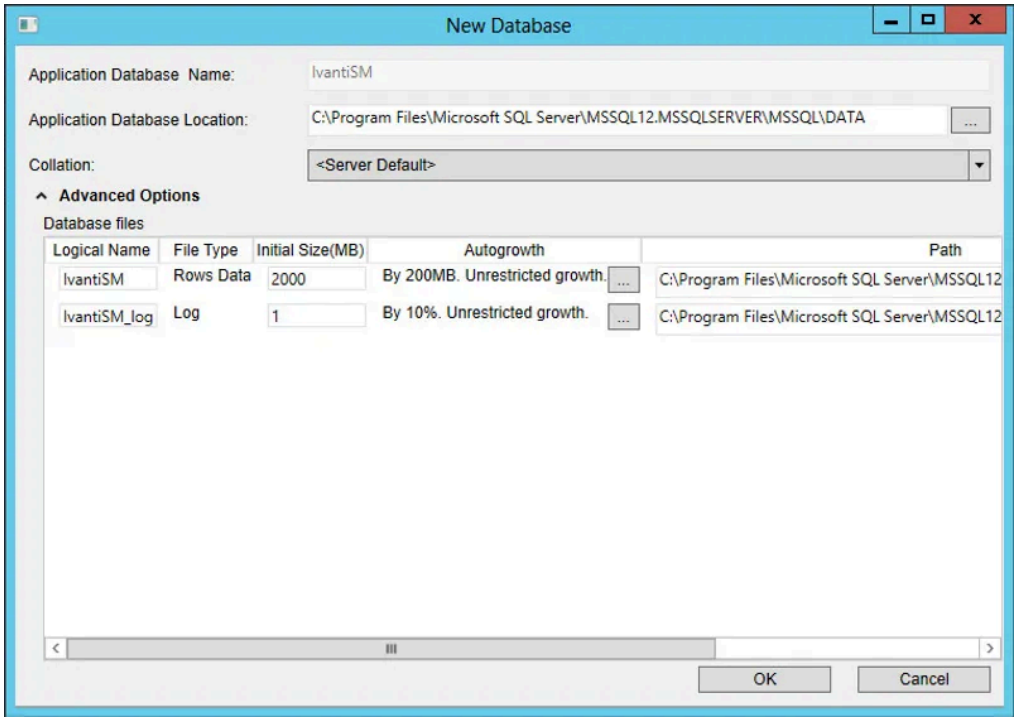


1. Enter values into the fields:

Parameter	Description
Application Database Server	The server where the application database is located. Usually this is the same server where the configuration database is located (specified earlier on the **Configuration Application** page). Do one of the following: 1. Accept the default value. 2. Enter the name of the server of the application database. Browse for a server name by clicking the down arrow and selecting <Browse for more...>. The system displays any Microsoft SQL Server instances in your network. Choose a server and click OK. Do not choose localhost unless you are setting up a **Demonstration or Proof-of-Concept Deployment**.
Application Database Name	The name of the application database. The default name is IvantiSM. The name that you specify here is used when the application database is created, and later, recreated.
Authentication Method	Select one of the following: • Windows Authentication: Uses Microsoft Windows authentication. • SQL Authentication: Uses Microsoft SQL authentication. If you select this option, enter the user name and password below.
User Name	(Only if you selected SQL Authentication) Enter the user name associated with the Microsoft SQL authentication.
Password	(Only if you selected SQL Authentication) Enter the password associated with the user name for the Microsoft SQL authentication.

3. Click Test Connection to test the connection to the application database server. The system displays *Connection successful!* if the connection is good. You must have a working connection to create a database. You cannot continue with the configuration if this connection does not work.
4. Depending if there is already an application database, click Load Application Database or Re-Create & Load Application DB. If the application database already exists, the system asks if you want to recreate the database. Click Yes. The system displays the **New Database** dialog box.

   New Database Dialog Box



5. Enter values into the fields:

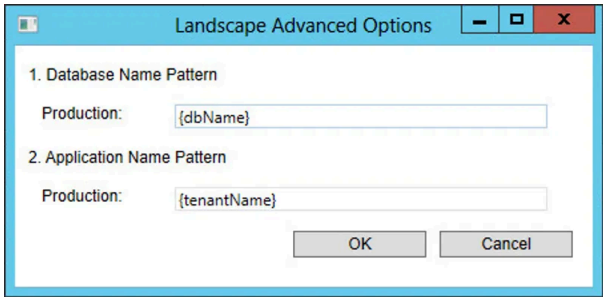
Parameter	Description
Application Database Name	The name of the application database. The default name is IvantiSM and cannot be changed.
Application Database Location	The location for the application database. NOTE: If you have customized the existing application database, specify a different location; otherwise, the system overwrites your customization when you load the recreated database.
Collation	The set of rules that describe how to compare and sort strings, such as the order in which letters are sorted and whether case matters. NOTE: ISM application does not support case sensitive SQL Collation.
Advanced Options	
Logical Name	The logical name of the application database. This name is stored as a file name, and is a separate entity from the read-only database name that is displayed in Microsoft SQL Server Management Studio.
File Type	The file type, either database records or database logs. These values are read-only.
Initial Size (MB)	The initial file size, in MB, when creating the application database.
Autogrowth	Specifies how the database will expand when it reaches its maximum file size. We recommend having at least 1 GB or 10% to start with. Click ... to update the values.
Path	The location of the file. Click ... to update the location.
File Name	The name of the file.

6. Click OK.

The system creates the application database and installs it on the application database server that you specified. The system displays several progress dialog boxes as the system executes various Microsoft SQL scripts that create and load the application database. After the system loads the application database, the system displays *Application Database loaded*.

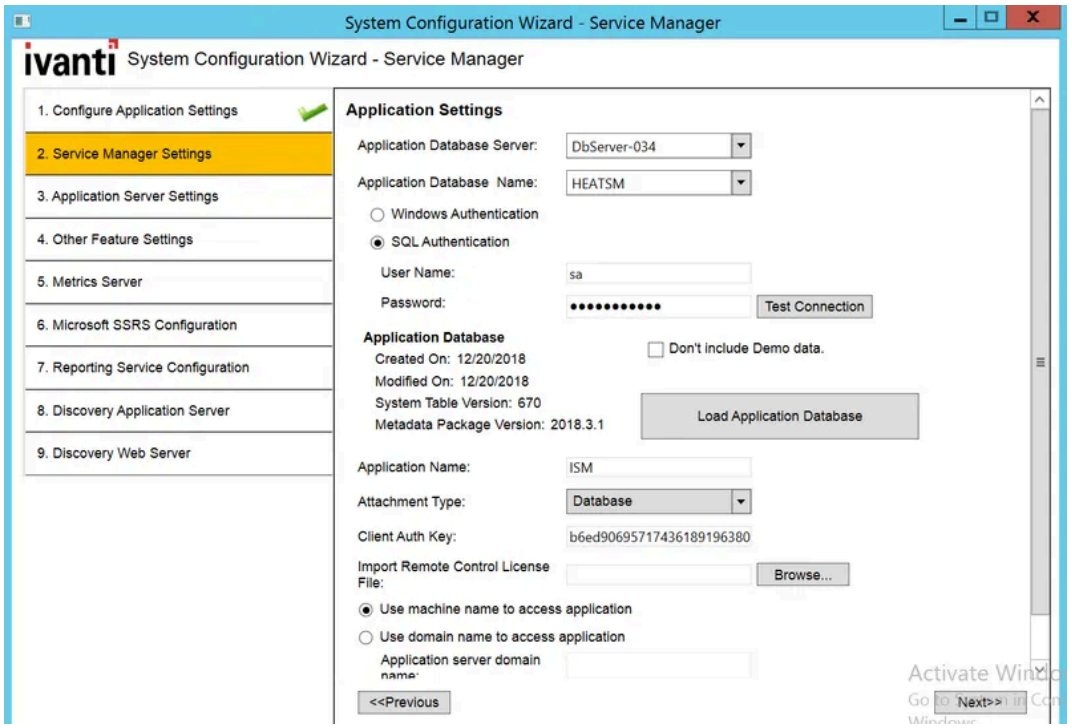
7. (Optional) Click **Advanced Options** to open the **Landscape Advanced Options** dialog box. See the "Adding a Landscape" topic in the *Operations Console User Guide for* Neurons for ITSM for more information about these patterns.

   Landscape Advanced Options Dialog Box



8. Click Next. The system refreshes the **Application** page. See **Application Page: Newly Created Database**. (If you end up on the next page of the System Configuration Wizard, click Previous to return to the **Application** page.)

   Application Page: Newly Created Database



9. Neurons for ITSM comes with optional transaction data that you can use to test and view the analytic metrics, financial, ITFM, and other features.
10. Enter values into the fields:

Parameter	Description
Application Name	The name of the application.
Attachment Type	The location of the folder on the application database server where any Neurons for ITSM attachments reside. Select one of the following: • Database • FILESTREAM • File System
Client Auth Key	The authentication key that is used when the web services API accesses Neurons for ITSM.
Import Remote Control License File	(Optional) If you are going to use the remote control feature, you must have a remote control license. • Contact Ivanti Software to get the license. • To import the remote control license file, click Browse... next to the **Import Remote Control License File** field. • Browse to the location of the license file and click OK to select it.
To access the application	Select whether to use the application database server machine name or the fully-qualified domain name for access to the application database server: • Use machine name to access Ivanti application: When other Neurons for ITSM components log into the application server, they use the machine name of the server. For example, they use **SERVER-01**. If you choose this option, the system detects the current machine name and uses it automatically; you do not need to specify the machine name. • Use domain name to access Ivantiapplication: When other Neurons for ITSM components log into the application server, they use the fully-qualified domain name of the server. For example, they use SERVER-01.company.com. If you plan on using SSL for connecting to the application server (configured later on the **Other Feature Settings** page), you must enter a fully-qualified domain name here, either for an individual machine or for a load-balanced instance.
Application server domain name	(Only if you selected Use domain name to access Ivanti application) The domain name of the application server.

11. Click Next to refresh the **Application** page. (If you end up on the next page of the System Configuration Wizard, click Previous to return to the **Application** page.)
12. Create the administrator account for the application database. This login ID and password allows you to access the Neurons for ITSM Service Desk Console and Configuration Console. See **Logging into Neurons for ITSM**. You only need to create this account once. If you return to this page in the System Configuration Wizard later, these fields are not displayed.

    Enter values into the fields:

Parameter	Description
Login ID	The system automatically displays the login ID of IvantiAdmin. You cannot change this.
Password	The password associated with the administrator. Record this password for future reference. If you lose the password, you must recreate and load the application database.
Confirm Password	Re-enter the password associated with the administrator.
First Name	The first name of the administrator.
Last Name	The last name of the administrator.
Email Address	The email address of the administrator.

13. If you chose Multiple Server Deployment on the **Configuration Application** page, your system has multiple tenants. Each tenant must have an administrator account, so that the administrator can access the Neurons for ITSM Service Desk Console and Configuration Console.

    To create a new administrator account for another tenant instance, do the following:

    a. In the **Application Database Name** field at the top of this page, enter the next tenant to get an account.

    For example, if you named your production tenant IvantiSM, your staging tenant IvantiSM-STG, and your UAT tenant IvantiSM-UAT, then the first time that you are on this page, enter IvantiSM in the **Application Database Name** field. The second time that you are on this page, enter IvantiSM-STG in the **Application Database Name** field. The third time that you are on this page, enter IvantiSM-UAT in the **Application Database Name** field.

    b. Create the administrator account for the application database for the Configuring Other Feature Settings tenant. See step 11 above.

    c. Click **Next**. The system displays the **Application Server Settings** page.

    d. Click Previous to return to the Neurons for ITSM **Application** page.

    e. Repeat steps a. through d. for the next tenant.

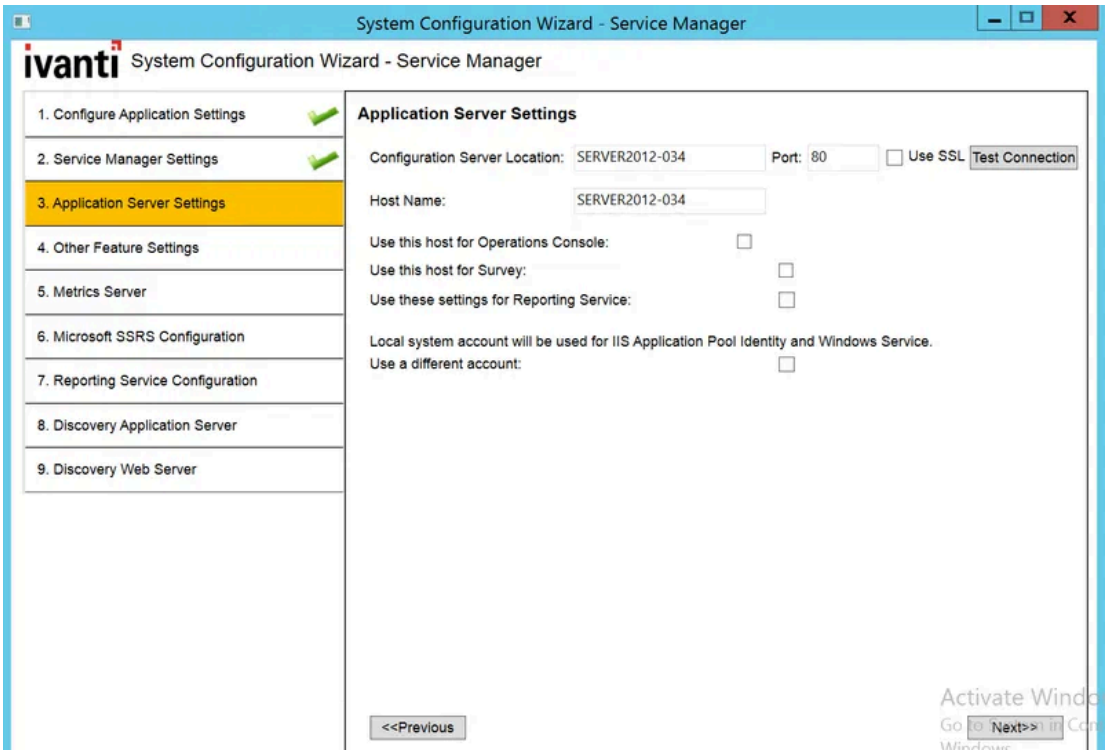
14. After all of your tenants have an administrator account, click Next Configuring the Reporting Feature to advance to the **Application Server Settings** page.

4.4.2.3. Configuring the Application Server Settings

You set connections to other databases in Neurons for ITSM on the **Application Server Settings** page. You will do the following:

- Specify the location of the configuration server and the host name.
- Specify if you will use this server for surveys or the reporting feature.
- If you use Windows Integrated Security for the Application Server, specify the Windows domain credentials for the service account.

Application Server Settings Page



1. Enter values into the fields:

Parameter	Description
Configuration Server Location	The host name or domain name of the configuration server. The default value is the name of the server that you are logged into now. If the host that you are logged into now is *not* the configuration server, enter the machine name or fully-qualified domain name of your Configuration Server. If you check Use SSL below, you must enter the host name of the configuration server.
Port	The port number of the configuration server. The default is 80, or 443 if you check Use SSL.
Use SSL	Check to use SSL for connections to the configuration server. NOTE: We do not recommend enabling SSL on the configuration server until you have fully tested Neurons for ITSM to ensure that it works with SSL. For information on configuring Neurons for ITSM with SSL, see **Optional SSL Configuration**.

2. Click Test Connection to test the connection to the configuration server. The system displays a success or failure message. Click OK to close the message.
3. Enter values in the fields:

Parameter	Description
Host Name	The location of the system that hosts the Application Server. The default value is the server that you are logged into now. Do one of the following: • If you are installing all Neurons for ITSM components on the same host, accept the default value. • If you enter a different location, it must be a machine name. Do *not* enter a fully-qualified domain name.
Use this host for Ivanti Operations Console	Uses this server for the Ivanti Service Manager Operations Console. NOTE: If you have an **Enterprise Production Deployment**, we recommend that you only install and use the Neurons for ITSM Operations Console on the staging (STG) instance of the tenant.
Landscape Type	(Only if you checked Use this host for Ivanti Operations Console) Displays the landscape type and cannot be changed.
Database Server	(Only if you checked Use this host for Ivanti Operations Console) Displays the database server and cannot be changed.
Operations Console Backup Location	(Only if you checked Use this host for Ivanti Operations Console) The backup location for the Neurons for ITSM Operations Console. The backup location can either be on a database server or on a network folder. Do one of the following: 1. Accept the default location. 2. Enter a new location. 3. Click Browse... and navigate to a new location.
Test Backup & Restore	(Only if you checked Use this host for Ivanti Operations Console) The database to test to ensure you can back up and restore it. Select one from the drop-down list. We recommend that you select the configuration database since it is small. Click test to test the connection. The system displays a message with the results of the test. Click OK to close the message.
Use this host for Survey	Uses this server for the Neurons for ITSM survey component. If you use Windows Authentication for the configuration server, the system prompts you for the Windows domain account and credentials.
Use these settings for Reporting Service	Uses this server for the Neurons for ITSM reporting feature. Usually, you configure the reporting feature on the **Microsoft SSRS Configuration** and **Ivanti Reporting Service Configuration** pages. However, if you have two different domains and the reporting feature server cannot reach the web server, you can configure these settings to enable the servers to communicate.
Use SSL for Reports	(Only if you checked Use these settings for Reporting Service) Uses SSL for reports.
Use Windows Authentication for SSRS	(Only if you checked Use these settings for Reporting Service) Uses Windows authentication for reports. If you plan on sharing the same Microsoft SSRS instance with other applications besides the reporting feature, you must use Windows authentication.
Web Service Virtual Directory	(Only if you checked Use these settings for Reporting Service) The name of the web service virtual directory.
Ivanti Web Application Pool	(Only if you checked Use these settings for Reporting Service) The application pool identity for the Neurons for ITSM web application pool. Select one of the following:

Parameter	Description
Identity Account	• Keep existing account • Specify new account If you select Specify new account, click Set to enter the credentials.
User Name	(Only if you checked Use these settings for Reporting Service and selected Specify new account for the **Ivanti Web Application Pool Identity Account** field) The user name for the application pool identity for the Neurons for ITSM web application pool.
Password	(Only if you checked Use these settings for Reporting Service and selected Specify new account for the **Ivanti Web Application Pool Identity Account** field) The password associated with the user above. Click OK.

4. If you are using Microsoft SQL authentication for the configuration server, the system displays this message: "Local system account will be used for IIS Application Pool Identity and Windows Service". See **Service Account** for more information about this account.

To use a different account, do the following:

a. Check Use a different account.

b. Click Set.

c. Enter the user name for the service account.

d. Enter the password associated with the service account.

e. Click OK.

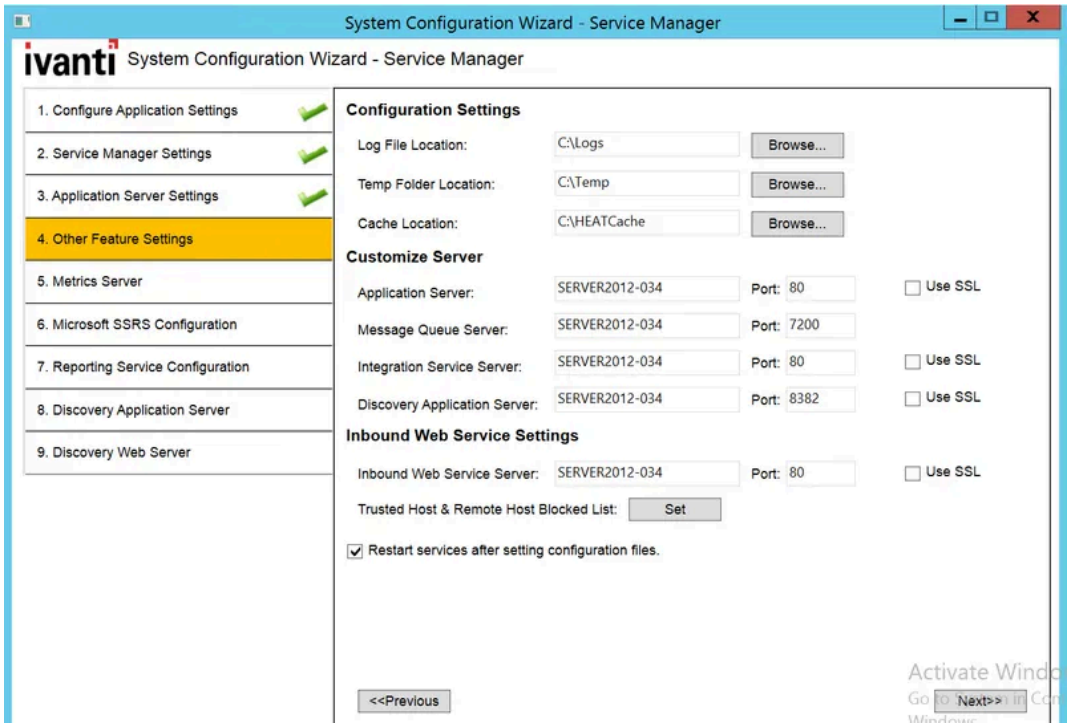
5. Each tenant in a multiple server deployment must have an administrator account. If you selected Multiple Server Deployment on the **Configuration Application** page, click Previous to return to the Neurons for ITSM application page and create additional tenant accounts.

4.4.2.4. Configuring Other Feature Settings

On this page you will do the following:

- Specify the log file and temporary file locations on the configuration server.
- Specify different servers and ports and whether to use SSL for the other Neurons for ITSM servers.
- Specify the inbound web server information.

Other Feature Settings Page



1. Enter values into the fields:

Parameter	Description
Configuration Settings	
Log File Location	The folder on the server that you are logged into now where the log files are located. Do one of the following: 1. Accept the default value. 2. Enter the name of the folder where the log files are located. 3. Browse for a different folder by clicking Browse.... Select a new folder and click OK. You have to provide the path for nlog.config files as well to store the non-configrable files.
Temp Folder Location	The folder on the server that you are logged into now where the temporary files are located. The temporary folder is used to cache JavaScript files and to enable integration server and Neurons for ITSM Operations Console features. Do one of the following: 1. Accept the default value. 2. Enter the name of the folder where the temporary files are located. 3. Browse for a different folder by clicking Browse.... Select a new folder and click OK.
Cache Location	The folder on the server that you are logged into now where the cached files are located. Do one of the following: 1. Accept the default value. 2. Enter the name of the folder where the temporary files are located. 3. Browse for a different folder by clicking Browse.... Select a new folder and click OK.
Customize Server	
Application Server	Where the application server is located. The default value is the server that you are logged into now. Do one of the following: 1. Accept the default value. 2. Enter the name where the application server is located. 3. If you use SSL, you must enter a fully-qualified domain name. NOTE: If you choose a different server for the application server and check Use SSL, SSL is only enabled for the application server. It does not enable SSL for the configuration server.
Port	The port number of the application server. The default is 80, or 443 if you check Use SSL.
Use SSL	Enables SSL encryption to the application server.
Message Queue Server	Where the message queue server is located. The default value is the server that you are logged into now. Do one of the following: 1. Accept the default value. 2. Enter the name where the Neurons for ITSM message queue server is located. If you use SSL, you must enter a fully-qualified domain name.
Port	The port number of the message queue server. The default is 7200.

Parameter	Description
Use SSL	Enables SSL encryption to the message queue server.
Integration Server	Where the integration server is located. The default value is the server that you are logged into now. Do one of the following: 1. Accept the default value. 2. Enter the server where the Neurons for ITSM integration server is located. If you use SSL, you must enter a fully-qualified domain name.
Port	The port number of the Neurons for ITSM integration server. The default is 80, or 443 if you check Use SSL.
Use SSL	Enables SSL encryption to the Neurons for ITSM integration server.
Discovery Application Server	Where the Discovery application server is located. The default value is the server that you are logged into now. Do one of the following: • Accept the default value. • Enter the server where the Discovery application server is located. • If you use SSL, you must enter a fully-qualified domain name.
Port	The port number of the Discovery application server. The default is 8382, or 443 if you check Use SSL.
Use SSL	Enables SSL encryption to the Discovery application server.
Inbound Web Service Settings	
Inbound Web Server	Where the inbound web server is located. The default value is the server that you are logged into now. Do one of the following: 1. Accept the default value. 2. Enter the server where the Neurons for ITSM inbound web server is located. If you use SSL, you must enter a fully-qualified domain name.
Port	The port number of the Neurons for ITSM inbound web server. The default is 80, or 443 if you check Use SSL.
Use SSL	Enables SSL encryption to the Neurons for ITSM inbound web server.

You can set a list of remote hosts to block and a list of remote hosts to trust. By default, the list is empty.

3. To designate a list of blocked servers, do the following:

   a. Next to the **Trusted Host & Remote Host Blocked List** field, click Set.

   b. In the **Remote Host Blocked List** area, click the plus icon.

   c. Enter the starting IP address and the ending IP address for the blocked servers.

   d. If you are blocking web scripts, check Is Web Script.

   e. Click OK.

4. To designate a list of trusted servers, do the following:

   a. Next to the **Trusted Host & Remote Host Blocked List** field, click Set.

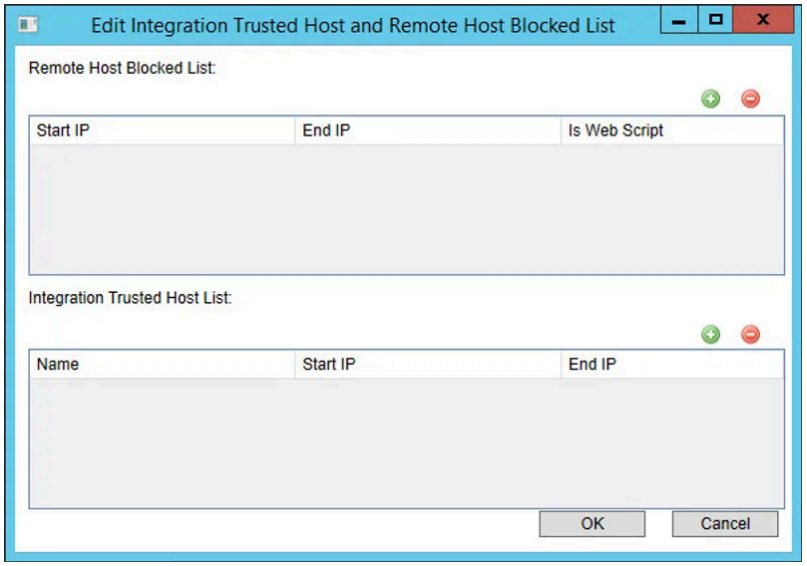
b. In the **Integration Trusted Host List** area, click the plus icon.

c. Enter a name for the trusted host list.

d. Enter the starting IP address and the ending IP address for the trusted servers.

e. Click OK.

Edit Integration Trusted Host and Remote Host Blocked List Dialog Box



To update the list at a later time, run the System Configuration Wizard, advance to the **Other Feature Settings** page, click Set, and make your changes. When you are done, click OK.

5. To restart the Neurons for ITSM services after you exit the System Configuration Wizard, check Restart services after setting configuration files.
6. Verify that the Neurons for ITSM services restart after the wizard closes.
7. If you did not check Restart services after setting configuration files, restart the Neurons for ITSM services manually.
8. Click Next. The system displays the **Metrics Server** page.

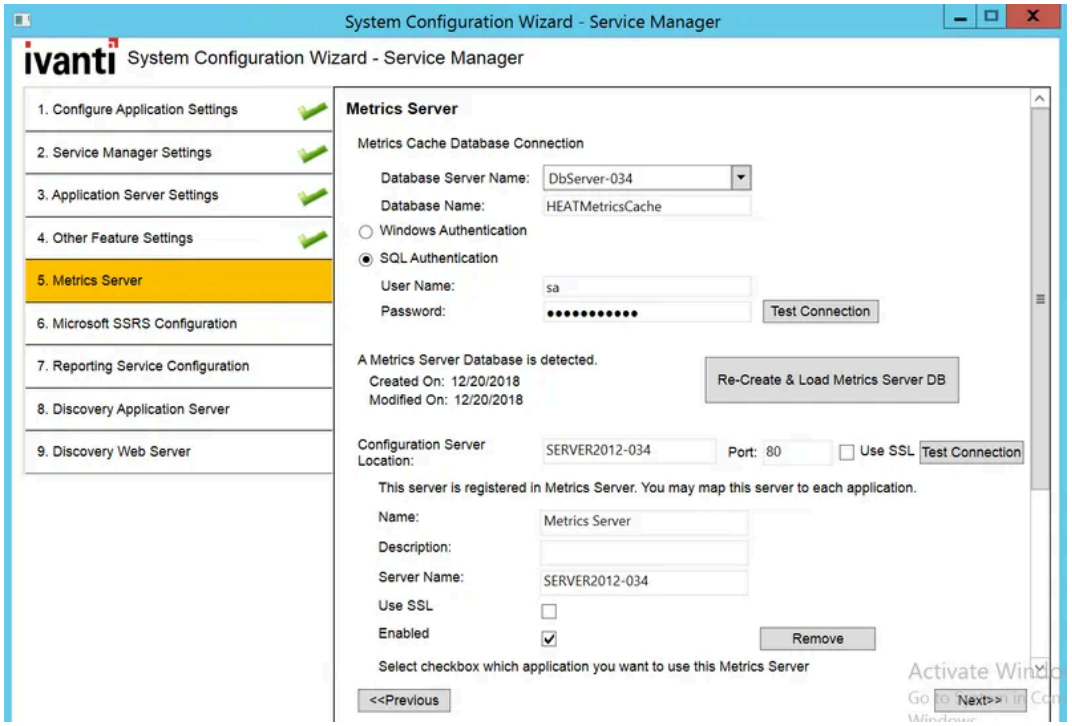
4.4.2.5. Configuring the Metrics Server

Perform these steps to configure one or more Metrics Server databases, which you can use to analyze data. Your deployment can have multiple Metrics Server databases to improve performance.

On this page, you will do the following:

- Specify the locations for the log file and the configuration server.
- Map applications to the Metrics Server.

Metrics Server Page



1. Enter values into the fields:

Parameter	Description
Metrics Cache Database Connection	
Database Server Name	Where the metrics cache database server is located. The default value is the server where the application server is located. Do one of the following: • Accept the default value. • Enter the name of the server of the metrics cache database. • Browse for a server name by clicking the down arrow and selecting <Browse for more...>. The system displays all available database servers in the network for your connection. Choose a server and click OK. NOTE: Do not choose localhost unless you are setting up a **Demonstration or Proof-of-Concept Deployment**.
Database Name	The name of the metrics cache database. Do one of the following: • Accept the default value. • Enter the name of the metrics cache database.
Authentication Method	Select one of the following: • Windows Authentication: Uses Microsoft Windows authentication. • SQL Authentication: Uses Microsoft SQL authentication. If you select this option, enter the user name and password below. NOTE: The metrics cache database server is not compatible with SSL if your system has Windows authentication set up.
User Name	(Only if you selected SQL Authentication) The user name associated with the Microsoft SQL authentication.
Password	(Only if you selected SQL Authentication) The password associated with the user name for the Microsoft SQL authentication.

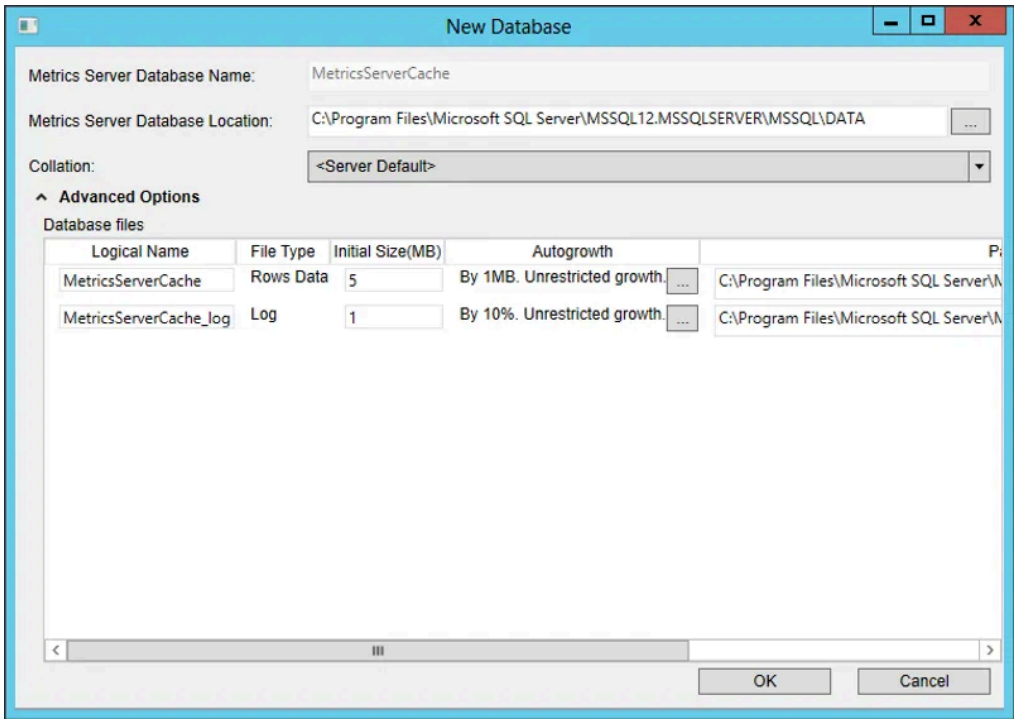
2. Click Test Connection to test the connection to the metrics cache database server. The system displays a success or failure message below the **Test Connection** button. You must have a working connection to create a database. You cannot continue with the configuration if this connection does not work.
3. Do one of the following:
    - If this is the initial setup, go to **Creating a New Metrics Server Database**.
    - If your system already has a metrics server database, go to **Recreating the Metrics Server Database**.

Creating a New Metrics Server Database

An initial installation does not have a metrics server database. Follow these steps to create one.

1. Click Create & Load Metrics Server DB. The system displays the **New Database** dialog box.

   New Database Dialog Box



2. In the **Location** field, select a location for the metrics server database.
3. Click Advanced Options to view and reconfigure these metrics server database settings:
   - Logical name: The logical name of the metrics database. This name is stored as a file name, and is a separate entity from the read-only database name that is displayed in Microsoft SQL Server Management Studio.
   - File type: The file type, either database records or database logs. These values are read-only.
   - Initial size in MB: The initial file size, in MB, when creating the metrics database.
   - Autogrowth size: Specifies how the database will expand when it reaches its maximum file size. We recommend having at least 1 GB or 10% to start with. Click ... to update the values.
   - The location of the database: The location of the file. Click ... to update the location.
   - File name: The name of the file.
4. Click OK in the **New Database** dialog box.

The **Configuration Server Location** field indicates the location of the system that hosts the configuration server. The default value is the name of the host that you are logged into now.

5. If this is the system that hosts the configuration server, accept the default value.

If the host that you are logged into now is not the configuration server, enter the machine name or fully-qualified domain name of your configuration server.

If you use SSL, you enter the fully-qualified domain name of the configuration server.

6. Enter the port number in the **Port** field. The default value is 80. If you click Use SSL, it changes to 443.
7. Select whether to use SSL for connections to the configuration server.

If you use SSL, you must provide a fully-qualified domain name in the **Configuration Server Location** field.

8. Click Test Connection to test the connection to the configuration server. A pop-up message notifies you of success or failure.

> We do not recommend enabling SSL on the configuration server until you have fully tested Neurons for ITSM to ensure that it works with SSL.
>
> For information on configuring Neurons for ITSM with SSL, see **Optional SSL Configuration**.

9. Enter the following information to register a server in a multi-server deployment to the metrics server database:
    - Accept the default server name or enter a different name.
    - Optional. Enter a description of the server.
    - The default server name value is the name of the host that you are logged into now. You must use this name or localhost.
    - Check the box to use SSL encryption.
    - Check the box to enable this metrics server.
10. Click Save.
11. To map an application to the new metrics server to an application, check Map. You can reference the same metrics server to multiple Neurons for ITSM applications.

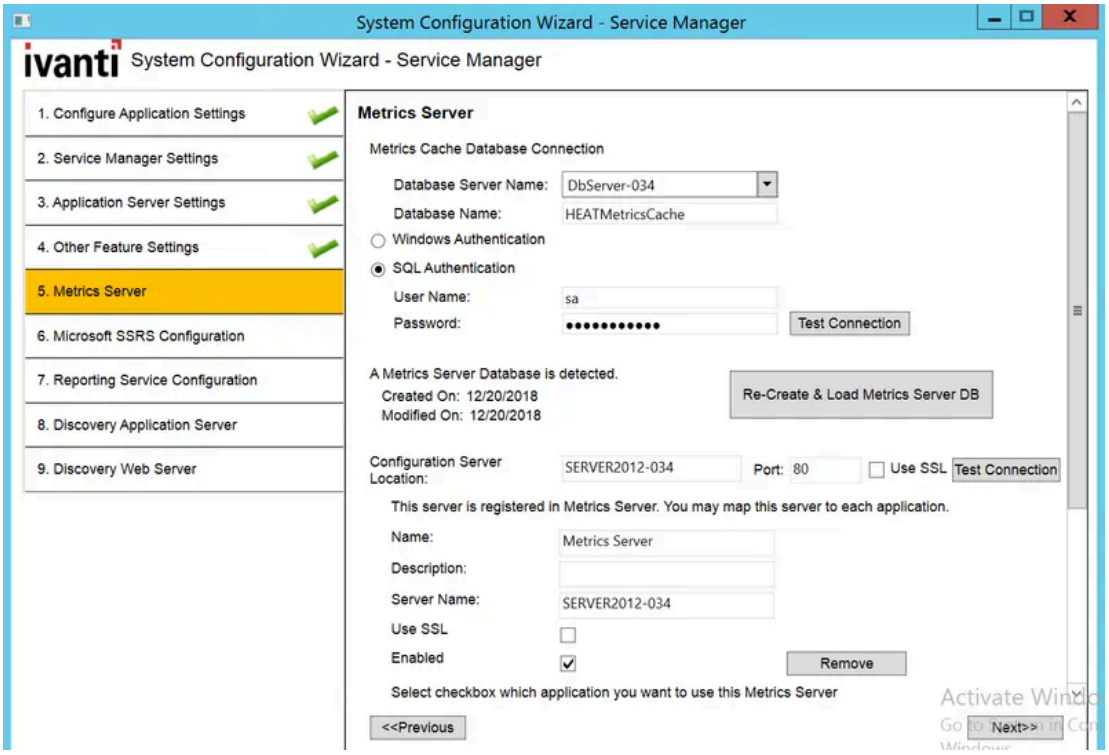
> To change this server at a later time, run the System Configuration Wizard, advance to the **Metrics Server** page, and click Remove.

12. Click Next. The system displays the **Discovery Application Server** page.

Recreating the Metrics Server Database

If you are upgrading or repairing Neurons for ITSM, you already have a metrics server database, and the **Metrics Server** page looks like **Metrics Server Page**.

Metrics Server Page



1. To recreate the metrics server database, click Re-Create & Load Metrics Server DB. The system displays the **New Database** dialog box.
2. Perform steps 2 through 12 in **Creating a New Metrics Server Database**.

If you have customized the existing metrics server database, specify a different location in the **Metrics Server Database Location** field in the **New Database** dialog box. Otherwise, the system overwrites your customization when you load the recreated database.

4.4.2.6. Configuring the Discovery Application Server

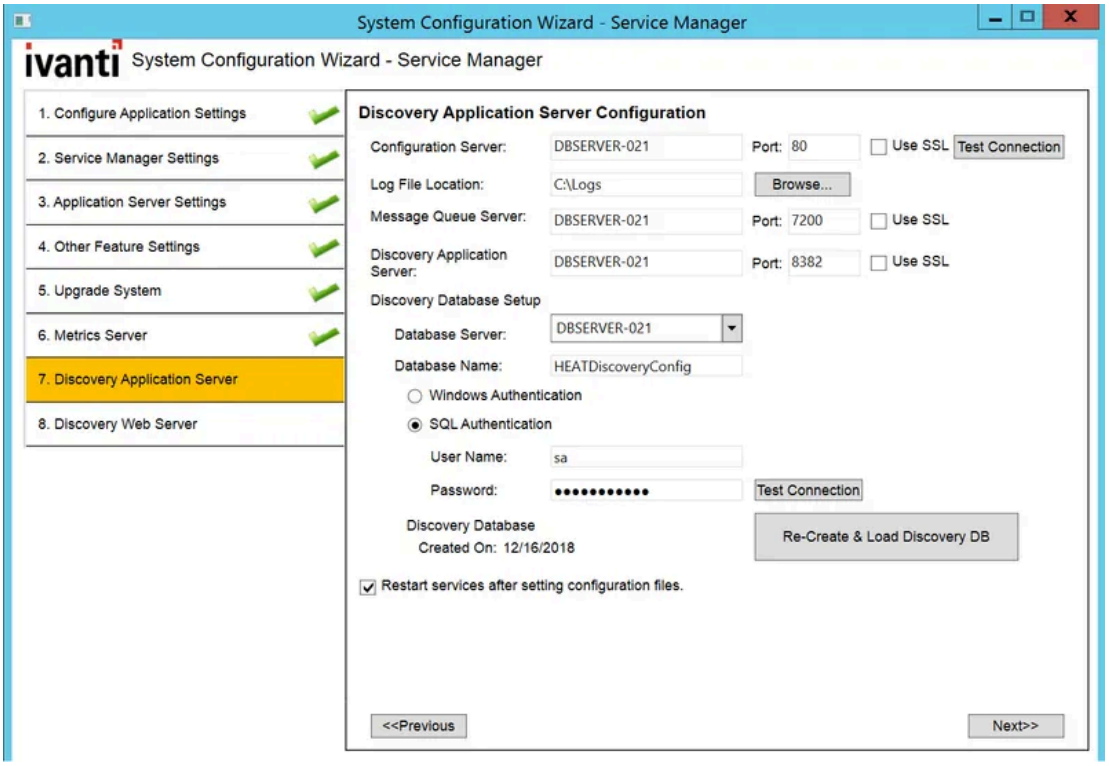
On this page you will specify information about the configuration of the Discovery application server and the Discovery configuration database. Follow these steps:

- Starting the Configuration for the Discovery Application Server
- Configuring a System without a Discovery Configuration Database (Creating a New One)
- Configuring a System with a Discovery Configuration Database (Editing One that Already Exists)

Starting the Configuration for the Discovery Application Server

The first time that you see this page, there is no Discovery configuration database, as shown in Discovery Application Server Page: No Database.

Discovery Application Server Page: No Database



1. Enter values into the fields:

Parameter	Description
Configuration Server	The host name or fully-qualified domain name of the configuration server. The default value is the name of the host that you are logged into now. Do one of the following: • Accept the default value. • Enter the name of the server of the configuration server. If you use SSL, you must enter a fully-qualified domain name.
Port	The port number of the configuration server. The default is 80, or 443 if you check Use SSL.
Use SSL	Enables SSL encryption to the configuration server.

2. Click Test Connection to test the connection to the configuration server. The system displays a success or failure message. Click OK to close the message. You must have a working connection to the server. You cannot continue with the configuration if this connection does not work.
3. Enter values into the fields:

Parameter	Description
Log File Location	The folder on the server that you are logged into now where the log files are located. The default location for the log files is C:\Logs. Do one of the following: • Accept the default value. • Enter the name of the folder where the log files are located. • Browse for a different folder by clicking Browse.... Select a new folder and click OK.
Message Queue Server	Where the Neurons for ITSM message queue server is located. The default value is the server that you are logged into now. Do one of the following: • Accept the default value. • Enter the name where the Neurons for ITSM message queue server is located. If you use SSL, you must enter a fully-qualified domain name.
Port	The port number of the Neurons for ITSM message queue server. The default is 7200.
Use SSL	Enables SSL encryption to the Neurons for ITSM message queue server.
Discovery Application Server	Where the Discovery application server is located. The default value is the server that you are logged into now. Do one of the following: • Accept the default value. • Enter the name where the Discovery application server is located. If you use SSL, you must enter a fully-qualified domain name.
Port	The port number of the Discovery application server. The default is 80, or 443 if you check Use SSL.
Use SSL	Enables SSL encryption to the Discovery application server.
Discovery Database Setup	
Database Server	The name of the system proposed for the Discovery configuration database. The default value is the server that you are logged into now. Do one of the following: • Accept the default value. • Enter the name of the Discovery configuration database server. • Browse for a server name by clicking the down arrow and selecting <Browse for more...>. The system displays any Microsoft SQL Server instances in your network. Choose a server and click OK.
Database Name	The name of the Discovery configuration database. The default name is HEATDiscoveryConfig and you cannot change the name.

Parameter	Description
Authentication Method	There are two authentication methods for the Discovery configuration database, one for creation and one for access. Select the authentication method for creating the Discovery configuration database: • Windows Authentication: Uses Microsoft Windows authentication. • SQL Authentication: Uses Microsoft SQL authentication. If you select this option, enter the user name and password below.
User Name	(Only if you selected SQL Authentication) The user name associated with the Microsoft SQL authentication.
Password	(Only if you selected SQL Authentication) The password associated with the user name for the Microsoft SQL authentication.

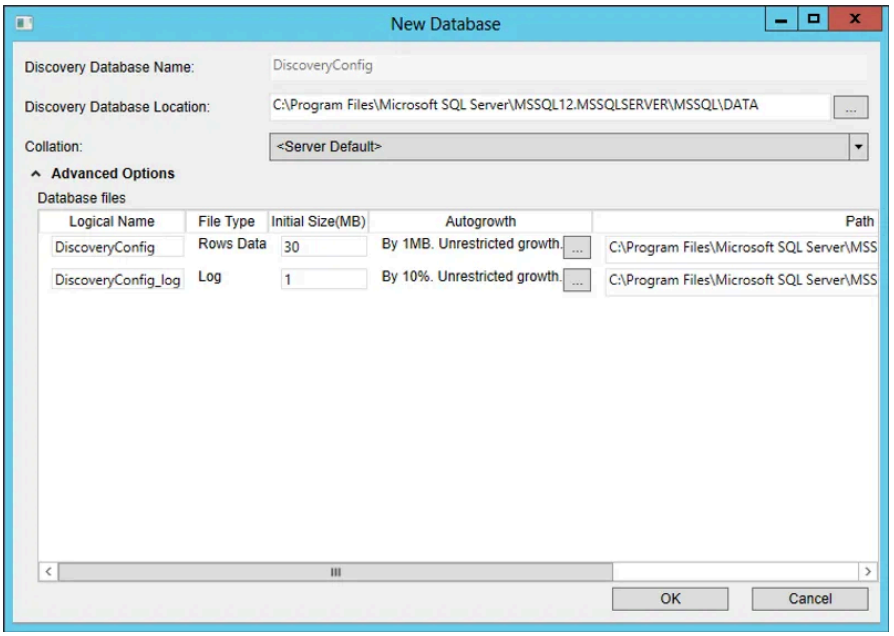
4. Click Test Connection to test the connection to the Discovery configuration database server. The system displays a success or failure message. Click OK to close the message. You must have a working connection to create a database. You cannot continue with the configuration if this connection does not work.
5. Do one of the following:
    - If this is the initial setup, go to **Configuring a System without a Discovery Configuration Database (Creating a New One)**.
    - If your system already has a Discovery configuration database, go to **Configuring a System with a Discovery Configuration Database (Editing One that Already Exists)**.

Configuring a System without a Discovery Configuration Database (Creating a New One)

An initial installation does not have a Discovery configuration database. The system displays this: "No Discovery (HEATDiscoveryConfig) Database is detected."

Follow these steps to create one.

1. Enter values into the fields:

Parameter	Description
Create & Load Discovery DB	Click to create and load a new Discovery configuration database. The system displays the **New Database** dialog box. *New Database Dialog Box* 
Discovery Database Name	The name of the Discovery configuration database. The default name is HEATDiscoveryConfig and you cannot change the name.
Discovery Database Location	The location for the Discovery configuration database.
Collation	The set of rules that describe how to compare and sort strings, such as the order in which letters are sorted and whether case matters. NOTE: ISM application does not support case sensitive SQL Collation.
Advanced Options	
Logical Name	The logical name of the Discovery configuration database. This name is stored as a file name, and is a separate entity from the read-only database name that is displayed in Microsoft SQL Server Management Studio.
File Type	The file type, either database records or database logs. These values are read-only.
Initial Size (MB)	The initial file size, in MB, when creating the Discovery configuration database.
Autogrowth	Specifies how the database will expand when it reaches its maximum file size. We recommend having at least 1 GB or 10% to start with. Click ... to update the values.
Path	The location of the file. Click ... to update the location.
File Name	The name of the file.

2. Click OK in the **New Database** dialog box. The system creates the Discovery configuration database.

3. To restart the Neurons for ITSM services after you finish the installation, check Restart services after setting configuration files..
4. Click Next. The system displays the **Discovery Web Server** page.

Configuring a System with a Discovery Configuration Database (Editing One that Already Exists)

If you are upgrading or repairing Neurons for ITSM you already have a Discovery configuration database.

1. To recreate the Discovery configuration database, click Re-Create & Load Discovery DB. The system displays a confirmation message.

> If you have customized the existing Discovery configuration database, specify a different location in the **Discovery Database Location** dialog box. Otherwise, the system overwrites your customization when you recreate the database.

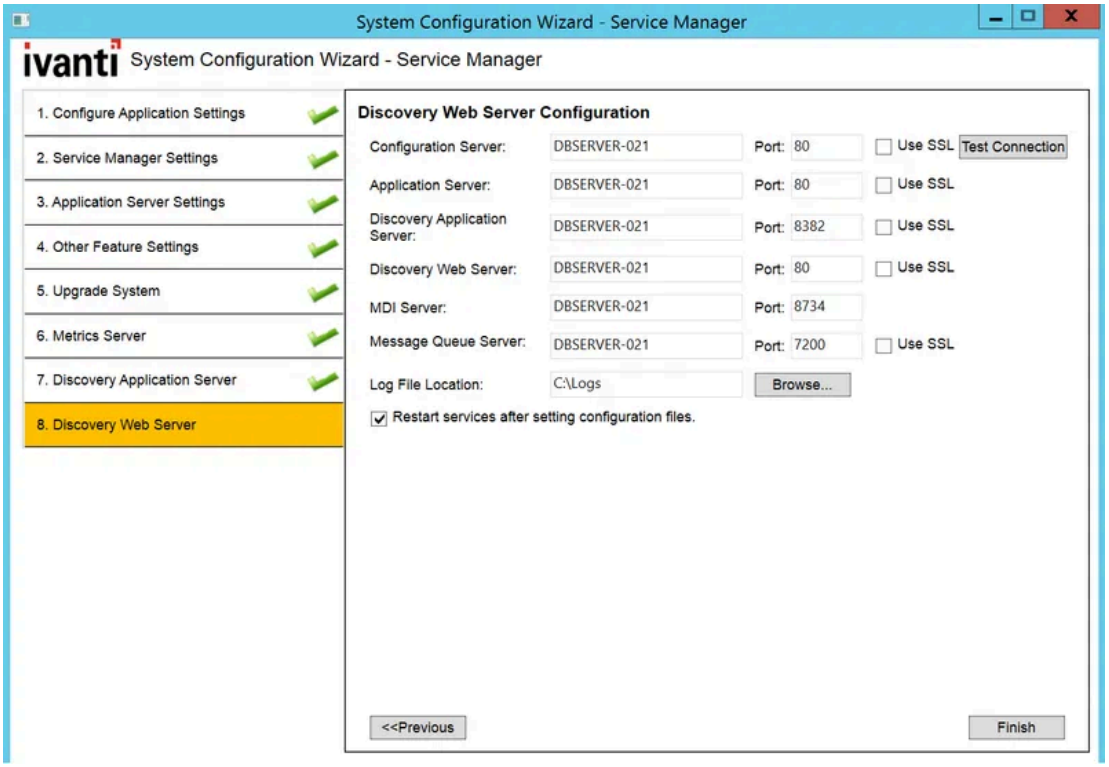
2. Click OK at the confirmation message.
3. To restart the Neurons for ITSM services after you finish the installation, check Restart services after setting configuration files..
4. Click Next. The system displays the **Discovery Web Server** page.

4.4.2.7. Configuring the Discovery Web Server

Configure the relationship between Neurons for ITSM, the Discovery web server, and the Discovery application server from the **Discovery Web Server** page of the System Configuration Wizard.

The system displays the **Discovery Web Server** page. See Discovery Web Server Page.

Discovery Web Server Page



1. Enter values into the fields:

Parameter	Description
Configuration Server	The host name or fully-qualified domain name of the configuration server. The default value is the name of the host that you are logged into now. Do one of the following: • Accept the default value. • Enter the name of the server of the configuration server. If you use SSL, you must enter a fully-qualified domain name.
Port	The port number of the configuration server. The default is 80, or 443 if you check Use SSL.
Use SSL	Enables SSL encryption to the configuration server.

2. Click Test Connection to verify the connection to the configuration server.
3. Enter values into the fields:

Parameter	Description
Application Server	Where the application server is located. The default value is the server that you are logged into now. Do one of the following: • Accept the default value. • Enter the name where the application server is located. If you use SSL, you must enter a fully-qualified domain name.
Port	The port number of the application server. The default is 80, or 443 if you check Use SSL.
Use SSL	Enables SSL encryption to the application server.
Discovery Application Server	Where the Discovery application server is located. The default value is the server that you are logged into now. Do one of the following: • Accept the default value. • Enter the name where the Discovery application server is located. If you use SSL, you must enter a fully-qualified domain name.
Port	The port number of the Discovery application server. The default is 8382, or 443 if you check Use SSL.
Use SSL	Enables SSL encryption to the Discovery application server.
Discovery Web Server	Where the Discovery web server is located. The default value is the server that you are logged into now. Do one of the following: • Accept the default value. • Enter the name where the Discovery web server is located. If you use SSL, you must enter a fully-qualified domain name.
Port	The port number of the Discovery web server. The default is 80, or 443 if you check Use SSL.
Use SSL	Enables SSL encryption to the Discovery web server.
MDI Server	Where the MDI server is located. (The Mobile Device Inventory [MDI] service is bundled with Discovery.) The default value is the server that you are logged into now. Do one of the following: • Accept the default value. • Enter the name where the MDI server is located.
Port	The port number of the MDI server. The default is 8734.
Message Queue Server	Where the message queue server is located. The default value is the server that you are logged into now. Do one of the following: • Accept the default value. • Enter the name where the Neurons for ITSM message queue server is located. If you use SSL, you must enter a fully-qualified domain name.
Port	The port number of the Neurons for ITSM message queue server. The default is 7200.
Use SSL	Enables SSL encryption to the Neurons for ITSM message queue server.

Parameter	Description
Log File Location	The folder on the server that you are logged into now where the log files are located. The default location for the log files is C:\Logs on the Discovery web server. Do one of the following: • Accept the default value. • Enter the name of the folder where the log files are located. • Browse for a different folder by clicking Browse.... Select a new folder and click OK.

4. To restart the Neurons for ITSM services after you exit the System Configuration Wizard, check Restart services after setting configuration files.
5. Verify that the Neurons for ITSM services restart after the wizard closes.
6. If you did not check Restart services after setting configuration files, restart the Neurons for ITSM services manually.

4.4.2.8. Finishing the System Configuration

When you have finished configuring the ISM Discovery web server, do the following:

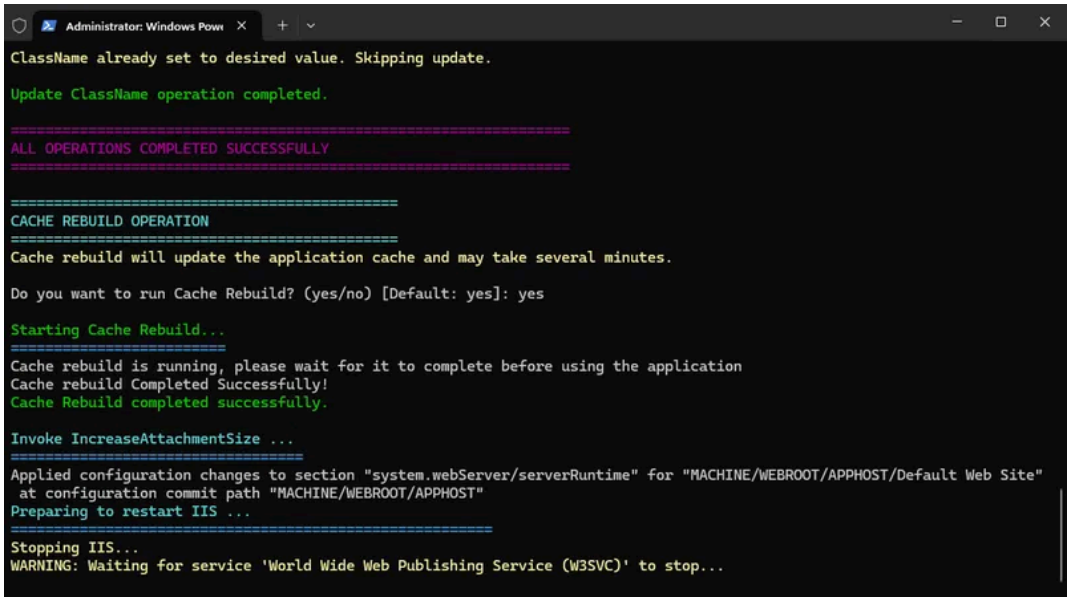
For up to 2025.3 version

1. Click Finish. The System Configuration Wizard closes and the system displays the Completed page of the install wizard.
2. Click Finish in the installer page.
3. Restart the host server that you just configured.
4. Verify that you can access Neurons for ITSM by navigating to http://*server_name*/HEAT.

For 2025.4 and above version

1. Click Finish. If AppServer Module is present in the environment, it launches the Powershell automatically and runs the **install.ps1** script from *C:|Program Files (x86)|HEAT Software|Prerequisite* and prompts you to:
   - **TeamsBot**: Select **Yes** to install TeamsBot or **no** to skip.
   - **Additional Services**: Select **Yes** for **ActionService**, **FRSSurvey**, and **FRSSurveyProxy** or **no** to skip.
   - **CacheRebuild**: Select **Yes** to install or **no** to skip.

   > Cache rebuild is mandatory while installing for the first time. When re-running the SCW you can skip the Cache rebuild.

   Once the **CacheRebuild** completes it restarts the IIS , closes SCW automatically, and returns to the installer.

2. Click Finish in the installer page.
3. Restart the host server that you just configured.
4. Verify that you can access Neurons for ITSM by navigating to http://*server_name*/HEAT.

4.4.3. Configuring the Reporting Feature

If your Neurons for ITSM deployment includes the reporting feature, you need to set up authentication. Authentication is only needed for the reporting feature, and not for deployments that do not use the reporting feature.

- Authenticating the Application Database
- Configuring Microsoft SSRS
- Configuring the Reporting Feature

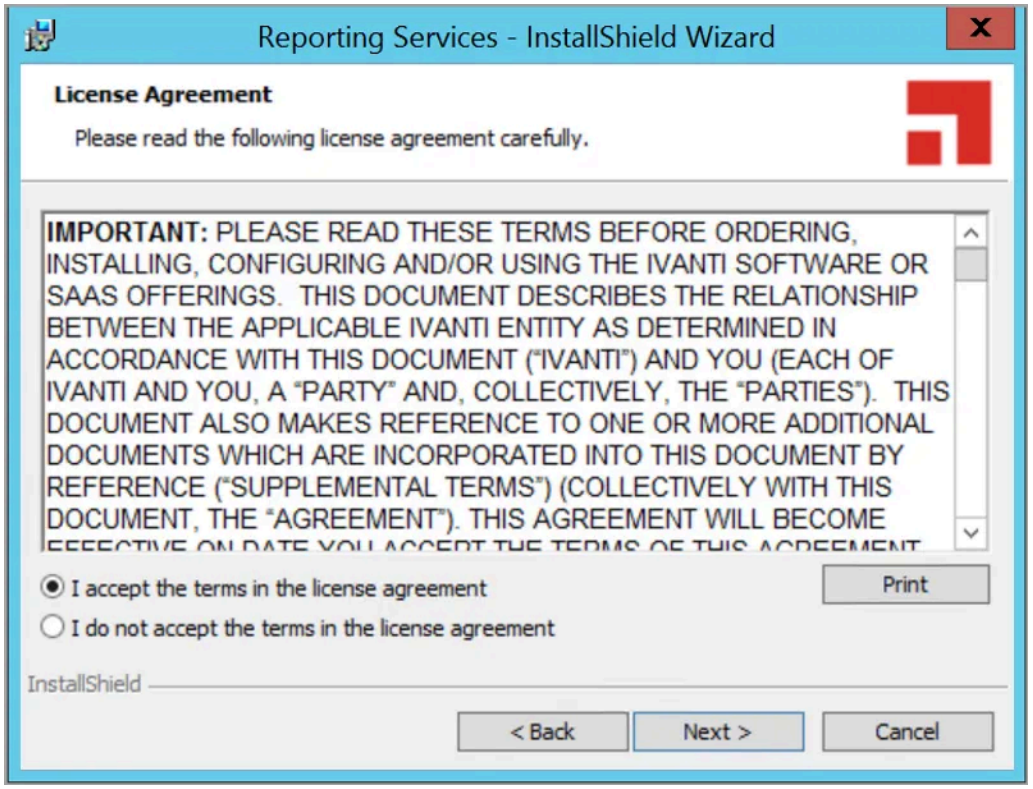
Authenticating the Application Database

The wizard walks you through the first-time set up and configuration as shown in the screenshots below.

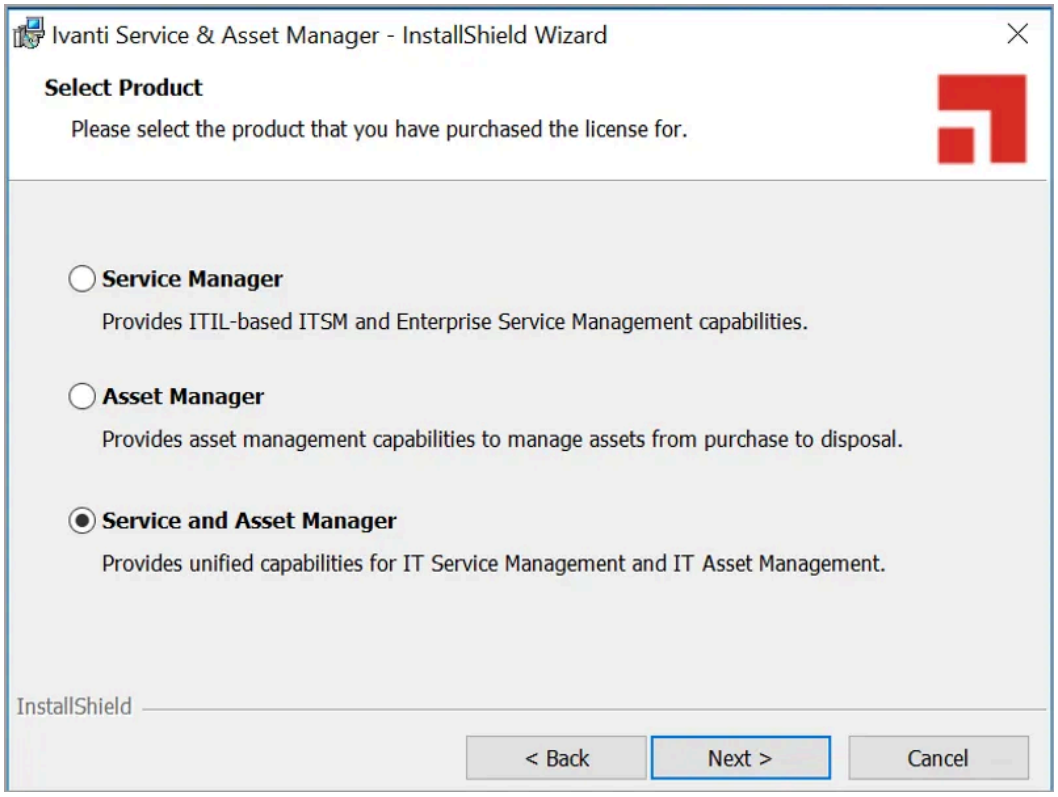
1. Click **Next**.



2. Click **Next** to continue the installation process.

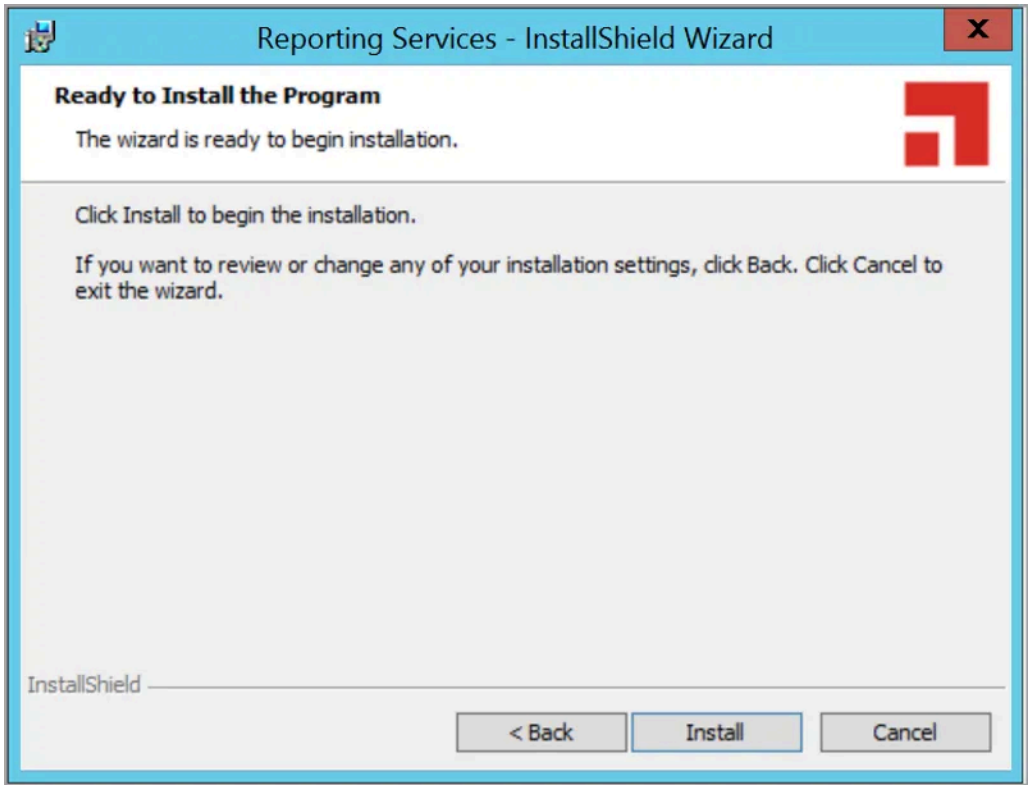


2. Select one of the options, **Service Manager**, **Asset Manager**, or **Service and Asset Manager**.



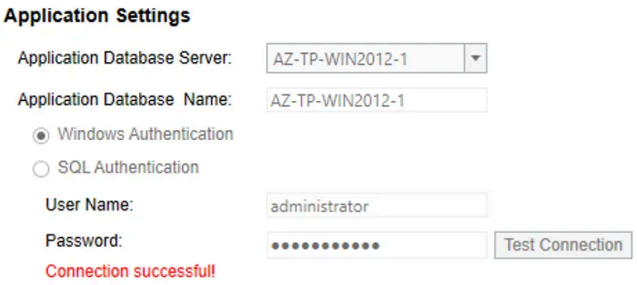
For more information on Asset Manager, see [Ivanti Asset Manager].

2. Click **Next**.



When you configure Neurons for ITSM, you set the authentication for the application database on the Neurons for ITSM **Application** page of the System Configuration Wizard, as shown in Ivanti Service Manager Application Settings.

Ivanti Service Manager Application Settings



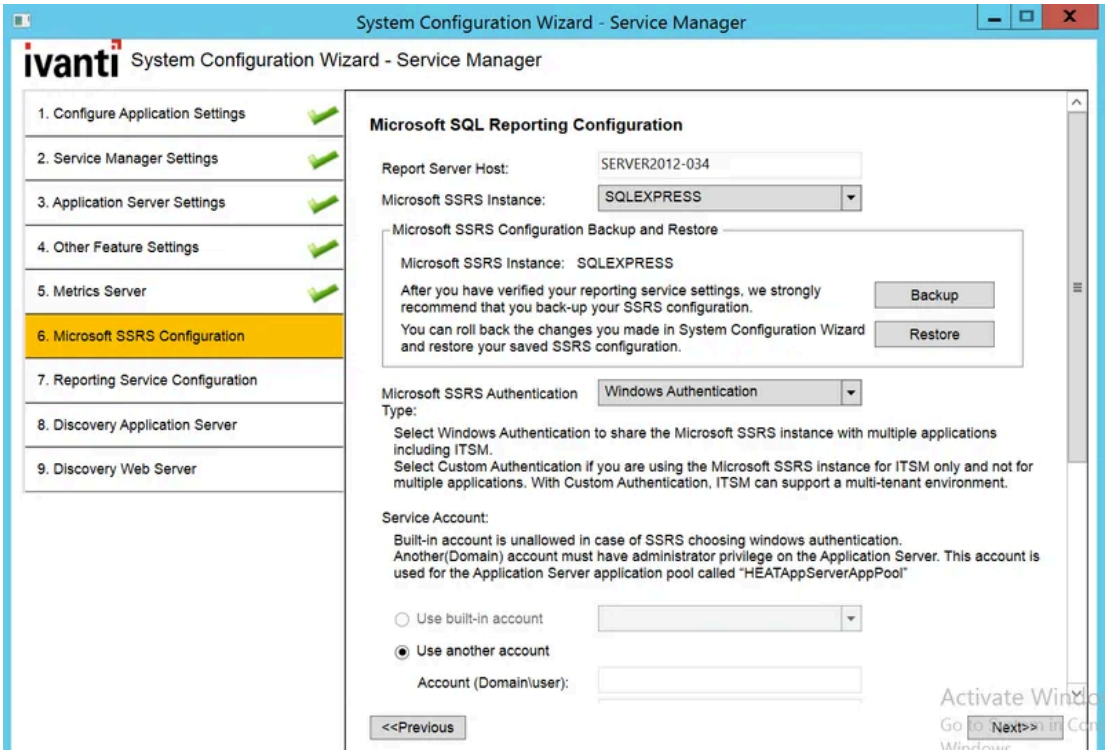
You can select either Windows Authentication or SQL Authentication. (When you configure the reporting feature, you have an opportunity to review and change the authentication method, if necessary, on the **Microsoft SSRS Configuration** page. See Configuring the Reporting Feature.)

- If you select Windows Authentication, you can use the Microsoft SSRS service account, the Microsoft IIS app pool identity account, or any other account, but they must have permission to access the application database and the Microsoft SSRS database.
- If you select SQL Authentication, you can use a different system account.

We recommend having separate authentication for Microsoft SSRS, the application database, and the Microsoft SSRS database.

Configuring Microsoft SSRS

Microsoft SSRS Configuration Page

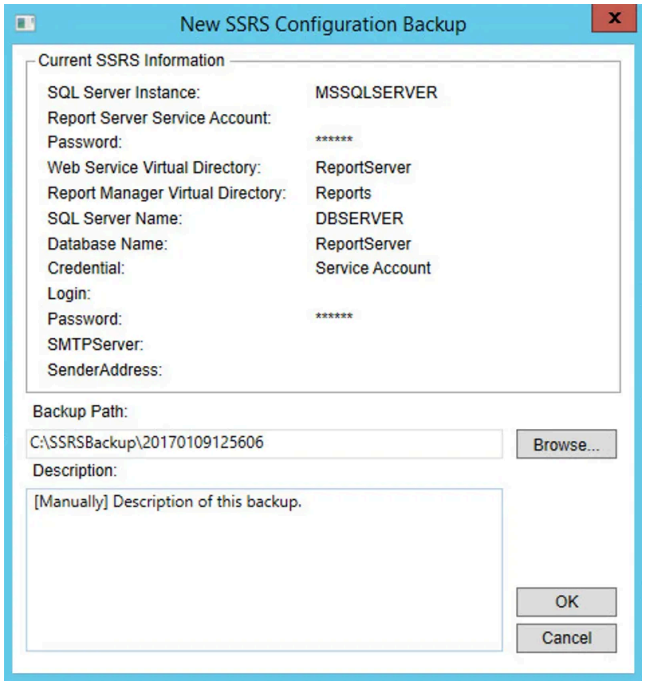


1. Enter values into the fields:

Parameter	Description
Report Server Host	The name of the Neurons for ITSM reporting feature server. The default name is the name of the server that you are currently logged into. You cannot change this value.
Microsoft SSRS Instance	The instance of Microsoft SSRS to use for the Reporting feature. Select a value from the dropdown list.

2. (Recommended) Click Backup to back up your Microsoft SSRS configuration. (If prompted, enter a password to unlock the backup file, and then click OK.) The system displays the **New SSRS Configuration Backup** dialog box with information about the backup.

   New SSRS Configuration Backup Dialog Box



3. To change the location of the backup, click Browse..., select a new location, and click OK.
4. Click OK to close the dialog box. The system displays the **Backup Encryption Key** dialog box. (If you do not see this dialog box immediately, the system displays it after you click Next at the bottom of the page when you are done with the configuration on this page of the System Configuration Wizard.)
5. Do the following:
    - (Optional) Enter a new file location.
    - Enter a password.
    - Confirm the password.
    - Click OK.
6. Select a Microsoft SSRS authentication type:
    - Windows Authentication: Shares the Microsoft SSRS instance with multiple applications, including Neurons for ITSM.
    - Custom Authentication: Does not share the Microsoft SSRS instance with any other applications. The Microsoft SSRS instance is used for Neurons for ITSM only. With custom authentication, you can have a multi-tenant environment.

If you have multiple instances of Microsoft SSRS, choose each instance carefully before choosing the authentication method to use with that instance.

- If you plan to share the same Microsoft SSRS instance with other applications besides the reporting feature, you must use Windows authentication.
- If you plan to create a multiple-tenant environment, you must use custom authentication.
- If you plan to use a dedicated Microsoft SSRS instance only for the reporting feature, you can use either Microsoft SSRS authentication type.

- If you select a Microsoft SSRS instance and select Custom Authentication, the System Configuration Wizard overwrites the existing Microsoft SSRS configuration.
- If the Microsoft SSRS instance that you selected is shared with another application, the Neurons for ITSM installer deletes the configuration for that application. Before proceeding, ensure that the Microsoft SSRS instance is not in use by another application.

7. Enter values into the fields:

Parameter	Description
Service Account	Select the service account to use. This can be the same account that is used for the IIS application pool identity and the Windows service, that you entered on the **Application Server Settings** page. You can select one of the following: • Use built-in account (NOTE: You cannot select this option if you selected Windows Authentication for the Microsoft SSRS authentication type in step 5.) • Use another account
Use built-in account	(Only if you selected Use built-in account) Select the built-in account to use. This can be the same account that is used for the IIS application pool identity and the Windows service, that you entered on the **Application Server Settings** page. You can select one of the following: • Local System: This account has the most permissions. • Network Service: We recommend that you use this account. • Local Service
Account	(Only if you selected Use another account) The name of the service account. Use the format domain/name.
Password	(Only if you selected Use another account) The password associated with the service account.
Web Service Virtual Directory	The name of the web service virtual directory. The system automatically fills in this field with the value that you entered in the **Web Service Virtual Directory** field on the **Application Server Settings** page. See Configuring the Application Server Settings. The default value for the web service virtual directory is ReportServer *+ _SSRS_instance_name*. Do one of the following: • Accept the default value. • Enter a new value. Click Advanced... to configure multiple identities for the reporting feature web service.
Report Manager Virtual Directory	The name of the report manager virtual directory. The default value is Reports *+ _SSRS_instance_name*. Do one of the following: • Accept the default value. • Enter a new value. Click Advanced... to configure multiple identities for the report manager.
Database Server	The name of the database server. The system automatically displays the default Microsoft SQL server name. Do one of the following: • Accept the default value. • Enter the name of the server of the configuration database in this format: machinename\instance. • Browse for a server name by clicking the down arrow and selecting <Browse for more...>. The system displays any Microsoft SQL Server instances in your network. Choose a server and click OK.

Parameter	Description
	NOTES: • Do not choose localhost unless you are setting up a **Demonstration or Proof-of-Concept Deployment**. • To restore the Microsoft SQL server name if you changed the name previously, open Microsoft SQL Server Management Studio. The system displays the Microsoft SQL server name in the **Connect to Server** dialog box.
Database Name	The name of the reporting database. The default name is ReportServer + *$SSRS instance name*.
Credentials	The credentials to use with the reporting feature. Select one of the following: • SQL Server Credentials • Windows Credentials • Service Credentials
User Name	The user name for the credential used with the reporting feature.
Password	The password associated with the credential used with the reporting feature.

> We recommend creating passwords using Alphanumeric characters. Special characters are not permitted.

8. Click Test Connection. The system displays *Connection successful!* if the connection is good.
9. Enter values into the fields:

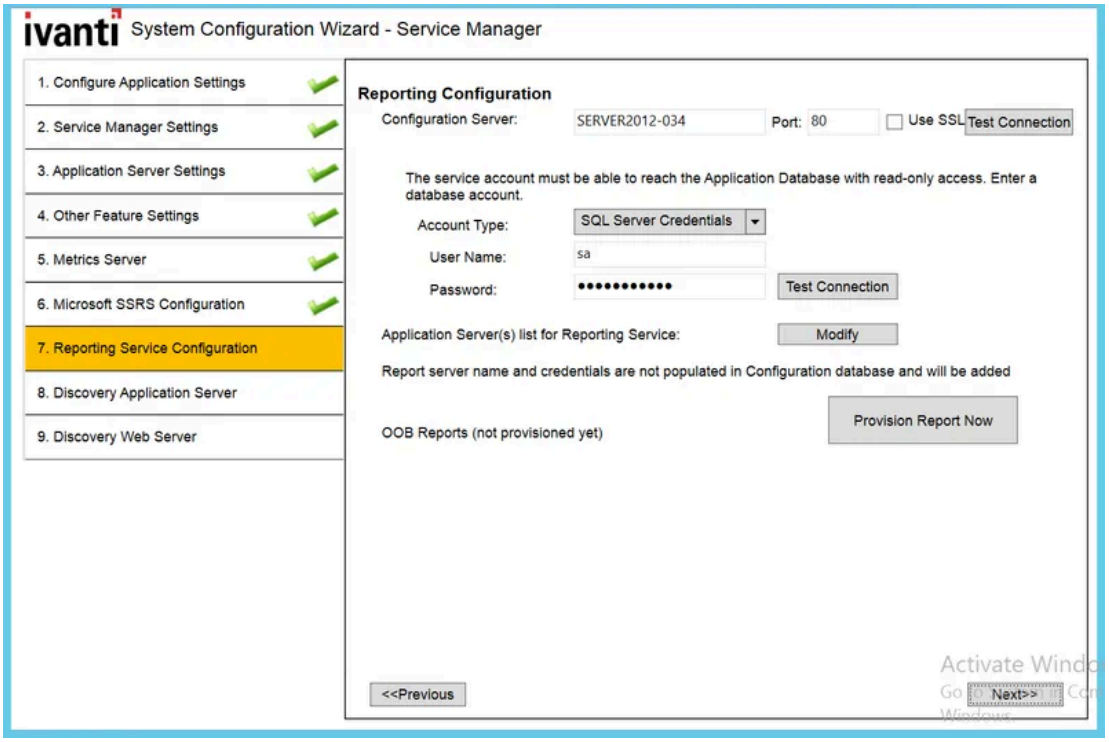
Parameter	Description
SMTP Server	The server that sends out the reports. The system automatically verifies the SMTP server name that you enter. If the name is not valid, the system displays an error message.
Sender Address	The email address for the account that sends out the reports.

10. Click Next. The system displays the **Ivanti Reporting Service Configuration** page. See **Configuring the Reporting Feature**. (If the system displays the Backup Encryption Key dialog box, see step 5.)

Configuring the Reporting Feature

You configure the reporting feature on the **Reporting Service Configuration** page. See **Reporting Service Configuration Page**.

Reporting Service Configuration Page



1. Enter values into the fields:

Parameter	Description
Configuration Server	The host name of the configuration server. The default value is the name of the server that you are logged into now. If the host that you are logged into now is *not* the configuration server, enter the machine name or fully-qualified domain name of your configuration server. If you check Use SSL below, you must enter the fully-qualified domain name of the configuration server.
Port	The port number of the configuration server. The default is 80, or 443 if you check Use SSL.
Use SSL	Check to use SSL for connections to the configuration server. NOTE: We do not recommend enabling SSL on the configuration server until you have fully tested Neurons for ITSM to ensure that it works with SSL. For information on configuring Neurons for ITSM with SSL, see **Optional SSL Configuration**.

2. Click Test Connection to test the connection to the configuration server. The system displays a success or failure message. Click OK to close the message.
3. Enter values into the fields:

Parameter	Description
Account Type	The database account type. Select from one of the following: • SQL Server Credentials • Windows Credentials NOTE: If you selected Custom Authentication in step 6 of **Configuring Microsoft SSRS**, this option automatically displays as SQL Server Credentials and you cannot change it.
User Name	The user name for the Neurons for ITSM database read-only account.
Password	The password associated with the user name.

4. Click Test Connection to test the connection to the configuration server. The system displays *Connection successful!* if the connection is good.
5. Click Modify to add servers to or remove servers from the list of Neurons for ITSM application servers that are used with the reporting feature.

> If you have multiple domains in your deployment, the reporting server may not be able to connect to the web server. In this case, remove the server from the list and configure reporting manually.

6. Enter values into the fields:

Parameter	Description
Report User Name	(Only if you selected Custom Authentication in step 6 of **Configuring Microsoft SSRS**) The user name for the user who configures the reporting feature.
Report User Password	(Only if you selected Custom Authentication in step 6 of **Configuring Microsoft SSRS**) The password associated with the user name.

7. To verify that the configuration settings work, do the following:

   a. Click Reprovision Report Now to update the sample report data to the application. See **Reporting Service Configuration Page**. If the systems returns an exception such as "Cannot Decrypt the Symmetric Key".

   b. Log into Neurons for ITSM as a Report Manager.

   c. Ensure that you can see the sample reports and create one.

8. When you see confirmation that the out-of-the-box reports have been provisioned, that is, the last line of this step changes from *OOB Reports (not provisioned yet)* to *OOB Reports (provisioned on date)*, click Finish to close the System Configuration Wizard.

The system displays the **Completed** page of the System Configuration Wizard.

> If the system displays an error message that the system failed to configure reporting for an Neurons for ITSM application server, ensure that you checked Use these settings for Reporting Service on the **Application Server Settings** page.

9. Restart your system (the host server that you just configured).

10. After the system restarts, ensure that you can access Neurons for ITSM. Go to http://*server_name*/HEAT and ensure that the reporting feature is functional.

4.4.4. Configuring Discovery

This topic describes how to configure Discovery.

- Configuring Discovery on a Dedicated Server
- Configuring Discovery on the Production Servers

Configuring Discovery on a Dedicated Server

After you have installed the Discovery components, the System Configuration Wizard automatically starts and displays the **Configuration Application** page.

1. On the **Configuration Application** page, do the following:

   a. Under **Target Environment**, choose Single Server Deployment.

   b. Under **Setup Connection**, verify that the configuration matches the settings that you configured for the Neurons for ITSM production servers. See Configuring the Configuration Database.

   c. Click Next.

2. On the **Service Manager Application** page, do the following:

   a. Verify that the configuration matches the settings that you configured for the Neurons for ITSM production servers. See Configuring the Neurons for ITSM Application.

   b. Check Discovery Stand alone.

   c. Click Next.

3. On the **Application Server Settings** page, do the following:

   a. Verify that the configuration matches the settings that you configured for the Neurons for ITSM production servers.

   b. Click Next.

4. On the **Other Features Settings** page, do the following:

   a. Under **Customize Server**, in the **Discovery Application Server** field, enter the name of the Discovery dedicated server.

      - If you are not using SSL, enter the host name.
      - If you are using SSL, enter the fully-qualified domain name.

   b. Verify that the rest of the configuration matches the settings that you configured for the Neurons for ITSM production servers. See Configuring Other Feature Settings.

   c. Click Next.

5. On the **Metrics Server** page, do the following:

   a. Verify that the configuration matches the settings that you configured for the Neurons for ITSM production servers. See Configuring the Metrics Server.

   b. Click Next.

6. On the **Discovery Application Server** page, do the following:

a. In the **Discovery Application Server** field, enter the name of the Discovery dedicated server.

- If you are not using SSL, enter the host name.
- If you are using SSL, enter the fully-qualified domain name.

b. Verify that the rest of the configuration matches the settings that you configured for the Neurons for ITSM production servers. See **Configuring the Discovery Application Server**.

c. Click Next.

7. On the **Discovery Web Server** page, do the following:

a. In the **Discovery Application Server** field, enter the name of the Discovery dedicated server.

- If you are not using SSL, enter the host name.
- If you are using SSL, enter the fully-qualified domain name.

b. Verify that the rest of the configuration matches the settings that you configured for the Neurons for ITSM production servers. See **Configuring the Discovery Web Server**.

c. Click Finish.

The system displays the **Completed** page of the System Configuration Wizard.

8. Restart your system (the host server you just configured).
9. After the system restarts, ensure that you can access Neurons for ITSM. Go to http://*server_name*/HEAT and ensure that the system is functional.

Configuring Discovery on the Production Servers

Now that there is a dedicated Discovery server in your deployment, you must configure your production servers to recognize it.

On each production server in your deployment, do the following:

1. Go to the Windows apps menu and click System Configuration Wizard. The system opens the System Configuration Wizard and displays the **Configuration Application** page.
2. On the **Configuration Application** page, click Next.
3. On the Neurons for ITSM application page, click Next.
4. On the **Application Server Settings** page, click Next.
5. On the **Other Features Settings** page, do the following:

a. In the **Discovery Application Server** field, enter the name of the Discovery dedicated server.

- If you are not using SSL, enter the host name.
- If you are using SSL, enter the fully-qualified domain name.

b. Click Next.

6. On the **Metrics Server** page, click Next.
7. On the **Discovery Application Server** page, do the following:

a. In the **Discovery Application Server** field, enter the name of the Discovery dedicated server.

- If you are not using SSL, enter the host name.

- If you are using SSL, enter the fully-qualified domain name.

b. Click Next.

8. On the the **Discovery Web Server** page:

a. In the **Discovery Application Server** field, enter the name of the Discovery dedicated server.

- If you are not using SSL, enter the host name.
- If you are using SSL, enter the fully-qualified domain name.

b. Click Finish.

4.4.5. Configuring the Deployment on the Neurons for ITSM Operations Console

- Configuring the Neurons for ITSM Operations Console
- Creating the Staging and UAT Instances of the Tenants
- Making Configuration Changes to Tenants

Perform these steps from the Operations Console, which you installed in the staging/UAT environment.

> This section only contains brief, high-level instructions.
>
> For complete information about using the Operations Console, including how to log in, see the *Operations Console User Guide for* Neurons for ITSM.

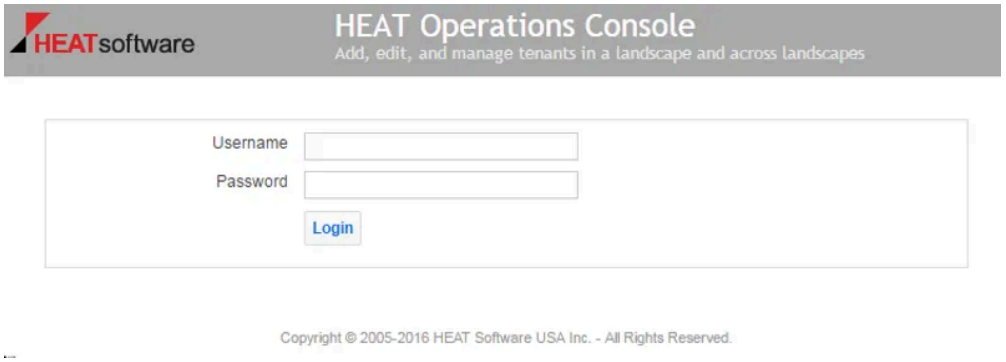
Configuring the Neurons for ITSM Operations Console

To log into the Operations Consoles, do the following:

1. In the **Start** menu, click the down arrow to see the **Apps** menu, and then click HEAT Operations Console.
2. Enter your user name and password and then click Login.

   > The default user name is admin and the default password is manage.

   Operations Console Log In Screen

3. Use the Operations Console to update the values for the Neurons for ITSM application database by doing the following:

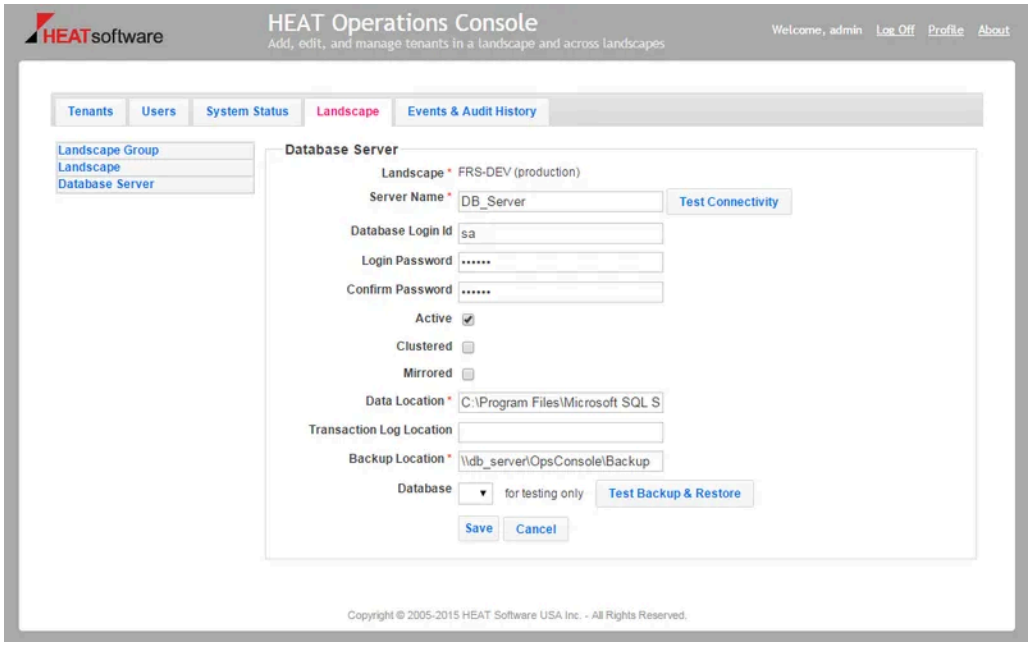
   a. Click the **Landscape** tab.

   b. Click Database Server on the left.

   c. Click Add Database Server.

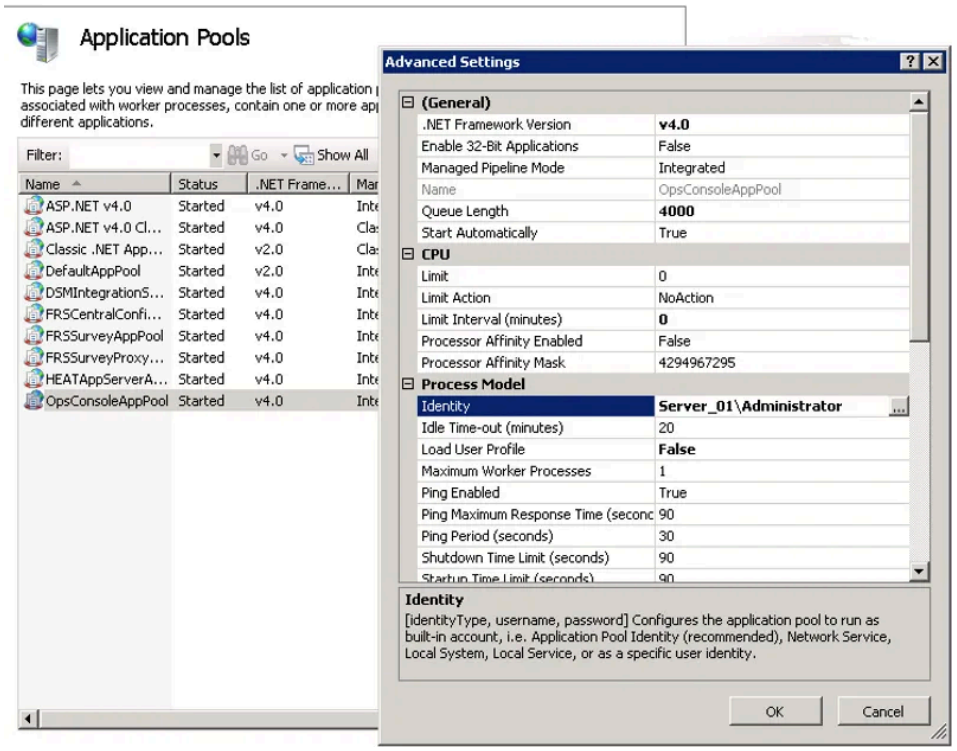
   d. Enter a database login ID and password and set the data and backup locations. See the example values in **Adding a Database Server**.

   Adding a Database Server



e. For the data and backup locations, change the application pool identity so that it has read access. See **Changing the Application Pool Identity**.

Changing the Application Pool Identity



See the *Working with Database Servers* section of the *Operations Console User Guide for* Neurons for ITSM for complete details about editing a database server.

4. Delete the Neurons for ITSM application databases on the staging and UAT instances of the tenant, by doing the following:

   a. Click the **Tenants** tab.

b. Click Expanded View to view all the details about the tenants.

c. Find the staging instance of the tenant and click Deactivate.

d. Click Deactivate at the confirmation prompt.

e. For the staging instance of the tenant, click Delete.

f. Check Skip Backup and click Delete at the confirmation prompt.

g. Repeat steps c. through f. for the UAT instance of the tenant.

You now have three instances of the configuration database and a production instance of the tenant.

See the *Editing a Tenant* section of the *Operations Console User Guide for* Neurons for ITSM for complete details about deleting tenants and tenant instances.

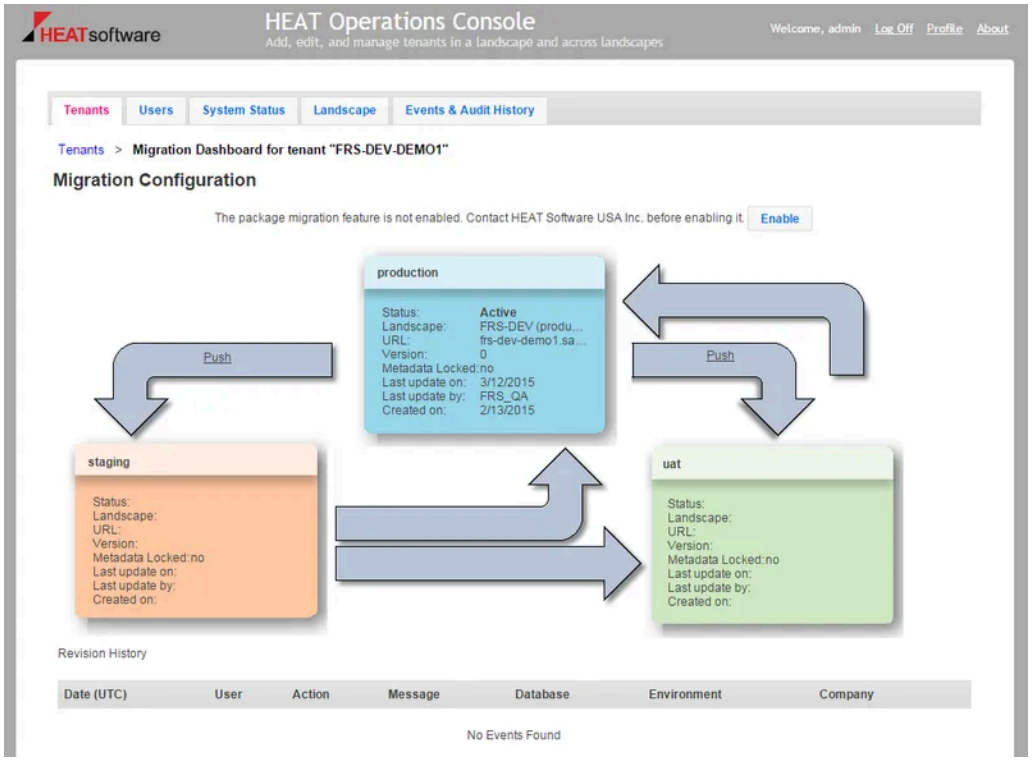
Creating the Staging and UAT Instances of the Tenants

To create the staging and UAT instances of the tenant, based on the production instance, do the following:

1. Log into the Operations Console.
2. Click the **Tenants** tab.
3. Navigate to the production instance of the tenant and click Manage Migration.
4. Create the staging instance of the tenant, by doing the following:

   a. In the migration dashboard, click Push which is located inside the arrow going from the production instance of the tenant to the staging instance of the tenant. See Creating a Staging Instance of the Tenant.

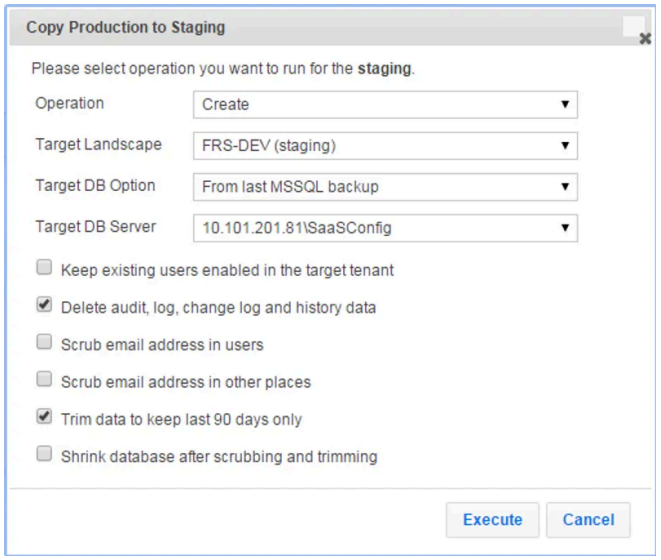
   Creating a Staging Instance of the Tenant



The system displays the **Copy Production to Staging** dialog box.

b. In the **Target DB Option** field, select From live MSSQL backup. See Copy Production Instance of the Tenant to the Staging Instance of the Tenant.

Copy Production Instance of the Tenant to the Staging Instance of the Tenant



c. Click Execute.

5. Create the UAT instance of the tenant:

a. In the migration dashboard, click Push which is located inside the arrow going from the production instance of the tenant to the UAT instance of the tenant. See Creating a Staging Instance of the Tenant. The system displays the **Copy Production to UAT** dialog box.

b. In the **Target DB Option** field, select From live MSSQL backup. See Copy Production Instance of the Tenant to the Staging Instance of the Tenant.

c. Click Execute.

You can now customize Neurons for ITSM and use the Operations Console to push those customizations to the other instances of the tenant.

Making Configuration Changes to Tenants

1. Log into the Operations Console.
2. Click the **Tenants** tab.
3. Make configuration changes in the staging instance of the tenant.
4. Push the configuration changes that you made in the staging instance of the tenant to the UAT instance of the tenant by doing the following:

a. Click Push which is located inside the arrow going from the staging instance of the tenant to the UAT instance of the tenant. See Creating a Staging Instance of the Tenant. The system displays the **Copy Staging to UAT** dialog box.

b. You can use simple mode by leaving Advanced Mode unchecked, or you can use advanced mode by checking Advanced Mode. If you use advanced mode, select Copy Configuration in the **Operation** field and check your options. See Copy

Production Instance of the Tenant to the Staging Instance of the Tenant.

c. Click Execute.

5. Verify that the configuration changes that you made to the staging instance of the tenant (in step 3) were successfully migrated to the UAT instance of the tenant.
6. Make additional configuration changes in the staging instance of the tenant as needed.
7. After you have finished making configuration changes and have verified them, push the configuration changes from the UAT instance of the tenant to the production instance of the tenant.

4.4.6. Optional SSL Configuration

> The Metrics Server is not compatible with SSL if your Neurons for ITSM system has Windows authentication set up.

You can configure Neurons for ITSM for SSL. This configuration is optional. There are three scenarios:

- The Neurons for ITSM application database and the configuration database both use SSL. See **Configuring SSL for the Configuration and Application Databases**.
- The Neurons for ITSM application database uses SSL but the configuration database does not use SSL. See **Configuring SSL for the Application Database Only**.
- Neither the Neurons for ITSM application database nor the configuration database uses SSL. In this scenario, do not check SSL on any of the pages of the System Configuration Wizard.

> For all fields in the System Configuration Wizard that ask for a server location, when using SSL, you must enter a fully-qualified domain name (FQDN). This is because SSL needs a certificate and the certificate authority requires an FQDN.

Configuring SSL for the Configuration and Application Databases

To configure both your Neurons for ITSM application database and the configuration database to use SSL, follow all of the steps in all of these sections:

- **Before You Begin**
- **Configuring SSL in Microsoft IIS Manager**
- **Configuring SSL in the System Configuration Wizard**

Before You Begin

- Ensure that https:\\localhost:443 displays the Microsoft IIS Manager welcome page.
- Ensure that your system has a valid certificate.

Configuring SSL in Microsoft IIS Manager

1. In Microsoft IIS Manager, navigate to Sites > Default Web Site and select SSL Settings.
2. On the **SSL Settings** page, check Require SSL and under client certificates, select Ignore.
3. Navigate to Sites > HEAT and select SSL Settings.
4. On the **SSL Settings** page, check Require SSL and under client certificates, select Ignore.
5. Navigate to Sites > CentralConfig and select SSL Settings.
6. On the **SSL Settings** page, check Require SSL and under client certificates, select Ignore.
7. Navigate to Sites > FRSSurveyProxy and select SSL Settings.
8. On the **SSL Settings** page, check Require SSL and under client certificates, select Ignore.
9. Add an SSL port by doing the following:

   a. Navigate to **Sites > Default Web Site**, right click, and select Edit Bindings....

   b. Click Add....

c. In the **Add Site Binding** dialog box, for the **Type** field, select HTTPS and in the **SSL certificate** field, select the certificate that you received from the certificate authority. The system automatically enters 443 for the port.

d. In the **IP Address** field, enter a fully-qualified domain name.

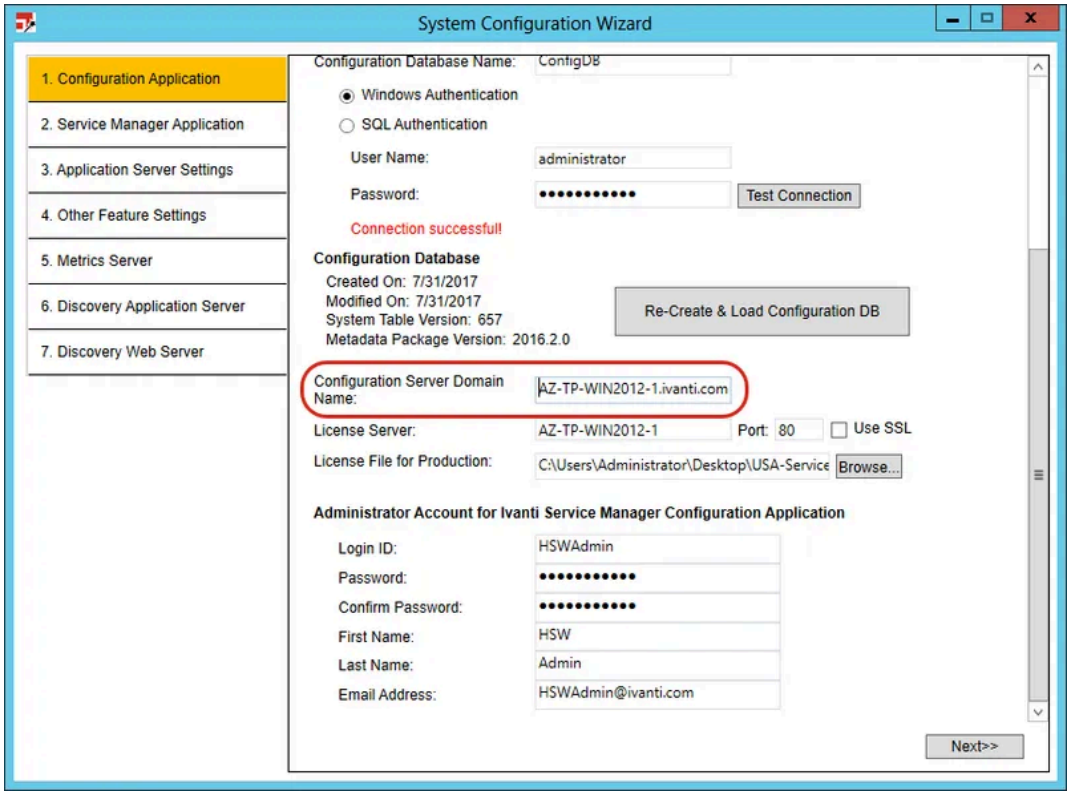
e. Click OK.

10. Verify that you can access https://*fully_qualified_domain_name*.

Configuring SSL in the System Configuration Wizard

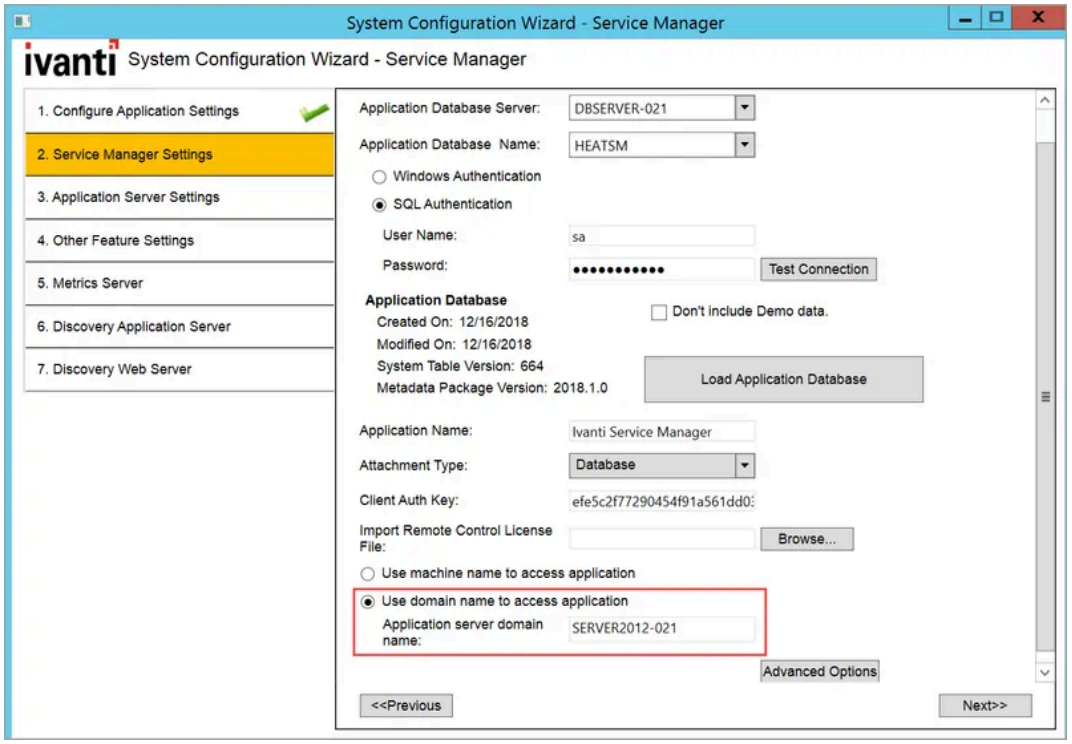
1. In the System Configuration Wizard, on the **Configuration Application** page, ensure that the value in the **Configuration Server Domain Name** field uses a fully-qualified domain name. Do not use a machine name.

   Configuration Application Page



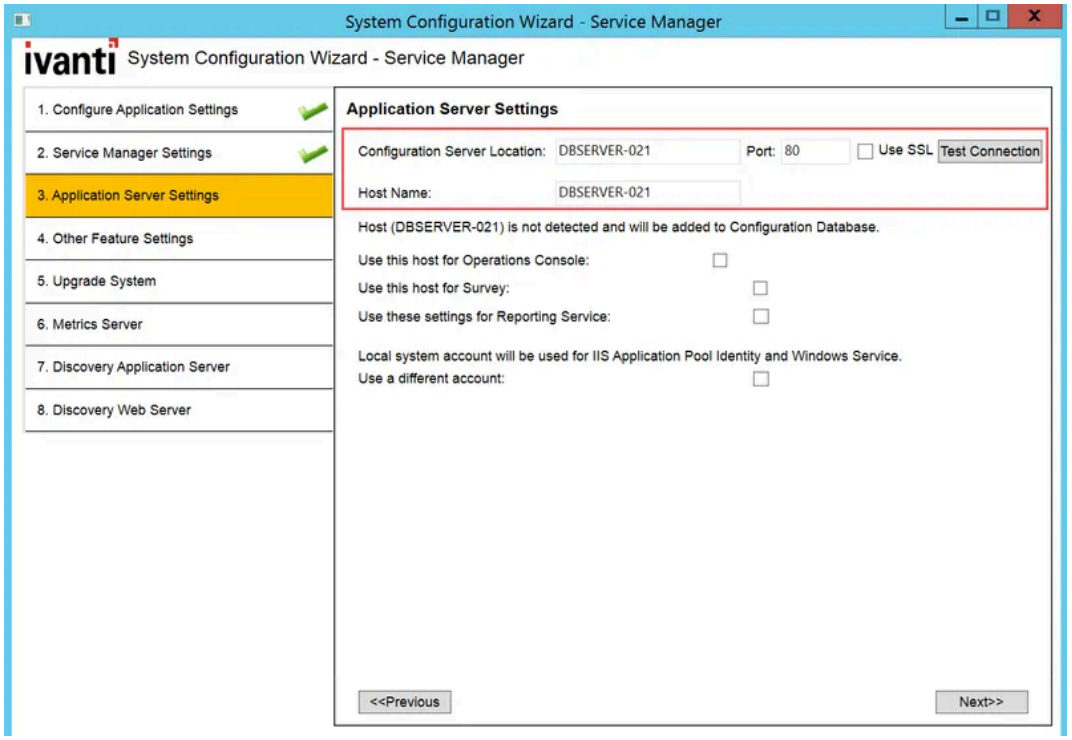
2. On the bottom of the Neurons for ITSM **Application** page, ensure that you check Use domain name to access Application and enter a fully-qualified domain name for the Neurons for ITSM application server. Do not use a machine name.

   Neurons for ITSM *Application Page*



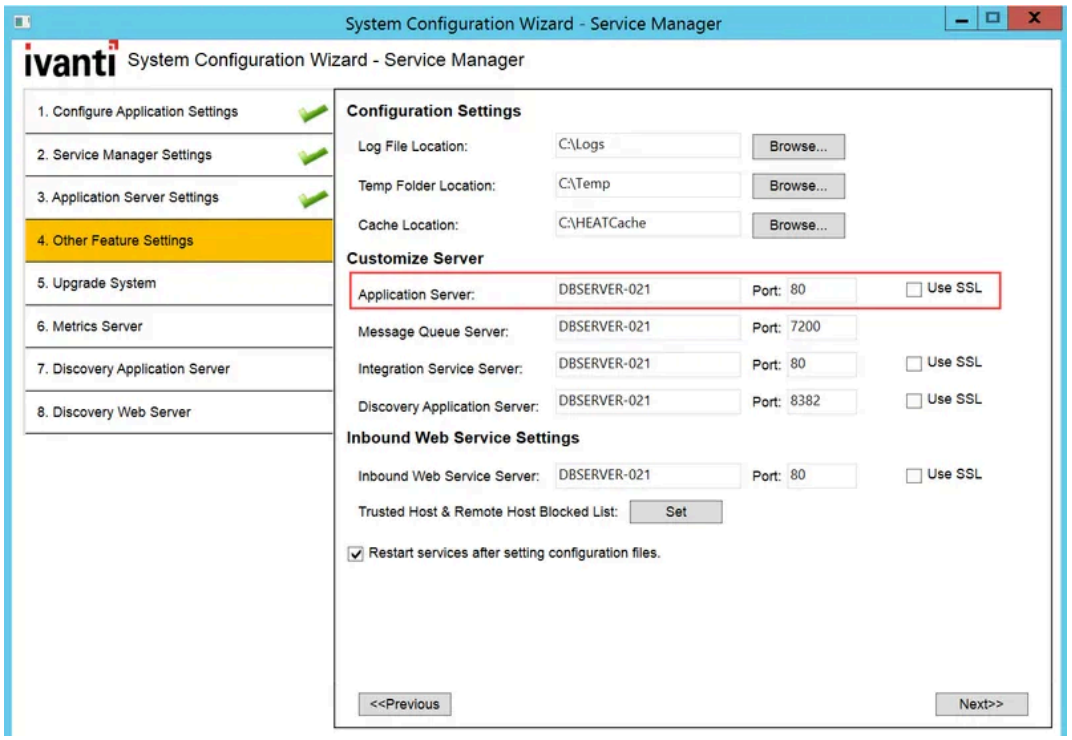
3. On the **Application Server Settings** page, do the following:
   - Ensure that you enter the fully-qualified domain name in the **Configuration Server Location** field.
   - Check Use SSL.
   - Enter the fully-qualified domain name in the **Host Name** field.

Application Server Settings Page.



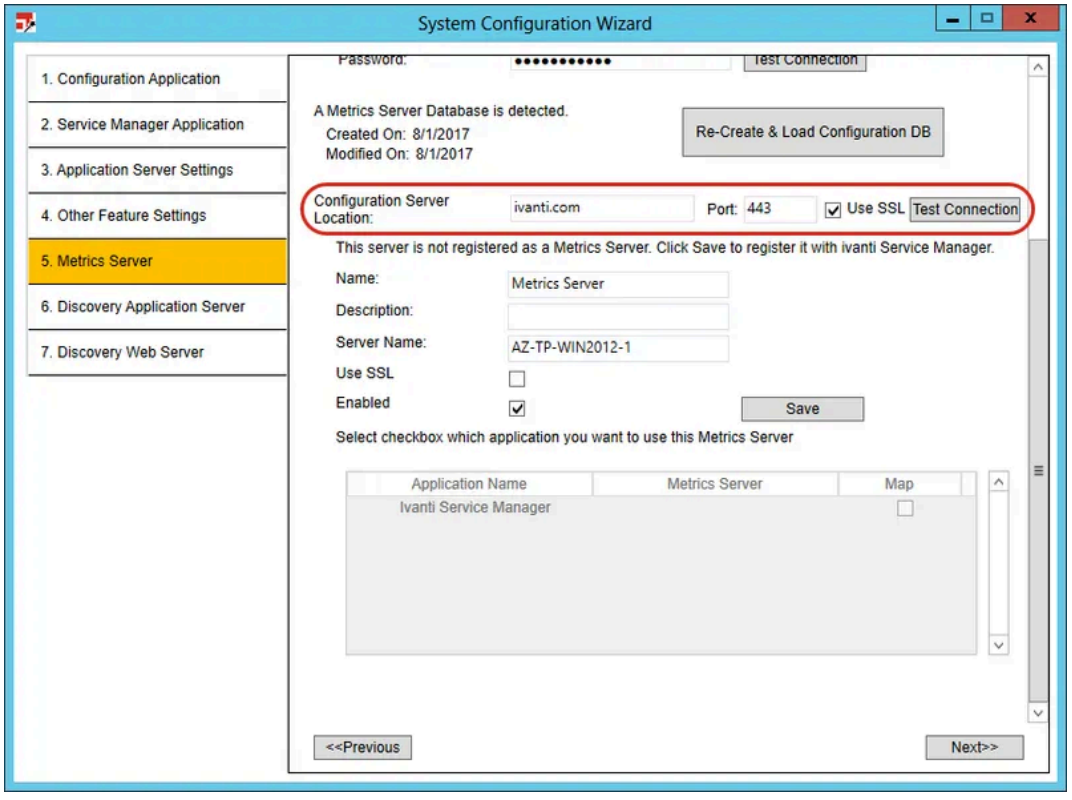
4. On the **Other Feature Settings** page, for the Neurons for ITSM application server, check Use SSL.

Other Feature Settings Page



5. On the **Metrics Server** page, check Use SSL.

Metrics Server Page



Configuring SSL for the Application Database Only

To configure your Neurons for ITSM application database to use SSL but not the configuration database, follow all of the steps in all three sections:

- Before You Begin
- Configuring SSL in Microsoft IIS Manager
- Configuring SSL in the System Configuration Wizard

Before You Begin

- Ensure that https:\\localhost:443 displays the Microsoft IIS Manager welcome page.
- Ensure that your system has a valid certificate.

Configuring SSL in Microsoft IIS Manager

1. In Microsoft IIS Manager, navigate to Sites > Default Web Site and select SSL Settings
2. On the **SSL Settings** page, ensure that Require SSL is not checked. Under client certificates, select Ignore.
3. Navigate to Sites > HEAT and select SSL Settings.
4. On the **SSL Settings** page, check Require SSL and under client certificates, select Ignore.
5. Navigate to Sites > CentralConfig and select SSL Settings.
6. On the **SSL Settings** page, ensure that Require SSL is not checked. Under client certificates, select Ignore.
7. Navigate to Sites > FRSSurveyProxy and select SSL Settings.
8. On the **SSL Settings** page, check Require SSL and under client certificates, select Ignore.
9. Add an **SSL** port by doing the following:

   a. Navigate to **Sites > Default Web Site**, right click, and select Edit Bindings....

   b. Click Add....

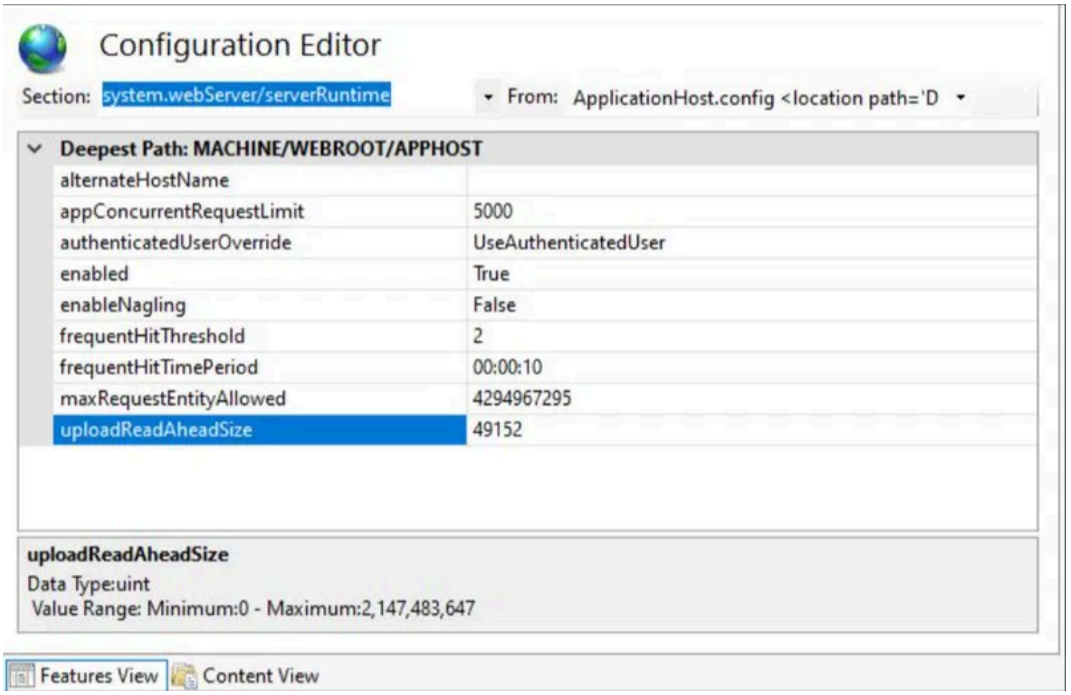
   c. In the **Add Site Binding** dialog box, for the **Type** field, select HTTPS and in the **SSL certificate** field, select the certificate that you received from the certificate authority. The system automatically enters 443 for the port.

   d. In the SSL certificate field, enter a fully-qualified domain name.

   e. Click OK.

10. Double-click **Configuration Editor**.

    The **Configuration Editor** dialog box opens.

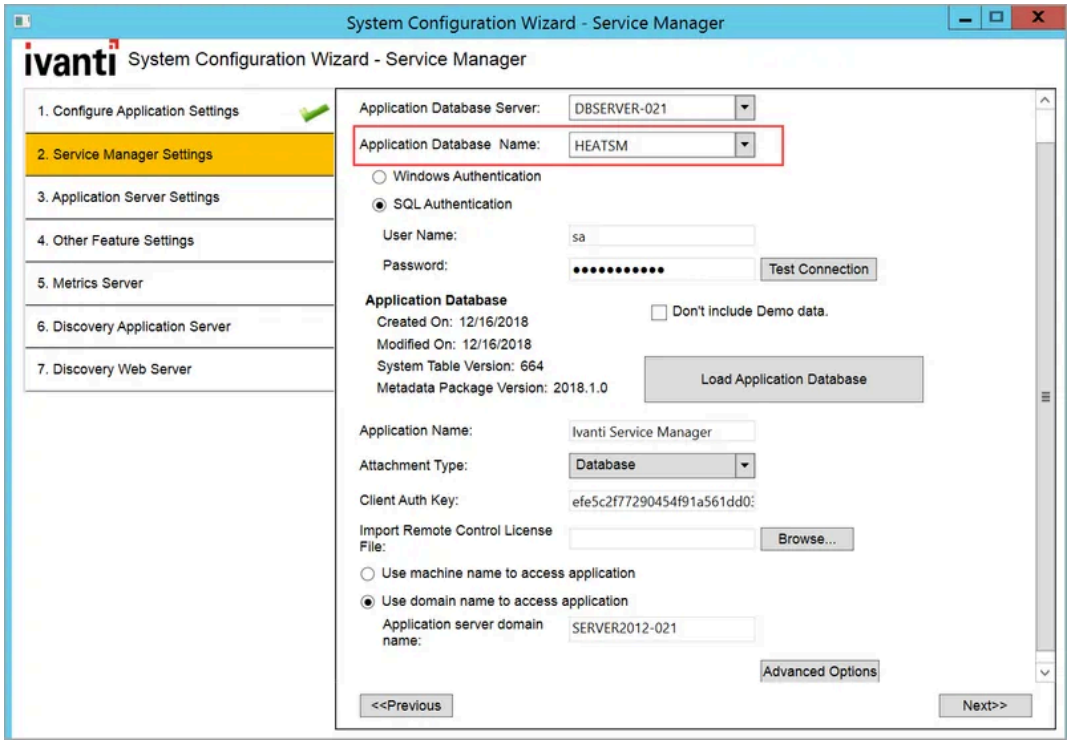


11. Select **System.webServer** and then select **serverRuntime**.
12. Modify the **uploadReadAheadSize** value to `2147483647`.
13. Click **Apply**.
14. Verify if you can access https://*local_host*.

Configuring SSL in the System Configuration Wizard

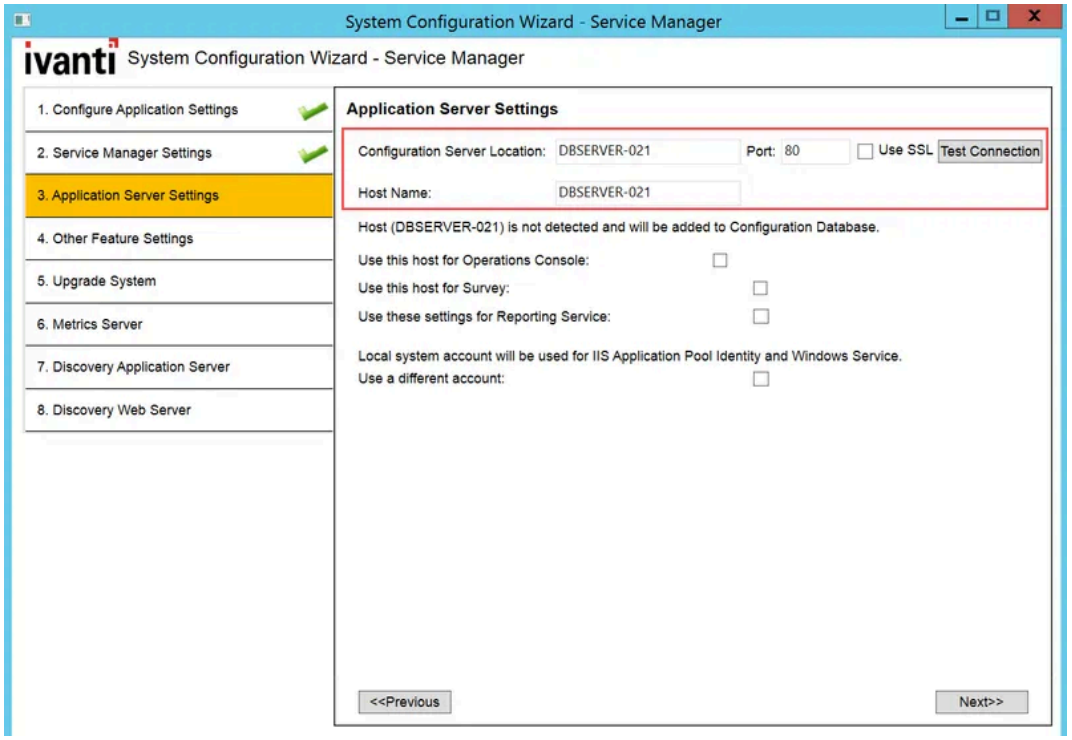
1. In the System Configuration Wizard, on the Neurons for ITSM **Application** page, ensure that the Neurons for ITSM application database uses a fully-qualified domain name and not a machine name.

   Neurons for ITSM *Application Page*



2. On the **Application Server Settings** page, do the following:
   - Ensure that Use SSL is not checked.
   - Ensure that you enter the fully-qualified domain name in the **Configuration Server Location** field.
   - Ensure that you enter the host name, and not the fully-qualified domain name, in the **Host Name** field.

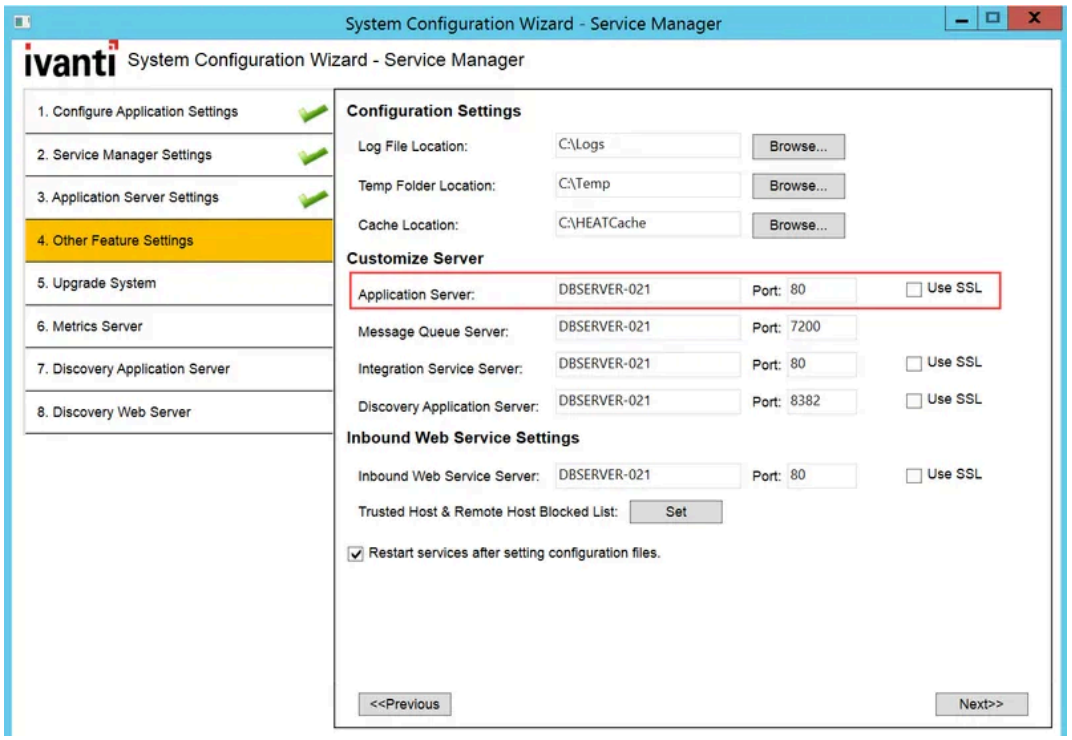
Application Server Settings Page



3. On the **Other Feature Settings** page, check Use SSL.

> This is the *only* place in the System Configuration Wizard where you check Use SSL. Do *not* check Use SSL on any other page.

Other Feature Settings Page



Configuring SSL for Integration Service

Ivanti Integration Service by default configured to work on non-SSL port irrespective of the “SSL” check box is selected in System Config Wizard (SCW).

<That SCW checkbox tells other parts of the product that the integration service is SSL (or not), so it needs to be checked, and the manual edit of the JSON file is also needed.>

1. To make Integration Service work on SSL port, following steps to be performed on each server where Ivanti Integration Service is running.
2. Stop the Ivanti integration service.
3. Add the following entries at the end of appsettings.json of Integration Service.

   The Default Path for Integration Service is “C:\Program Files\HEAT Software\HEAT\IntegrationServer\appsettings.json”

```Bash
"Kestrel": {
   "Endpoints": {
    "HttpsInlineCertFile": {
       "Url": "https://<IntegrationURL>:1213",
            "Certificate": {
          "Subject": "<SystemCertificate>",
          "Location": "LocalMachine",
          "AllowInvalid": true
        }
       }
      }
     }
```

> The **IntegrationURL** should be replaced with the actual integration URL configured in SCW.
> And a trusted certificate is available in your local Computer > Personal certificate store.<Ensure the value in the **appsettingss.json** file is the "Issued To" string.>

Ivanti recommends that there is only one If there is more than one cert here with the same Name, the OS will choose one so it is best practice to ensure only one cert is present with the same name as configured in the json file

> **SystemCertificate** should be replaced with the installed certificate name on server machine where the Integration Service is running.

4. Start the Integration Service.
5. Verify that https://<IntegrationURL>:1213/IntegrationService.svc is the correct URL.
6. Open the Web.Config of HEAT application located in “C:\Program Files\HEAT Software\HEAT\AppServer” and update the IntegrationURL with right value for the following binding.

   <endpoint address="https://< IntegrationURL >:1214/IntegrationService.svc"

   binding="wsHttpBinding" bindingConfiguration="wsLargeDataBinding"

   contract="IntegrationService.IIntegrationService" name="IntegrationServiceEndpoint">

7. Open the appsettings.json of HEAT application located in “C:\Program Files\HEAT Software\HEAT\AppServer” and update the IntegrationURL with the right value for the following binding:

   "IntegrationServiceUri": https://< IntegrationURL >:1214

8. Reset the IIS.

4.4.7. Optional LDAP Configuration

Neurons for ITSM supports LDAP. LDAP is configured from the Configuration Console.

After installation, go to the online help and view the topic called *Configuring LDAP Settings*.

4.4.8. About Configuring with ADFS

Neurons for ITSM supports ADFS. ADFS is configured from the Configuration console.

After installing the system, go to the online help and view the topic called *Working with ADFS/SAML*.

4.4.9. About Configuring Throttling Settings

- About Throttling
- Checking the Throttling Status
- Configuring Outbound Throttling Settings

About Throttling

In Neurons for ITSM, throttling is when the system intentionally slows the speed of web services. We use this process to regulate the network traffic and to minimize bandwidth congestion. When using web services, Neurons for ITSM may limit the message rate, based on overall load. If you exceed the rate of 100,000 API calls per day, you may find your responses slowed to an appropriate rate. This helps ensure that you get a fair share of resources, and Ivanti Software can prevent unintended performance degradation from runaway integrations and denial of service attempts.

For more information about using the Neurons for ITSM integration web services, go to the online help and view the topic called *Working with the Integration Web Service*.

Checking the Throttling Status

Configuring Outbound Throttling Settings

In on-premise implementations, your web service use only affects your environment. If you exceed the recommended rate, you may see performance implications. You can adjust your outbound throttling thresholds to maximize performance. The hardware capacity of your system also affects the values that you set these parameters to.

The following are the outbound throttling parameters:

Parameter	Description
ResetRuntimeCache ExpectedTimeout	The amount of time, in ms, that the system allocates to reset the runtime cache operation. If it takes longer than this, the system moves the operation to the pending actions and marks the server as unhealthy.
ResetRuntimeCache UnhealthyDelayTime	If the server is considered unhealthy, this is the amount of time, in ms, to delay things. After this amount of time has passed, the system executes the action.
PendingActionClean upPeriod	How often, in ms, to clean up the pending actions that are stored in memory.
PendingActionsThres hold	The amount of memory, in bytes, used to store all of the pending actions. Whenever a pending action exceeds the timeout value, the system adds the action to the pending action collection in memory.
ThrottlingEnable	Turns on or off outbound throttling.

Change the throttling parameters as follows:

1. Navigate to the AppServer folder and open the web.config file using a text editor such as Notepad.
2. Search for and find the following strings:

Bash

```
<!--Expected timeout on executing reset runtime cache action
during throttling (value in milliseconds)-->
<add key="ResetRuntimeCacheExpectedTimeout" value="1000" />

<!--Expected delaying on executing reset runtime cache before
executing action in during throttling (value in milliseconds)-->
<add key="ResetRuntimeCacheUnhealthyDelayTime" value="5000" />

<!--Expected period to clean up the pending actions (value
in milliseconds)-->
<add key="PendingActionCleanupPeriod" value="300000" />

<!--Expected size threshold of pending actions collection -->
<add key="PendingActionsThreshold" value="15728640" />

<add key="ThrottlingEnable" value="true" />
```

3. Change the values as needed.
4. Save the file.
5. Perform a Microsoft IIS reset.

4.5. Setting up Redis

You will have to setup Redis if you want to use the Chat service in Neurons for ITSM. All the different setup scenarios are explained below.

Different install and uninstall options and scenarios

Upgrading/Installing Redis for existing/New customer:

- Installing Redis in Linux Container on Windows
  - Installing Linux container on Windows Server with Docker
  - Installing Redis 6.2.5 in Docker
- Installing Redis on Linux Machine
  - Installing Redis on Linux Machine (Ubuntu)
  - Establish the remote Redis Connection
  - Install Redis on Windows Using WSL2
- Uninstalling Redis from Windows Server (existing customer)
  - Remote Redis Connection - using redis.bat script
  - Remote Redis Connection from Windows UI
  - Remote Redis Connection using redis-server.exe
- Uninstalling from Linux Container
- Uninstall from Linux Machine

Option 1 – Install Redis in Linux Container on Windows Server

Prerequisite:

- Windows server - version 2019 or above with HyperV enabled.
- For more information on Linux container requirements check out - **Windows container requirements**.
- For upgrading your Windows Servers, check out this **Windows Server Upgrade guide** and look for target version you need to upgrade.

Installation steps:

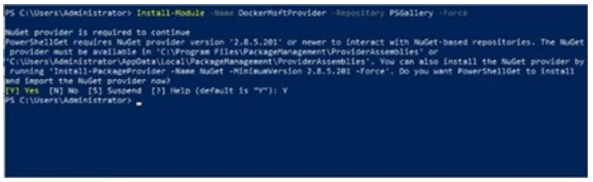
Installing Linux container with docker on Windows Server

1. Open PowerShell as Administrator and run the following command:

```
Install-Module -Name DockerMsftProvider -Repository PSGallery -Force
```

   This installs the Docker-Microsoft PackageManagement provider.

   Sample Output



2. Install the preview build of Docker EE and run the following commands:

Install-Module DockerProvider

Install-Package Docker -ProviderName DockerProvider -RequiredVersion preview

3. Use “Yes” or “Y” to agree to install or use “A” to agree to all.
4. When the installation is complete, reboot the system.
5. To run Linux containers, enable LinuxKit system and use the following command:

[Environment]::SetEnvironmentVariable("LCOW_SUPPORTED", "1", "Machine")

6. Use the following command to restart the system/Docker service after the change:

Restart-Service docker

   The Linux container should now successfully run on your Windows Server.

For more information, refer - **How to run Docker Containers on Windows Server 2019**

Installing Redis 6.2.5 in Docker

1. Open PowerShell as Administrator and run the following command:

docker run -d -p 6379:6379 --name redis-con redis:6.2.5

2. Check if it has successfully installed by executing the command - `docker ps`

Best Practices

- Run the command - `docker update --restart always redis-con` to always restart Redis Container whenever the docker/machine is restarted. If this is not enabled you have to enable Redis container manually everytime the docker/machine is restarted.
- Enable Authentication for Redis. Refer, Add Authentication to Redis.

Setting up Redis Authentication

Add Auth to Redis server

1. Open Powershell and run the command - `docker exec -it redis-con sh`.

   This will give you Redis console.

2. Run **Redis-cli** to set password to Redis server.
3. Run this command to set Redis password - `redis-cli Config Set requirepass <your-password>`.

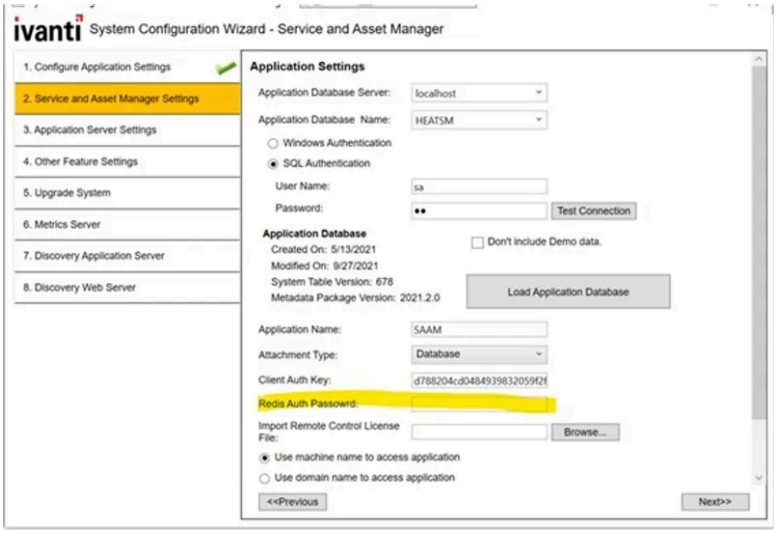
   Your password is successfully set now.

4. To test run the command - `redis-cli Auth <your-password>” it should return “Ok`.

Set the password through SCW for the application to authenticate through Redis password.

In step 2 of SCW configuration (Service and Asset Manager settings) in the field **Redis Auth Password**, enter the same password that is set in the **redis-cli**, and save it.

> Once docker/Redis server/machine is restarted , the auth will be set back to noAuth, sensure to remember and update the password to redis server every time restarted.



Option 2 – Install Redis on Linux Machine

Prerequisite:

- Ubuntu 16.04 or above

Steps to install:

1. Execute the following commands:

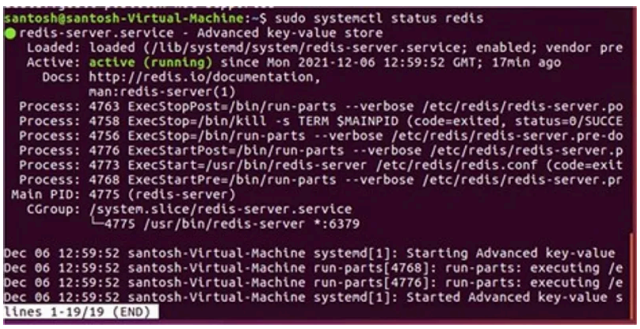
Sudo apt update

Sudo apt install redis-server

2. Check if Redis is set up using the following command:

sudo systemctl status redis

If the result displays as `Active: active (running)` then Redis is successfully setup.



3. Edit the **redis.conf** file to make Redis port available to other servers (by default, the Redis port is on :6379) "*sudo gedit /etc/redis/redis.conf*" or "*sudo nano /etc/redis/redis.conf*"
4. Check for the line "**bind 127.0.0.1**" comment this line by prepending '**#**'

   Since you have removed binding, it is highly recommended to add authentication to Redis Server.
5. Edit **redi.conf "sudo gedit /etc/redis/redis.conf"**
6. Search for requirepass foobared, replace foobared with your_passowrd.
7. Restart the Redis Server to apply changes.

   "sudo systemctl restart redis.service"
8. Then follow "Remote Redis Connection" and "Set password through SCW for the application to authenticate through Redis password."

Option 3 – Install Redis on Windows Using WSL2

This procedure describes how to install Redis on Windows by using Windows Subsystem for Linux (WSL). Redis is installed inside a Linux environment, which runs on Windows through WSL. This method is supported on Windows 10, Windows 11, and Windows Server 2019 or later.

To install Redis on Windows, enable WSL2 (Windows Subsystem for Linux). WSL2 lets you run Linux binaries natively on Windows.

Prerequisites:

- Windows 10, Windows 11, or Windows Server 2019/2022.

Steps to enable WSL:

For Windows 10, Windows 11, or Windows Server 2022.

1. Open **PowerShell** or **Command Prompt** as an administrator.
2. Run the following command:
   wsl --install
3. Restart the system when prompted.

> The **wsl --install** command installs the WSL feature and sets Ubuntu as the default Linux distribution.

For Windows Server 2019 (version 1709 and later)

1. Open **PowerShell** as an administrator.
2. Run the following command:
   Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Windows-Subsystem-Linux
3. Restart your computer when prompted.
4. Download a Linux distribution from the Microsoft WSL Store: *"https://aka.ms/wslstore"*
5. Download the latest **ubuntu** from the following link: *"Manual installation steps for older versions of WSL | Microsoft Learn"*
6. Open PowerShell, Install the downloaded distribution using the following command::
   curl.exe -L -o ubuntu-2004.appx https://aka.ms/wslubuntu2004
7. Replace **app_name.appx** with the actual file name of the downloaded package.

1. Add-AppxPackage .\ ubuntu-2004.appx

8. This installs Linux on your computer.

Steps to Install Redis:

1. Open the installed Linux distribution (for example, Ubuntu) from the Start menu.
2. Update the package list and install Redis:
   curl -fsSL https://packages.redis.io/gpg | sudo gpg --dearmor -o /usr/share/keyrings/redis-archive-keyring.gpg
   echo "deb [signed-by=/usr/share/keyrings/redis-archive-keyring.gpg] https://packages.redis.io/deb $(lsb_release -cs) main" | sudo tee /etc/apt/sources.list.d/redis.list
   sudo apt-get update
   sudo apt-get install redis
3. Start the Redis server:
   sudo service redis-server start
4. Verify that Redis is running:
   redis-cli
   ping
5. If Redis is running correctly, the response will be:
   PONG

Establish the remote Redis Connection

1. Edit the **.env** file in the path "*<install-Path>|Heat|Appserver|chat|.env*"
2. Edit the attribute **REDIS_HOST=localhost to REDIS_HOST=<remote-server-ip>**
3. Save the file.
4. This will install redis successfully.
5. Check if Redis is up by using the command `sudo systemctl status redis" – you should see the active (running status).`

Uninstall Redis for Windows Server (Existing Customer)

This is a mandatory step for existing customers who have already installed Redis and are upgrading to the new version. Uninstall the obsolete Redis version from your machine by following the below server details and uninstall steps.

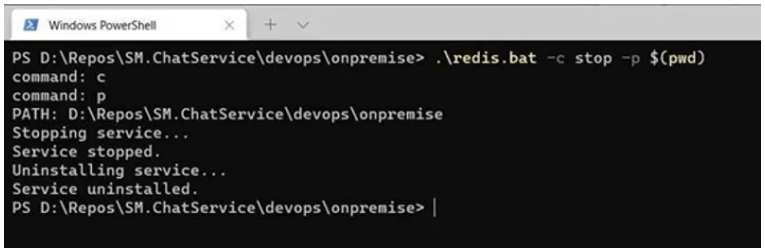
Redis server details:

Version	3.0.503
Path	<install_path>\HEAT\Appserver\AppServer.Main\redis
Project	https://github.com/microsoftarchive/redis
License	https://github.com/microsoftarchive/redis/blob/3.0/license.txt

Option 1 – Uninstall using redis.bat script

1. Run the following in Powershell with administrator privileges.

cd <install_path>\HEAT\Appserver\AppServer.Main\redis; .\redis.bat -p $(pwd) -c stop



2. Optional Step: Cleanup Redis installation files by using the below command:

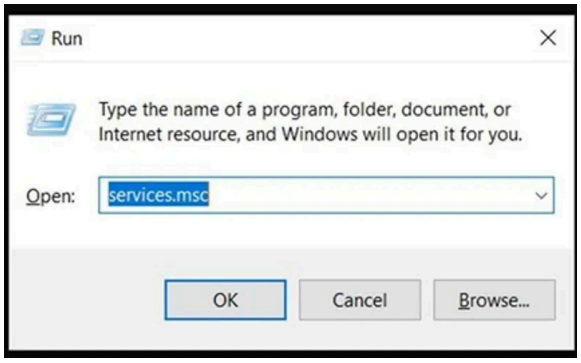
.\redis.bat -p $(pwd) -c cleanup



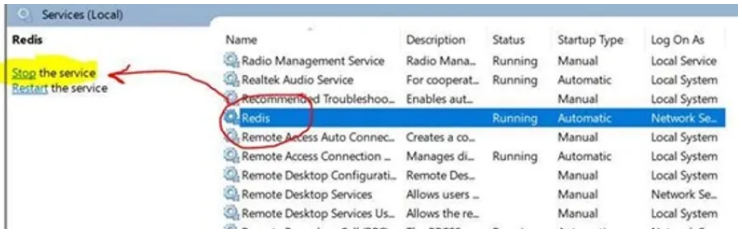
Option 2 – Uninstall from Windows UI

1. Run the following in Powershell with administrator privileges:

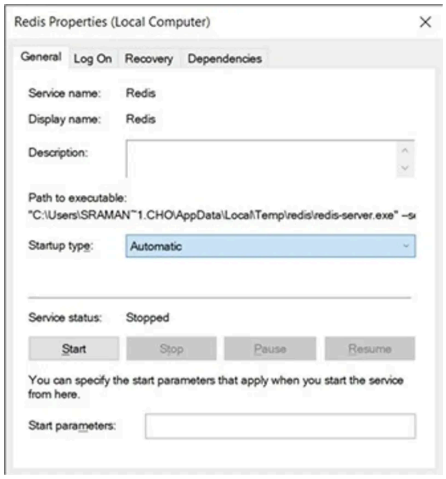
Run services.msc



2. Locate Redis in the list and stop the service.



3. Right click > **Properties**, Change **Startup type** to **Disabled**.



Option 3 – Uninstall using redis-server.exe

Stop the service

Run the following in Powershell with administrator privileges.

.\redis-server.exe --service-stop

```
PS D:\Repos\SM.ChatService\devops\onpremise\redis> .\redis-server.exe --service-stop
[16956] 15 Sep 16:49:03.174 # Redis service successfully stopped.
PS D:\Repos\SM.ChatService\devops\onpremise\redis>
```

Uninstall service

Run the following in Powershell with administrator privileges.

.\redis-server.exe --service-uninstall redis.windows.conf

Uninstall Redis

If you do not want to use the Chat service, you may uninstall Redis by the following the below steps:

Option 1 – Uninstall Redis in container

Execute the following command:

Docker rm /redis-con

Option 2 – Uninstall Redis in Linux Machine

For ubuntu distributions

if you use apt-get to install redis then use

sudo apt-get purge --auto-remove redis-server

if you compiled redis manually then follow the

steps below to remove it completely from linux/ubuntu

sudo service redis_version stop

Now delete everything related to Redis server from /usr/local/bin/

```
sudo rm /usr/local/bin/redis-*

# Now delete Redis Configuration files directory and it's content.

sudo rm -r /etc/redis/

# Delete existing Redis log files.

sudo rm /var/log/redis_*

# Delete existing Redis data directory and it's content.

sudo rm -r /var/lib/redis/

# Delete existing Redis server init scripts

sudo rm /etc/init.d/redis_*

# Remove existing Redis PID files (Only if exists)

sudo rm /var/run/redis_*
```

4.6. Logging into Neurons for ITSM

- Logging In Using the Standard Login Dialog Box
- Logging In Using the Standard Login Dialog Box with an Application Menu
- Adding the Application Menu Option to the Standard Login Dialog Box

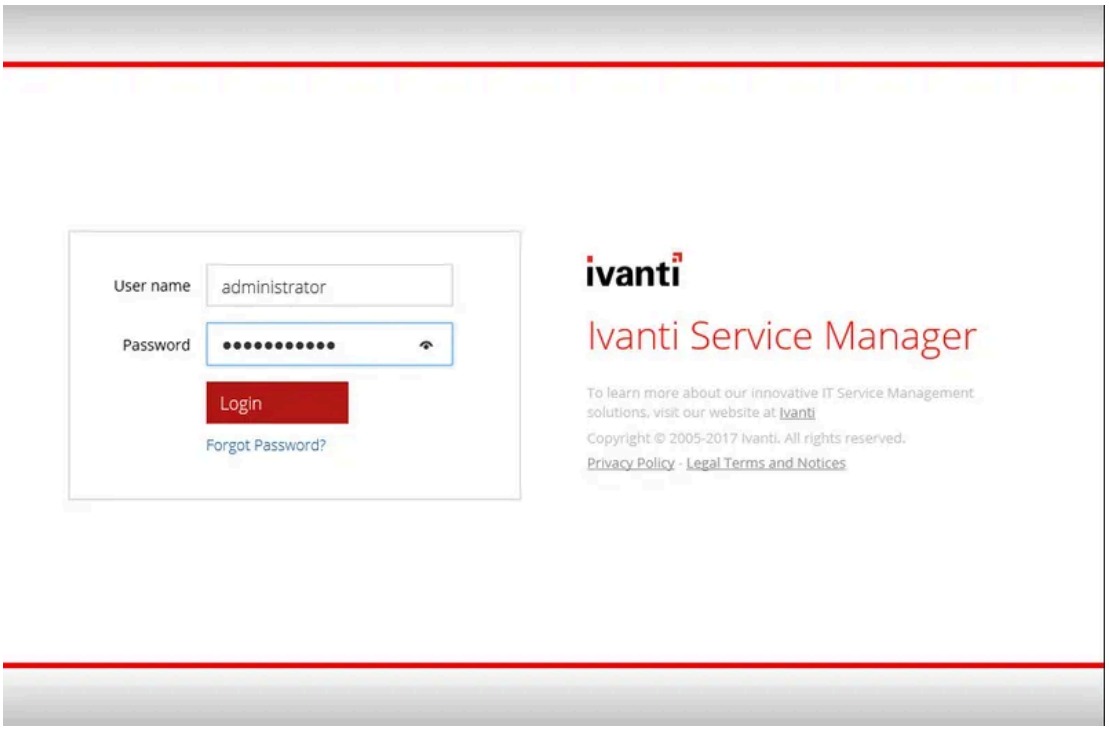
Logging In Using the Standard Login Dialog Box

The standard login dialog box enables you to access Neurons for ITSM on the host that you are logged into now. You see this dialog box when you click the any of the following:

- The Neurons for ITSM desktop shortcut
- The Windows apps menu shortcut

In the dialog box, enter your user name and password, and then click Login.

Standard Login Dialog Box



> The default username for Neurons for ITSM is HEATAdmin. You created the password when you configured the application database.

Logging In Using the Standard Login Dialog Box with an Application Menu

This login dialog box enables you to access the following:

- The Neurons for ITSM system on the host that you are logged into now.
- Individual tenants, such as staging or UAT, if you have more than one.
- The configuration database (ConfigDB).

In the dialog box, enter your user name and password, select an application, and click Login.

- The default username for the configuration database (ConfigDB) is HSWAdmin. You can choose the password when you create the database.
- The default username for Neurons for ITSM is HEATAdmin. You can choose the password when you created the application database.

Adding the Application Menu Option to the Standard Login Dialog Box

To see the login dialog box with the application menu option (only for server-side login), do the following:

1. Go to C:\Program Files\HEAT Software\HEAT\AppServer\.
2. Right-click the HEAT.url file, and choose Properties.
3. Change the host name and port number to localhost. For example, change http://SERVER2012:80/HEAT to http://localhost/HEAT.
4. Click OK.
5. Open a directory window or a browser window and enter either localhost/HEAT or http://localhost/HEAT.

4.7. Adding Features and Changing Settings

- Adding and Deleting Features
- Changing Feature Settings by Running the System Configuration Wizard

Adding and Deleting Features

Adding features to or deleting features from an existing Neurons for ITSM system is much easier than upgrading the system.

If you have an Enterprise Production Deployment, your action might apply to just one, some, or all of the servers, depending on the features that you are adding or deleting.

> To add servers to your deployment, follow the instructions under Installing the Neurons for ITSM System

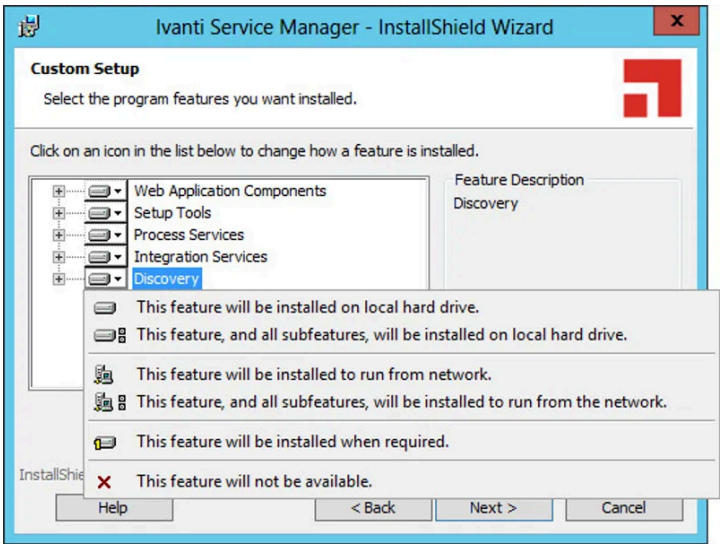
To add features or delete features, do the following:

1. Access the installation folder on the Ivanti Software product CD or zip file and run IvantiServiceManager.exe. Right-click and select Run as Administrator to ensure proper installation.

The installer checks for the prerequisite software components. If any of those components is not installed, the system prompts you to install them now.

2. Select **Install** at prompt. Installation of the prerequisite software can take several minutes.
3. If you are prompted to restart the system, select **Yes**. The system displays the **Welcome** dialog box.
4. Click **Next**. The system displays the **Program Maintenance** dialog box.
5. Choose Modify and click **Next**. The system displays the **Custom Setup** dialog box.

Custom Setup Dialog Box



6. Click the down arrow on each feature and make your selection.

> Adding features requires more hard-drive space. If there is not enough space on your hard drive, the installer displays a message.

7. Click Next and then click Install.

The Neurons for ITSM installation begins. The system displays a status dialog box showing the installation progress of each component over the next few minutes.

8. If you cancel the installation at any time, click Finish to close the installer.

When the components are installed, the system automatically launches the System Configuration Wizard.

9. Go to **Configuring Neurons for ITSM** for instructions about using the System Configuration Wizard.

Changing Feature Settings by Running the System Configuration Wizard

To change settings without adding or deleting features, you do not need the Ivanti Software product CD or zip file.

1. In the **Start** menu, click the down arrow to see the **Apps** menu, and then click System Configuration Wizard.

> You must click Run as administrator when you start the System Configuration Wizard manually.

2. Go to **Configuring Neurons for ITSM** for instructions about using the System Configuration Wizard.

4.8. Upgrading Neurons for ITSM from an Earlier Release

Landscapes

During the upgrade of Neurons for ITSM, you can upgrade the development (staging) or testing (UAT) landscapes and test them without affecting the production landscape. After you have verified that the development (staging) or testing (UAT) landscape upgrade was successful, go ahead with production landscape upgrade.

Reporting Feature

You must upgrade both Neurons for ITSM and the reporting feature at the same time. The versions of the software for both Neurons for ITSM and the reporting feature must be the same.

Licenses

The versions of Ivanti Service Management before the Release 2015.1 used a different licensing structure. When you upgrade to Neurons for ITSM Release version 2015.1, the system automatically uses the licenses that you used in the previous release.

In the System Configuration Wizard, on the **Configuration Application** page, there are fields to enter the license files. The system populates these fields automatically based on your previous license file. See **Configuring the Configuration Database**.

DSM Integration

If your Neurons for ITSM deployment includes a DSM integration, you must configure and verify your DSM integration for the new release.

> - For Ivanti Service Management releases prior to 2015.2, use the DSM Integration service component available in the Neurons for ITSM installer for DSM Integration.
> - For Ivanti Service Management releases 2015.2 and later, use the package from the Ivanti App Store for DSM Integration.

After you have upgraded Neurons for ITSM, access the Ivanti Software App Store and the DSM integration by doing the following:

1. Log into **https://support.heatsoftware.com/.**
2. Go to the Ivanti Software App Store by selecting App Store from the top menu.
3. From within the Ivanti Software App Store, open the DSM integration package.
4. Download the DSM integration package for Release 2019.1 and the accompanying instructions.
5. Follow the instructions to configure and verify your DSM integration.

For more information about DSM, see ***Working with DSM*** in the online help.

About Upgrading from Earlier Releases

- The upgrade to the N version of Ivanti Service and Asset Manager should always be performed from the N-1 version. For example, upgrading the on-premise version from 2018.1.1 to 2019.1.0 should be done only after the upgrade path of 2018.3.1. Skipping any released version of the application during an upgrade might cause the upgrade to fail or the application to stop working after the upgrade.
- 2019.2 is a cloud-only release and has no updates for on-premise customers.
- The system upgrade log file under c:\Logs\ReleaseTool will be “ApplyPatchConfigDB2019.3.1.log” for configuration database, which is expected for 2020.1 release.

When upgrading Neurons for ITSM, you must upgrade every release version between your currently installed and the latest available version in a sequential order. For example, if you are on ITSM version 2023.1 and want to install the latest version 2025.3, you need to install all the versions between 2023.1 and 2025.3 in a sequential order. Refer to the below table for information on all ITSM versions.

Upgrade From Release	Details	Install Release
2013.1.x	The current Ivanti Service Management version and the Inventory Management component, if present. See **Uninstalling the Inventory Management Component**.	2013.2.2
2013.2.x		2014.1.1
2014.1.x		2014.2.1
2014.2.x		2014.3.1
2014.3.x		2015.1
2015.1.x	Nothing.	2015.2
2015.2.x		2016.1
2016.1.x		2016.2
2016.2.x		2017.2
2017.2.x		2017.3.x
2017.3.x		2018.1.x
2018.1.x		2018.3.x
2018.3.x		2019.1.x
2019.1.x		2019.3.x
2019.3.x		2020.1.x
2020.1.x	Important steps to be performed. **Click here** to view the steps.	2020.2.x
2020.2.x	Nothing	2020.3.x
2020.3.x		2020.4.x
2020.4.x		2021.1.x
2021.1.x		2021.2.x
2021.2.x		2021.3.x
2021.3.x		2021.4.x
2021.4.x		2022.1.x
2022.1.x		2022.2.x
2022.2.x	Install the patch - **HotFix_BUG_997037_994439-OnPrem.zip** (install this patch after upgrading to 2022.3 and on fresh installation).	2022.3.x
2022.3.x	**Click here for details.**	2022.4.x
2022.4.x		2023.1.x
2023.1.x		2023.2.x
2023.2.x		2023.3.x
2023.3.x	Refer to **Installing Neurons for ITSM**	2023.4.x

Upgrade From Release	Details	Install Release
2023.4.x		2024.2.x
2024.2.x		2024.3.x
2024.3.x		2025.2.x
2025.2.x		2025.3.x
2025.3.x	Refer to **Finishing the System Configuration**	2025.4.x

- Installing upgrade versions in a sequential manner is mandatory. You cannot skip any version and install a higher version. For example, if you are on version 2023.2, you cannot directly install 2025.2. You have to install every version that is released in the sequence. However, the cache re-build is required only after installing the final version that you use to access the ITSM application.
- Installing the Additional App Domain provided with the package is mandatory. To install, follow the instructions given in the **Additional App Domain Deployment (Up to version 2025.3)**.
- The installation process is the same for all hosts in your deployment except for the additional features installation. And this selection is made in **step - 6 Your selection of features from the Setup dialog box depends on the role of the individual host in your deployment plan**.
- After the Additional App Domain powershell script is successfully run, the Cache Rebuild is automatically executed. If the Additional App Domain powershell script is not run in the same machine, then the Cache Rebuild should be executed manually. For instructions on how to execute Cache Rebuild, refer to **Neurons for ITSM Cache Rebuild**.

Upgrading from Ivanti Service Management Release 2017.3.x

If you are upgrading to Neurons for ITSM Version 2018.3, you must first upgrade to Neurons for ITSM Version 2018.1.1. You do not have to uninstall Ivanti Service Management Release 2017.3.x.

If you have a multiple server environment, you must shut off the notifications to the web servers before you upgrade the system. This is because when you make metadata changes as part of the upgrade, the system tries to synchronize them. As the versions are different, the system stops responding or takes a lot of time. See step 2 below.

Perform these steps on the system that hosts the Neurons for ITSM components:

1. Back up the following items:
   - Neurons for ITSM Configuration Database (ConfigDB)
   - Neurons for ITSM Application Database
   - Attachment folder (if used)
2. (Optional, if your system uses multiple servers) Deactivate the web servers by doing the following:

   a. Log into the Configuration Database. See the **Configuration Database** guide for Neurons for ITSM for information about using the configuration database.

   b. Open the **Web Servers** workspace. The system displays a list of web servers.

   c. Open a web server record.

   d. Clear the **Server is Active** check box.

e. Click **Save**.

f. Repeat for all of the web servers.

3. Access the installation folder on the Ivanti product CD or zip file and run *IvantiServiceManager.exe*. Right-click and select **Run as Administrator** to ensure proper installation.

   The installer checks for the prerequisite software components. If any of those components is not installed, the system prompts you to install them now.

4. Click **Install** at the prompt. Installation of the prerequisite software can take several minutes. If you are prompted to restart the system, click **Yes**.
5. In the **Upgrade Confirmation** dialog box, click **Yes**. The system displays the **Welcome** dialog box. The installer checks for space and other requirements before displaying the **Next** button.
6. In the **Welcome** dialog box, click **Next**. The system displays the **License Agreement** dialog box.
7. Choose **I accept the terms in the license agreement** and click **Next**. The system displays the **Destination Folder** dialog box.
8. Click **Next** to accept the default installation folder, or click **Change** and select a different folder. The system displays the **Setup Type** dialog box.
9. Your selection of features from the **Setup Type** dialog box depends on the role of the individual host in your deployment plan.
   - For the Demonstration or Proof-of-Concept Deployment, choose **Complete**, click **Next**, and install all components.
   - For the Minimum Production Deployment , choose **Complete**, click **Next**, and install all components.
   - For the Enterprise Production Deployment, your choice depends on the role of the host:
     - Production processing server: Choose **Custom**, click **Next**, and install all components except for the Operations Console.
     - Staging or UAT processing servers: Choose **Complete**, click **Next**, and install all components.
     - Web servers: Choose **Custom**, click **Next**, and only install the Ivanti Service Manager application server.

   For the Enterprise Production Deployment, note the following:

   - If your deployment includes different processing servers for production, staging, and UAT, you must install the License Manager in every one of those landscapes.
   - You can choose **Complete** and install all Neurons for ITSM features on all of your servers, but doing that takes up more disk space than necessary.

   When installing components, note the following:

   - To not install a component, click the down arrow next to the server icon next to the category name, highlight the component, right click, and select **This feature will not be available**.
   - If your deployment includes DSM, you do not need to install the DSM integration service because that component has been incorporated into Ivanti Service Management since Release 2015.2.
10. Click **Next**. The system displays the **Ready to Install the Program** dialog box.
11. Click **Install**. The system begins installing Neurons for ITSM and displays a status dialog box, showing the installation progress of each module over the next few minutes.
12. Click **OK** if the system displays a dialog box saying that reboot is required after installation.
13. If the system displays a dialog box saying that some of the files that need to be updated are currently in use, choose **Automatically close and attempt to restart applications** and click **OK**.

When the installation is completed, the system displays the System Configuration Wizard.

14. Go through the pages in the wizard making any changes that might be necessary.

> If you already have an existing administrator account, you do not need to create a new account. Just enter the information for your existing administrator account.

15. On the **Upgrade System** page, do not select **Do not create backup prior to upgrade** check box unless you already made a backup.
16. Notice that the system table version and metadata package version for the configuration database are not the latest versions. See the .

image

17. Click the top **Upgrade System** button.
18. When prompted, enter the administrator role credentials for the configuration database and click **OK**.
19. If your deployment has additional databases to upgrade, repeat steps 15 and 16 until:

    a. All of your databases have been upgraded:

    - Configuration database
    - Application database (IvantiSM)

    b. All of your environments have been upgraded:

    - Production
    - Staging
    - UAT
    - Development (optional combination of staging and UAT)

20. Click **Finish**.
21. Restart the host server.
22. (Optional) If you also installed the Neurons for ITSM reporting feature, you must upgrade it to the match the release number of your Neurons for ITSM system. Access the installation folder on the Ivanti Software product CD or download folder and run *ReportingService.exe* to upgrade Neurons for ITSM reporting feature.
23. Log in to the configuration database and activate the web servers that you deactivated in step 2.

Upgrading from Ivanti Service Management Release 2014.3 or Earlier

- About Upgrading From Ivanti Service Management Release 2014.3 or Earlier
- Uninstalling the Inventory Management Component

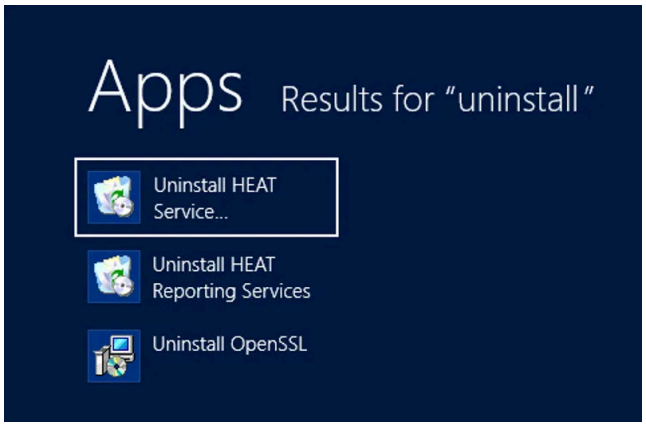
About Upgrading From Ivanti Service Management Release 2014.3 or Earlier

Beginning with Ivanti Service Management Release 2015.1, the Inventory Management component is known as Ivanti HEAT Discovery. As a result, before you can upgrade to the current version of Neurons for ITSM, if you are upgrading from Release 2014.3 or earlier, you must first uninstall the Inventory Management component.

Uninstalling the Inventory Management Component

Perform the following steps on the system that hosts Neurons for ITSM:

1. Click the arrow at the bottom of the **Start** menu to view the **Apps** menu.
2. Mouse-over the **Uninstall Ivanti Service Management** icons to find the one for the Inventory Management component.



3. Click **Uninstall Ivanti Service Management**.
4. In the confirmation dialog box, click **Yes**.

Upgrading from Ivanti Tryvium 2020.1 Release to 2020.2 Release

Perform the following steps before installing 2020.2 from 2020.1 release.

1. Log in to Neurons for ITSM with the Administrator role.
2. Open the Configuration console.
3. Under **Security Controls**, select **Security and Sessions**.
4. Under **Concurrent Session**, change the value of **Named User**, **Privileged User**, and **Non Privileged User** to *0*.
5. Save the changes and refresh the page and re-validate the values are changed.
6. Now you can install 2020.2 Neurons for ITSM.
7. After installation is complete, you can change the concurrent users values as it was earlier.

4.9. Using the License Manager

- About the License Manager
- Types of Licenses
- About License Bundles
- Using the License Manager

About the License Manager

Use the License Manager to track your license usage.

For the Demonstration or Proof-of-Concept Deployment and Minimum Production Deployment , the License Manager is installed on the Neurons for ITSM server along with the other Neurons for ITSM components.

For the Enterprise Production Deployment, the License Manager is installed on the Neurons for ITSM processing servers along with the other Neurons for ITSM components.

During installation, the System Configuration Wizard enabled you to import one production license, one non-production license, or both, depending on your landscape environment. You can import additional licenses, if needed. See Importing Licenses.

Types of Licenses

Neurons for ITSM mainly uses production and non-production licenses:

License Type	Description	Expiration
Production	Used to access the production landscape of a tenant.	Does not expire unless the MAC address is unspecified. In that instance, the license expires after 30 days. If you move the license to a new server with a different MAC address, then you must acquire a new license tied to the new MAC address.
Development	A non-production license used to access the staging and UAT landscapes.	Expires according to the maintenance agreement between your organization and Neurons for ITSM.
Not For Resale	A non-production license, mainly used by Ivanti Service Manager partners. It can also be used for staging, UAT, or testing if you already have development licenses.	Expires after 30 days if you do not specify the MAC address. After you set the MAC address, it expires according to the maintenance agreement between your organization and Neurons for ITSM.
Evaluation	A non-production license used by potential customers. Has the maximum number of named and concurrent user licenses.	Expires after 30 days. Is not associated with a MAC address.

There are two types of user licenses, for both production and non-production licenses:

- Concurrent licenses: For any user who is currently logged into the system. This is a shared pool of licenses. The concurrent license that is used is specified by the bundle that is specified for each role.

- Named user licenses: Specific to a certain user. The user can log in to the system from many places at one time and it is counted as only one license. You specify which users are named users in the employee record, and you can change which employees get a named user license at any time.

About License Bundles

- List of License Bundles and Components
- Assigning a Bundled License to a Role
- License Tracking in Neurons for ITSM

Neurons for ITSM has defined several license bundles, which are sets of modules, or user interfaces. Each role is associated with a license bundle.

List of License Bundles and Components

License Bundle	Components
Help Desk	• Dashboards and Reporting • Incident Management • Knowledge Management • Mobile • Self Service • Service Catalog • Survey • Voice Automation (optional add-on) • Workflow Automation
Service Desk	• Dashboards and Reporting • Incident Management • Knowledge Management • Mobile • Self Service • Service Catalog • Survey • Voice Automation (optional add-on) • Workflow Automation • Change Management • CMDB • Configuration Management • Problem Management • Service Level Management
Service Management	• Dashboards and Reporting • Incident Management • Knowledge Management • Mobile • Self Service • Service Catalog • Survey • Voice Automation (optional add-on) • Workflow Automation • Change Management • CMDB • Configuration Management • Problem Management

License Bundle	Components
	• Service Level Management • Availability Management • Event Management • Financial Management • Portfolio and Project Management • Release Management

The following are considered license add-on modules:

Add-on Module	Components
Discovery	Discovery
Voice	Voice
Mobile	Mobile

Assigning a Bundled License to a Role

To assign a bundled license to a role, follow the procedure in the "Assigning a Bundled License to a Role" topic in the Neurons for ITSM online help. By default, roles do not have a license bundle associated with them, so you must assign a license bundle to each role manually. If you do not, you may get license violation errors logged in your system even though you have the correct licenses.

> If you have imported licenses but do not see the correct license bundles displayed in the Configuration Console, clear the validation cache.
>
> Go to Configure > Cache Management and then click Reset cached validation lists only.
>
> See the Neurons for ITSM online help for more information.

License Tracking in Neurons for ITSM

You cannot use Neurons for ITSM unless you have a license. When you purchased Neurons for ITSM, you also purchased a set number of licenses. If needed, you can purchase additional licenses.

If you consume more seats than you have licenses for, the system does not log you out. It does, however, log an event into a log file. For example, if you have 10 concurrent licenses and try to log in 11 times, the system allows it.

Due to the way that web browsers work, it is very possible for one user to consume more than one license. For example, if a user logs in to Neurons for ITSM using Microsoft Internet Explorer and then opens Google Chrome and logs in again, another license is consumed. If this happens, the system logs a license violation error that you can view in the License Manager in the **System Audit Information** workspace.

Using the License Manager

- Logging Into the License Manager
- Configuring Landscape Information

- Importing Licenses
- Viewing the Discovery Node Count
- Viewing the Active Licenses
- Configuring the Email Address for Notifications
- Configuring the SMTP Settings
- Working with Scheduled Jobs
- Viewing Reports

Logging Into the License Manager

The License Manager uses the same credentials that are used to access the configuration database.

To log into the License Manager, follow these steps:

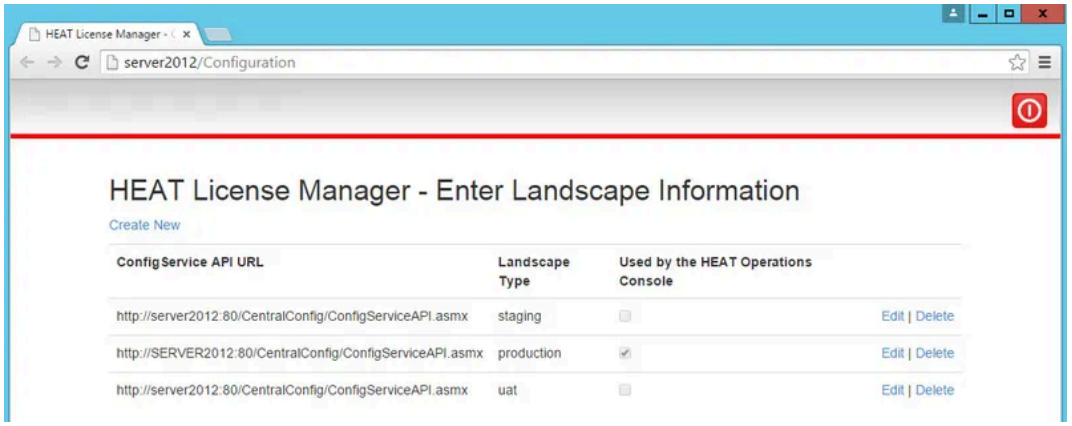
1. In the **Start** menu, click the down arrow to see the **Apps** menu, and then click License Server.
2. Select a landscape.
3. Enter your user ID and password. This is the same user ID and password that you use for the configuration database.
4. Click Continue.

Configuring Landscape Information

You can create, edit, and delete information about the landscapes to which the License Manager is associated.

1. Log into the License Manager.
2. Click either Configure Landscapes in the top right corner of the login page or click Configure Landscape Information from any page in the License Manager. The system opens a new browser window and displays the **Enter Landscape Information** page.

   Enter Landscape Information Page



3. To allow the Operations Console to use this landscape, check Used by the Operations Console on the line associated with the landscape.

   The Operations Console can only use one landscape at a time.

4. To edit the information for an existing landscape, do the following:

   a. Click Edit on the line associated with the landscape. The system displays the **Edit Landscape** page.

b. Make changes as needed.

c. Click Save.

5. To delete the landscape information, do the following:

a. Click Delete on the line associated with the landscape.

b. Click Delete at the confirmation message.

6. When you configured Neurons for ITSM using the System Configuration Wizard, the system automatically created a link to the landscape.

Therefore, in general, you do not need to add a link to a landscape. However, if you do need to add a new link to a landscape, follow these steps:

a. Click Create New. The system displays the **Create Landscape** page.

b. In the **ConfigService API URL** field, enter the URL of the configuration server.

c. Check Used by the Operations Console if the system should use this link to connect with the Operations Console. The Operations Console can only connect to one landscape.

d. Click Load Landscape Types and then select a landscape from the drop-down list.

e. Click Create.

Importing Licenses

When you initially configured the License Manager in the System Configuration Wizard, you imported either one production license file, one non-production license file, or both. (See **Configuring Neurons for ITSM** .)

If your deployment has multiple License Managers, when you import a license, ensure that you are importing it on the server that has the same MAC address as the license file.

For example, say you installed License Managers on the following:

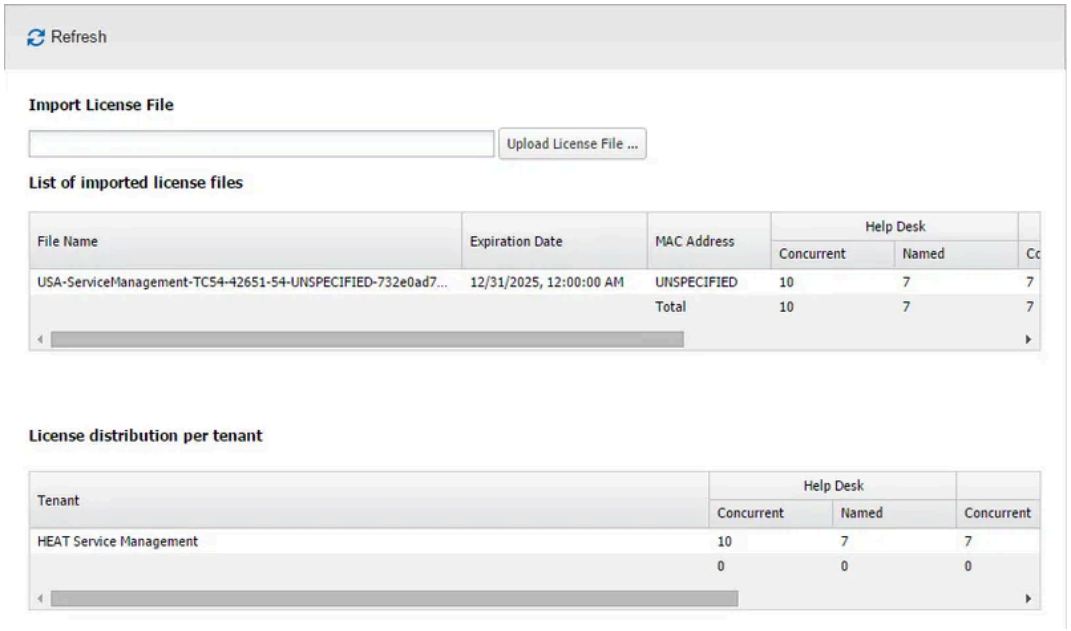
- Application Server 1 with a MAC address of AA🕒23🕒00
- Application Server 2 with a MAC address of BB🕒23🕒00

Then you ask for and receive a production license for MAC address AA🕒23🕒00. You import the license from the License Manager that is installed on Application Server 1. After you have imported the license, the License Managers on both Neurons for ITSM application servers respect the license. You do not need a second license for the second Application Server.

To import additional license files, follow these steps:

1. Log into the License Manager.
2. Click Licensing > Import License Files. The system displays the **Licensing - Import License Files** page.

Import License Files Page



3. Click Upload License File....
4. Navigate to and select a license file and click Open.

   The system adds the license to the system. This page shows the following information:

Field	Description
File Name	The name of the license file that you uploaded.
Expiration Date	The expiration date of the license file.
MAC Address	The MAC address associated with the license file.
Self Service	Self Service is the bundle name. The number of Self Service licenses contained in the license file. Broken into concurrent and named licenses.
Service Catalog	Service Catalog is the bundle name. The number of Service Catalog licenses contained in the license file. Broken into concurrent and named licenses.
Service Management	Service Management is the bundle name. The number of Service Management licenses contained in the license file. Broken into concurrent and named licenses.

5. Click the **delete** icon to remove the license file and its associated licenses from the License Manager.
6. Click the **show details** icon to display the details. The details contain the following information:

Field	Description
Bundle	The name of the bundle.
Name	The name of the module.
Named Licenses	The number of named licenses contained in the license file.
Concurrent Licenses	The number of concurrent licenses contained in the license file.
Expiration Date	The date when the license expires.

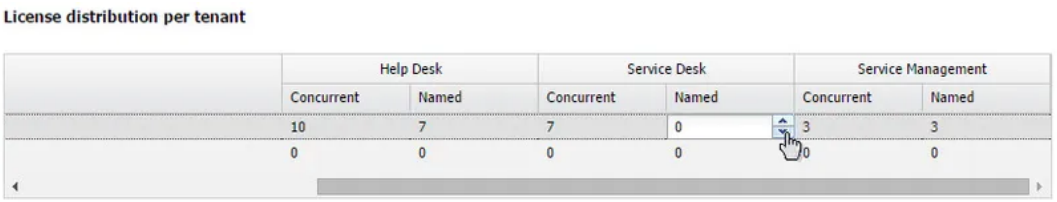
If your deployment only has one tenant, the information displayed here is the same as in the **List of imported license files** section.

If your deployment has more than one tenant, this table displays the license distribution across the tenants.

Field	Description
Tenant	The name of the tenant.
Self Service	The number of Self Service licenses allocated per tenant. Broken into concurrent and named licenses.
Service Catalog	The number of Service Catalog licenses allocated per tenant. Broken into concurrent and named licenses.
Service Management	The number of Service Management licenses allocated per tenant. Broken into concurrent and named licenses.

7. To distribute licenses among multiple tenants, click in a cell and change the number of licenses to allocate to a tenant. The total number of licenses across the tenants cannot exceed the total number of licenses. See **Allocating Licenses Across Tenants**.

   Allocating Licenses Across Tenants



Viewing the Discovery Node Count

This information tracks the trend associated with the Discovery licenses and is based on the Discovery **Count** scheduled job.

1. Log in to the License Manager.
2. Click Licensing > Discovery. The system displays the **Licensing - Discovery** page.

   This page shows the following information:

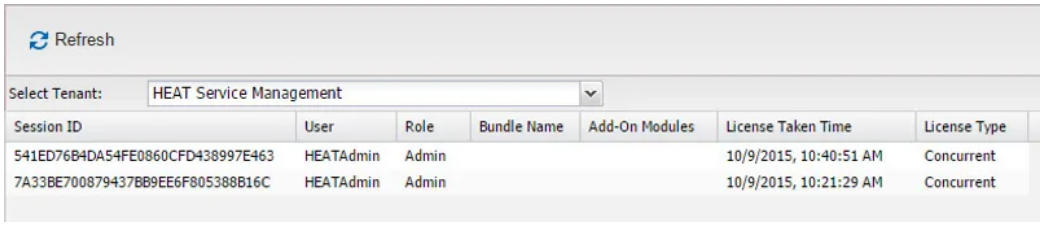
Field	Description
Tenant Name	The name of the tenant.
Last Scan Date	The date and time when the job was run.
Nodes Count	The number of configuration items found during the job.

Viewing the Active Licenses

This page displays the active sessions on Neurons for ITSM. It does not show active sessions on the configuration database.

1. Log in to the License Manager.
2. Click Licensing > Active Licenses. The system displays the **Licensing - Active Licenses** page.

   Active Licenses Page



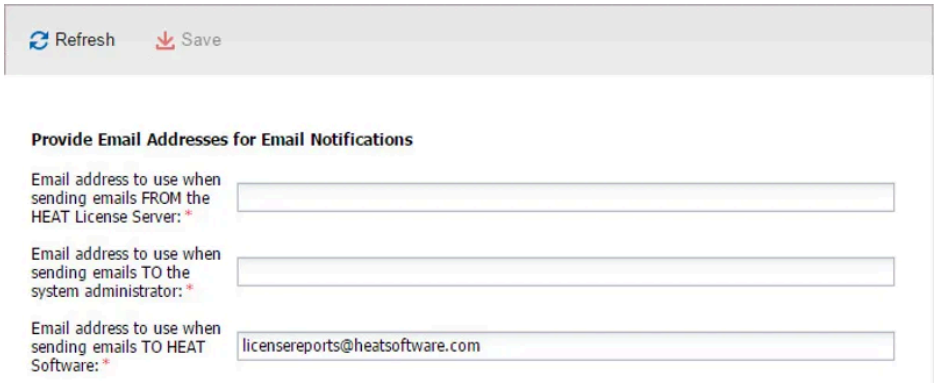
3. Select a tenant from the drop-down list. This page shows the following information:

Field	Description
Session ID	The session ID associated with the logged-in user.
User	The user name of a logged-in user.
Role	The role with which the user is logged in.
Bundle Name	The bundle associated with the license associated with the user. See **About License Bundles**.
Add-On Modules	Any add-on modules associated with the bundle. See **About License Bundles**.
License Taken Time	The time when the system used the license.
License Type	The license type. Can be named or concurrent.

Configuring the Email Address for Notifications

1. Log in to the License Manager.
2. Click Configuration > Email Address. The system displays the **Configuration - Email Address** page.

   Email Address Page



3. Enter the following information:

Field	Description
Email address to use when sending emails from the License Server	The email address to use as the FROM email address for emails sent from the License Manager.
Email address to use when sending emails to Ivanti Software	The email address to use as the TO email address for emails sent from the License Manager to Ivanti Software. The system sends license usage information based on the **License Violation Notification to Frs** scheduled job.
Email address to use when sending emails to the system administrator	The email address to use as the TO email address for emails sent from the License Manager to the system administrator of your organization. The system sends license usage information based on the **License Violation Notification to Admin** scheduled job.

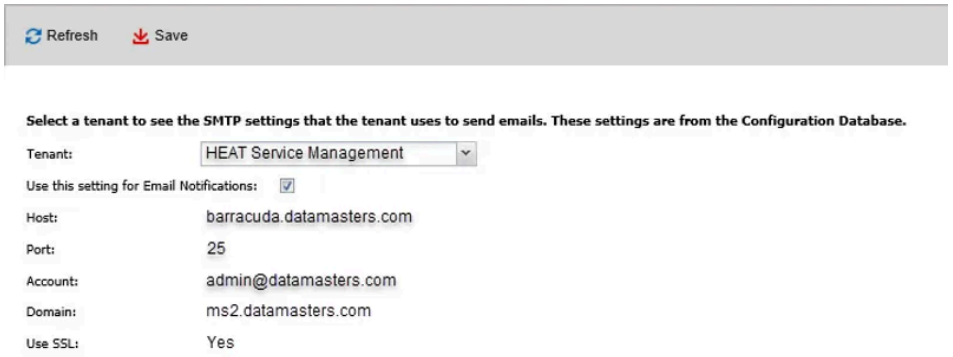
4. Click Save.

Configuring the SMTP Settings

Use this page to view and edit the SMTP settings used to send emails. The information on this page comes from the System Configuration Wizard.

1. Log into the License Manager.
2. Click Configuration > SMTP Setting. The system displays the **Configuration - SMTP Setting** page.

SMTP Setting Page



The system displays the following information:

Field	Description
Host	The email server host.
Port	The port associated with the email server.
Account	The account associated with this tenant. This is the same as the value in the **Username** field on the **Email Configuration** workspace in the Service Desk Console.
Domain	The domain associated with the account.
Use SSL	Specifies if this account should use SSL.

3. Select a tenant from the **Tenant** drop-down list.
4. Check Use this setting for Email Notifications to use these SMTP settings for emails.
5. Click Save.

Working with Scheduled Jobs

The License Manager has several predefined scheduled jobs. These are used to send information to the administrator on a regular basis instead of sending an email every single time there is an event. All violations are listed on the **System Audit Information** page.

The following are the predefined scheduled jobs that cannot be changed:

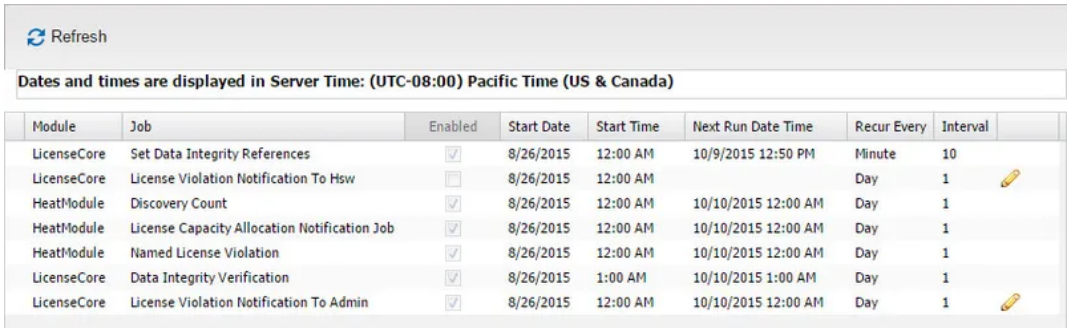
- License Capacity Allocation Notification Job: Determines if your licenses are valid and that you actually have the licenses that you are allocating.
- Named License Violation: Determines if you have more users listed as named users than you have named user licenses.
- Data Integrity Verification: Determines if the licenses have been modified.
- Discovery Count: Determines if the licenses for your configuration items are valid.

The following are the predefined scheduled jobs that you can edit:

- License Violation Notification to FRS: Sends an email to Ivanti Software with information about license violations. By default, this job is disabled.
- License Violation Notification to Admin: Sends an email to your system administrator with information about license violations.

1. Log in to the License Manager.
2. Click **System Jobs**. The system displays the **System Jobs - Scheduled Jobs** page.

   Scheduled Jobs Page



   This page contains the following information:

Field	Description
Module	The name of the module. Can be either LicenseCore or HeatModule.
Job	The name of the job.
Enabled	Specifies if the scheduled job is enabled.
Start Date	The start date of this scheduled job.
Start Time	The start time of this scheduled job.
Next Run Date and Time	The next date and time when the scheduled job runs.
Recur Every	The time period for the recurrence. Can be day, hour, or minute.
Interval	The interval for the recurrence.

For each entry, the system displays additional information at the bottom of the page:

Field	Description
Module	The name of the module. Can be either LicenseCore or HeatModule.
Job	The name of the job.
Run Time	The date and time when the scheduled job was run.
Details	Details of the job. If the job is not successful, contains detailed information.

3. To see the job details in a larger format, click the **show details** icon at the end of the row.
4. You can edit the details associated with the **License Violation Notification To Frs** and the **License Violation Notification to Admin** scheduled jobs.

   To edit a scheduled job, do the following:

   a. Click the **Edit** icon at the end of the row. The system displays a dialog box.

   b. Enter or change any of the following information:

Field	Description
Job	The name of the scheduled job. You cannot change this name.
Description	A description of the scheduled job.
Start Date	The start date and time.
Enabled	Specifies if the scheduled job is enabled.
Next Run Date Time	The next date and time when the scheduled job runs.
Recur Every	Specifies the recurrence for this job.

5. Click Save.

Viewing Reports

You can view two types of reports in the License Manager:

- System audit information

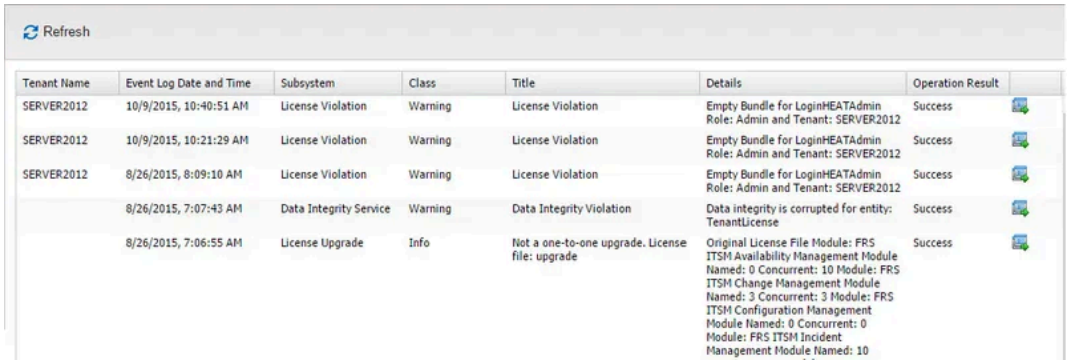
- Logs

Follow these steps to view the reports.

1. Log in to the License Manager.
2. Click Reports.
3. To view the system audit information, click System Audit Information. The system displays the **Reports - System Audit Information** page. See **System Audit Information Page**. Examples of the information on this page include license upgrades and license violations.

   We recommend that you review the information on this page if you have any problems and then review the information on the **Log** page. See **Log Page** .

   System Audit Information Page



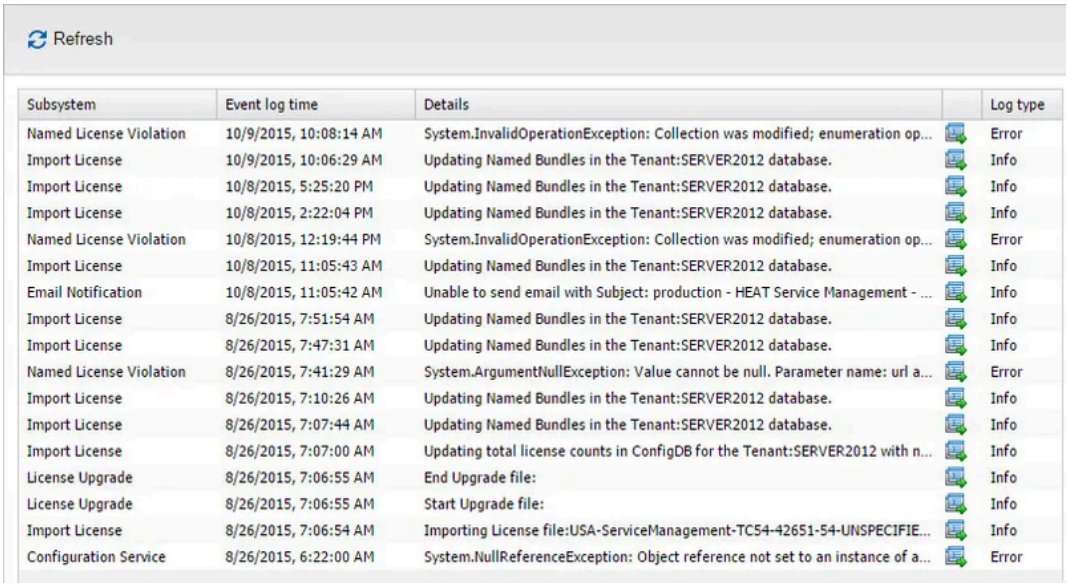
   This page contains the following information:

Field	Description
Tenant Name	The name of the tenant.
Event Log Date and Time	The date and time of the log event.
Subsystem	The specific area. Can be license violation, license upgrade, or data integrity service.
Class	The type of information. Can be warning, info, debug, fatal, or error.
Title	A short description of the event.
Details	A longer description of the event.
Operation Result	The result of the operation. Can be success or failure.

   If you cannot see all of the columns, resize the columns that you can see so that the remaining columns are displayed.

4. Click the **show details** icon at the end of the row to see the details. If you cannot see the **Show Details** icon, resize the columns until it is displayed.
5. To view log information, click Log. The system displays the **Reports - Log** page. Use this information for troubleshooting.

   Log Page



This page contains the following information:

Field	Description
Subsystem	The name of the subsystem. Can be email notification, import license, license upgrade, named license violation, or configuration service.
Event Log Time	The time and date of this log event.
Details	Information about this log event.
Log Type	The type of log. Can be info, error, or warning.

6. To see the log details in a larger format, click the **show details** icon at the end of the row.

4.10. Troubleshooting - Installation and Deployment

If there are problems with your installation or deployment, refer to this topic.

This topic does contain a long list of problems and its solutions which may be difficult to go through, it is recommended you search using relevant keywords to get to the right problem/resolution faster

Problem 1: Error message "Error upon executing commands: The following error occurred when authenticating with the tenant <tenantUrl>" is displayed.

Status: TenantNotFound

Possible Cause

The tenant specified is invalid.

Solution

Ensure that the tenant URL and tenant ID are valid.

Problem 2: Error message "Error upon executing commands: The following error occurred when authenticating with the tenant <tenantUrl>" is displayed. Status: AccessDenied

Possible Cause

Invalid user name or password is provided.

Solution

Ensure that the user name and password are valid.

Problem 3: Error message "Error upon executing commands: The following error occurred when authenticating with the tenant APPSERVER" is displayed. Status: InvalidRole

Possible Cause

No administrator rights.

Solution

Ensure administrator credentials exist. See **About Roles**.

Problem: 4
Error message "Error upon executing commands: Could not establish trust relationship for the SSL/TLS secure channel with authority '<IP_address>" is displayed.

Possible Cause

Did not connect to the Neurons for ITSM application server using HTTPS.

Solution

Configure the Neurons for ITSM application server to use SSL. See **Optional SSL Configuration**. Enter the URL for the Neurons for ITSM application server that is configured to use SSL.

Problem 5:
Error message "Failed to execute applyPatch: The remote server returned an unexpected response: (400) Bad ". is displayed.

Possible Cause

The database version is not up to date.

Solution

If the system table upgrade fails, review the log files and fix any metadata issues. Then upgrade the system table.

Problem 6:
Cannot view the login dialog box. This message is displayed - "Please use the subdomain when accessing the application". For example, https://yourname.saas.heatsoftware.com.

Possible Cause

The web.config file parameter "RequireTenantIdInURL" value is set to true.

Solution

1. Go to C:\Program Files\HEAT Software\HEAT\AppServer\ and open the web.config file.
2. Search for the parameter "RequireTenantIdInURL" and change its value to false.
3. Save and close the web.config file.

Problem 7:
After upgrading to the latest version of Neurons for ITSM, the list of tenants in the Operations Console is empty.

Possible Cause

During the upgrade, the system updated the connection strings for the landscape with the incorrect name for the data source.

Solution

In the Operations Console, edit each landscape to update the data source to the correct database server name. See **Upgrading Neurons for ITSM from an Earlier Release** for more information on how to do this.

Problem 8:
Failed to execute applyPatch: The request channel timed out while waiting for a reply after 00:59:59.7741622. Increase the timeout value passed to the call to request or increase the SendTimeout value on the binding. The time allotted to this operation may have been a portion of a longer timeout.

Possible Cause

By default, the upgrade tool is set to wait for one hour for a response from the Neurons for ITSM application server. This message indicates that the Neurons for ITSM application server is still applying the patch or package. The patch can be completed at any time.

Solution

Check the status of the upgrade tool later by looking at the patch log in the Configuration Database.

Problem 9:
Error message "No database backup location specified in configuration database. Please contact support." is displayed.

Possible Cause

- The backup location is not set in the configuration database.
- The backup failed to execute because of permissions issues.

Solution

Ensure the backup location is set the correct permissions are set.

Problem 10:
Error message "Error during metadata commit operation: System.Exception: The following errors were encountered when synchronizing the schema...:" is displayed.

Possible Cause

This indicates a serious problem in business object metadata and is a synchronization schema error. The system cannot synchronize the database schema change with the business object definition. If you get this error, the system usually stops upgrading the metadata.

Solution

If this happens in the production environment, restore the database from a backup that was made before the upgrade. Then contact Ivanti support so that they can look into the error.

Problem 11:
Error message "Error during metadata commit operation: DataLayer.SaaSDbException: Invalid object name 'Frs_ITFM_Account_Status'. ---> System.Data.SqlClient.SqlException: Invalid object name 'Frs_ITFM_Account_Status'" is displayed.

Possible Cause

Part of the database is corrupt.

Solution

Contact Ivanti Software support. They may suggest to restore the database from a backup that was made before the upgrade.

Problem 12:
Error message "Error during metadata commit operation: DataLayer.SaaSDbException: Invalid object name 'Frs_ITFM_Account_Status'. ---> System.Data.SqlClient.SqlException: Invalid object name 'Frs_ITFM_Account_Status'" is displayed.

Possible Cause

Part of the database is corrupt.

Solution

Contact Ivanti Software support. They may suggest that to restore the database from a backup that was made before the upgrade.

Problem 13

**Error message An error occurred when applying patch named <some_filename>.sql in package <some_package>.MetadataPackage to tenant <tenantUrl>.
Unable to execute SQL due to Invalid column name 'IPCMUrlPort'.:
update FRS_IPCM_Integration set IPCMUrlPort = 2323 where IPCMUrlPort is null
Unable to execute SQL due to Invalid column name 'IPCMUrl'.:
update FRS_IPCM_Integration set IPCMUrl = 'http://' + IPCMServerHost + ':' + convert(varchar(6), IPCMUrlPort) where IPCMUrl is null and IPCMServerHost is not null" is displayed.**

Possible Cause

An embedded Microsoft SQL statement is corrupt.

Solution

Review the Microsoft SQL statements.

Problem 14:

**Error message Error during metadata update operation. MetadataType: BusinessObject, ID: Frs_AuthenticationProvider#
SaaS.StandardizedMetadata.UpdateKeyNotFoundException: Update key not found: key = Rel2s
at SaaS.StandardizedMetadata.MetadataExtensions.Patch(XElement element, XElement differences)
at DataLayer.MetadataPatch.ApplyGroupedActions(IMetadataProvider provider, ISessionContext sessionContext, IEnumerable`1 definitionGroup) in c:\depot\Eng\SaaS\main\Platform\AppServer\MetadataServices\MetadataPatch.cs:line 1034 is displayed.**

Possible Cause

There is a problem updating the metadata and the system did not update the business object.

Solution

Review the business object to make sure it was updated.

Problem: 15

Error Message – Form

Error during metadata update operation. MetadataType: Form, ID: Task.WorkOrder System.Exception: Update key not found: key = Details
at SaaS.StandardizedMetadata.MetadataExtensions.Patch(XElement element, XElement differences) in c:\depot\Eng\SaaS\main\Platform\StandardizedMetadata\MetadataExtensions.cs:line 1377
at SaaS.StandardizedMetadata.MetadataExtensions.Patch(XElement element, XElement differences) in c:\depot\Eng\SaaS\main\Platform\StandardizedMetadata\MetadataExtensions.cs:line 1359
at DataLayer.MetadataPatch.ApplyGroupedActions(IMetadataProvider provider, ISessionContext sessionContext, IEnumerable`1 definitionGroup) in c:\depot\Eng\SaaS\main\Platform\AppServer\MetadataServices\MetadataPatch.cs:line 1016.

Error Message – Dashboard

Error during metadata update operation. MetadataType: Dashboard, ID: 8e2cce05-c593-4545-bf4d-5f719a9bd5a5 System.Exception: Update key not found: key = OLA Target Compliance (e94a602f-959e-4cf1-9c07-be4d5eb0da68)
at SaaS.StandardizedMetadata.MetadataExtensions.Patch(XElement element, XElement differences) in c:\depot\Eng\SaaS\main\Platform\StandardizedMetadata\MetadataExtensions.cs:line 1377
at SaaS.StandardizedMetadata.MetadataExtensions.Patch(XElement element, XElement differences) in c:\depot\Eng\SaaS\main\Platform\StandardizedMetadata\MetadataExtensions.cs:line 1359
at DataLayer.MetadataPatch.ApplyGroupedActions(IMetadataProvider provider, ISessionContext sessionContext, IEnumerable`1 definitionGroup) in c:\depot\Eng\SaaS\main\Platform\AppServer\MetadataServices\MetadataPatch.cs:line 1016

Error Message – Validation Data

Error during metadata update operation. MetadataType: ValidationData, ID: Justification# System.Exception: Update key not found: key = Justification#_culture#pt-BR
at SaaS.StandardizedMetadata.MetadataExtensions.Patch(XElement element, XElement differences) in c:\depot\Eng\SaaS\main\Platform\StandardizedMetadata\MetadataExtensions.cs:line 1377
at SaaS.StandardizedMetadata.MetadataExtensions.Patch(XElement element, XElement differences) in c:\depot\Eng\SaaS\main\Platform\StandardizedMetadata\MetadataExtensions.cs:line 1359
at SaaS.StandardizedMetadata.MetadataExtensions.Patch(XElement element, XElement differences) in c:\depot\Eng\SaaS\main\Platform\StandardizedMetadata\MetadataExtensions.cs:line 1359
at DataLayer.MetadataPatch.ApplyGroupedActions(IMetadataProvider provider, ISessionContext sessionContext, IEnumerable`1 definitionGroup) in c:\depot\Eng\SaaS\main\Platform\AppServer\MetadataServices\MetadataPatch.cs:line 1016

Error Message – Rule

Error during metadata update operation. MetadataType: Rule, ID: Incident# System.Collections.Generic.KeyNotFoundException: The given key was not present in the dictionary.
at System.Collections.Generic.Dictionary`2.get_Item(TKey key)
at SaaS.StandardizedMetadata.MetadataExtensions.Patch(XElement element, XElement differences) in c:\depot\Eng\SaaS\main\Platform\StandardizedMetadata\MetadataExtensions.cs:line 1386
at SaaS.StandardizedMetadata.MetadataExtensions.Patch(XElement element, XElement differences) in c:\depot\Eng\SaaS\main\Platform\StandardizedMetadata\MetadataExtensions.cs:line 1359
at DataLayer.MetadataPatch.ApplyGroupedActions(IMetadataProvider provider, ISessionContext sessionContext, IEnumerable`1 definitionGroup) in c:\depot\Eng\SaaS\main\Platform\AppServer\MetadataServices\MetadataPatch.cs:line 1016

Possible Cause

There is a problem updating the metadata.

Solution

No action.

Problem 16:
Error during metadata delete operation. MetadataType: Workflow, ID: 69d31c7248b54f92bc1bff415cd29e0f System.Exception: Could not deactivate workflow definition RecId: 69d31c7248b54f92bc1bff415cd29e0f ---> System.Exception: Error updating existing active workflow definition to inactive, RecId 69d31c7248b54f92bc1bff415cd29e0f at SaaS.WebUI.WorkflowMetadataServices.WorkflowMetadataUpdate.Delete(MetadataType metadataType, String id) in c:\depot\Eng\SaaS\main\Platform\AppServer\BPE\src\WorkflowMetadataServices.cs:line 409 Error during metadata insert operation. MetadataType: Workflow, ID: 871d9466c4754fa19ffb03de179cd057 System.Exception: Inactive definition exists with the same RecId: Name:LDAP Sync_LC_133k134k135k13k14k15k16k17k18k19k, ObjectType:ScheduleEntry, Definition RecId 871d9466c4754fa19ffb03de179cd057 at SaaS.WebUI.WorkflowMetadataServices.WorkflowMetadataUpdate.CreateWorkflowDefinition(String name, String objectType, String defRecId, String typeRecId, XElement workflow) at SaaS.WebUI.WorkflowMetadataServices.WorkflowMetadataUpdate.Insert(MetadataType metadataType, XElement workflow) at SaaS.StandardizedMetadata.MetadataProvidersContainer.Insert(MetadataType metadataType, XElement definition) at DataLayer.MetadataPatch.ApplyGroupedActions(IMetadataProvider provider, ISessionContext sessionContext, IEnumerable`1 definitionGroup)

Possible Cause

There is a problem updating the workflows.

Solution

No action.

Problem 17:
Discovery cannot detect assets on the network.

Possible Causes

- Service Center Configuration Manager (SCCM) service is stuck in starting mode.
- Using a domain account for running all Neurons for ITSM services but the domain account has insufficient rights.

Solution

Use the local system account and restart Neurons for ITSM.

Problem 18: System cannot search.Possible Cause Full-text search is disabled. Solution Enable full-text search. See Full Text Search .

Possible Cause

Full-text search is disabled.

Solution

Enable full-text search. See **Full Text Search** .

Problem 19:
The Neurons for ITSM demo database does not load properly.

Possible Cause

Full-text search is disabled.

Solution

Enable full-text search. See **Full Text Search** .

Problem 20:
Web servers do not work correctly.

Possible Cause

In multi-server environments, each Neurons for ITSM web server must meet the system requirements.

Solution

Refer to the *System Requirements and Compatibility Matrix for* Neurons for ITSM for more information.

Problem 21:
Provision report now returns an exception: Cannot Decrypt the Symmetric Key.

Possible Cause

The Neurons for ITSM reporting server cannot use the symmetric key to access the data from the report server database.

Solution

1. Click **OK**, but do not close the System Configuration Wizard.
2. Start the Reporting Services Configuration Manager and connect to your Microsoft SRSS instance.
3. Select **Encryption Keys** from the list in the left panel.
4. Click **Delete** and then click **Yes** at the confirmation message.
5. Return to the wizard and click **Provision Report Now**.

Problem 22:
Cannot download or edit reports in Chrome.

Possible Cause

The Chrome ClickOnce extension is not installed.

Solution

Install the ClickOnce extension for Chrome. Navigate to https://chrome.google.com/webstore/detail/eeifaoomkminpbeebjdmdojbhmagnncl# and download the extension.

Problem 23:
Cannot download or edit reports in Firefox.

Possible Cause

The Firefox Microsoft .NET framework assistant extension is not installed.

Solution

Install the Microsoft .NET framework assistant extension. Navigate to https://addons.mozilla.org/en-US/firefox/addon/9449 to download the extension.

Problem 24:
Cannot download files or run controls in Internet Explorer.

Possible Cause

Scripting is disabled.

Solution

Set properties in Internet Explorer as follows:

1. Go to the Tools > Internet Options > Security > Custom level page.
2. Set the following options to enable:
3. Run ActiveX controls and plug-ins
4. File download
5. Scripting > Active scripting

Problem 25:
Cannot open websites in Internet Explorer.

Possible Cause

The Neurons for ITSM is not set to be a trusted site.

Solution

Set properties in Internet Explorer as follows:

1. Go to the Tools > Internet Options > Security page.
2. Highlight Trusted sites and click Sites.
3. Click Add.

Problem 26:
Logging into a tenant using Internet Explorer fails.

Internet Explorer displays this message: You cannot login to the system now. Please contact your administrator. Additional information is available in the logs.

Possible Cause

Internet Explorer does not accept cookies if the host name of the DNS server or the server NetBIOS names contain an underscore '_'.

Solution

- Change the host name of the DNS server or its NetBIOS names so they no longer contain an underscore '_'
- Use a Chrome or Firefox browser.

Problem 27:
Unable to use integrated components, such as Voice.

Possible Cause

Using the Microsoft Windows operating system when using integrated components.

Solution

Change to the Microsoft Windows operating system. Refer to the *System Requirements and Compatibility Matrix for* Neurons for ITSM for more information.

Problem 28:
When trying to access the Configuration Database, the application redirects the user from http://localhost/HEAT to https://localhost on the machine where SSL is not setup.

Possible Cause

There is a known limitation in the 2018.3.1 release with respect to FedRamp requirements related to the URL redirect configuration.

Solution

To resolve this, there is a need to update an entry in the web.config file. Set the parameter highlighted below to ‘false’ and restart IIS. Once the process is completed, try to access **Configuration Database**.

</rule>

<rule name="HTTP to HTTPS redirect" stopProcessing="true">

<match url="(.*)">

</match>

<conditions>

<add input="" pattern="^localhost$" negate="true">

</add>

<add input="" pattern="off" ignoreCase="true">

</add>

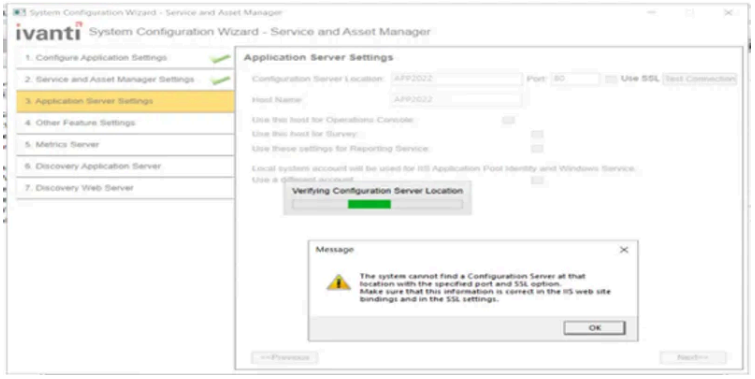
</conditions>

<action type="Redirect" url="https:///{R:1}" redirectType="Permanent">

</action>

</rule>

Problem 29:
From the System Configuration Wizard (SCW) - unable to connect to Central Central Config URL configuration.





Possible Cause

ErrorCode - 0x8007000d - URL Rewrite or IIS Hosting Bundle is not installed correctly.

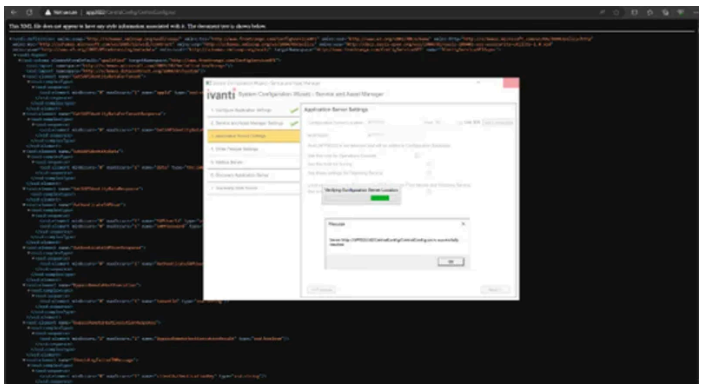
Solution

1. Open IIS.
2. Browse CentralConfig from Default Websites.



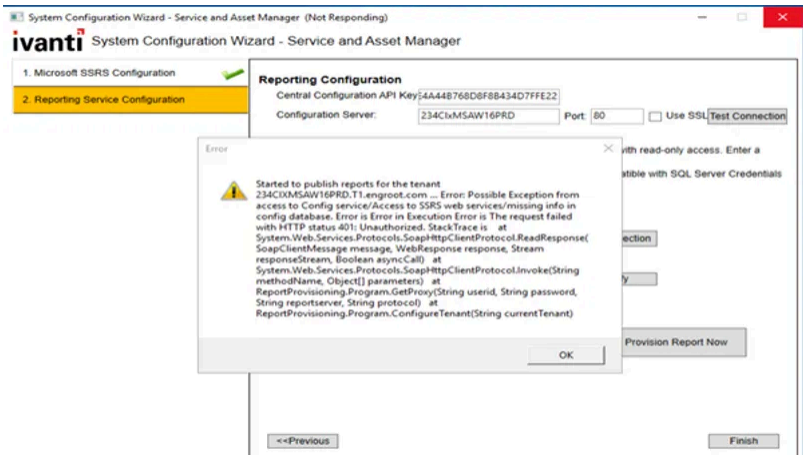
   Check CCF Url - **http(s)://<TenantID>/CentralConfig/CentralConfig.svc**. For example, http://app2022/CentralConfig/CentralConfig.svc.

3. If it is not accessible check error code in the browser - 0x8007000d.
4. Restart IIS and check if the CentralConfig URL is accessible.
5. If the CentralConfig is URL not accessible – install the **IIS Hosting bundle** and the **Restart IIS**.
6. Check the CentralConfig endpoint - http(s)://<TenantID>/CentralConfig/CentralConfig.svc.



7. If the CentralConfig is accessible, continue with the SCW setup.

Problem 30: System Configuration Wizard (SCW) - Report Provisioning fails displaying the following error - "Possible Exception from access to Config service/Access to SSRS web services/missing info in config database. Error is Error in Execution Error is The request failed with HTTP status 401: Unauthorized".



Possible Cause

The SSRS report service is not reachable by SCW.

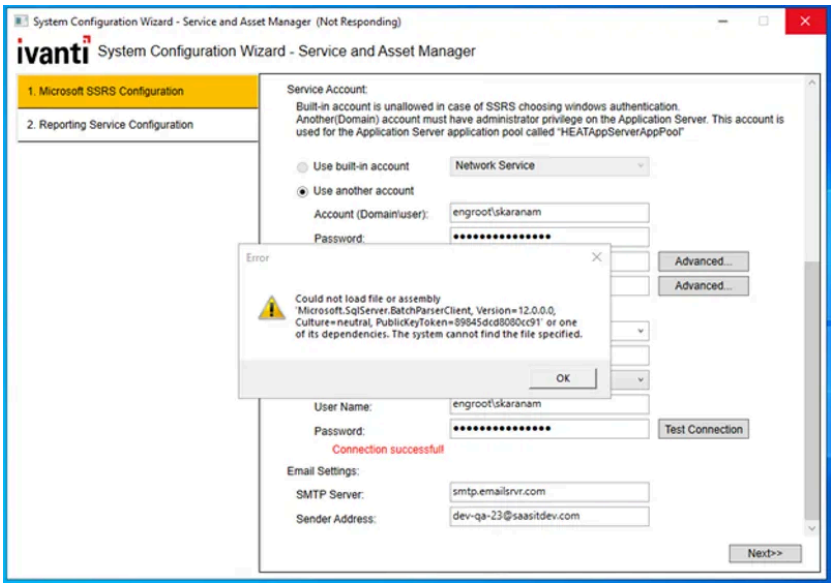
Solution

Check if the SSRS report configuration is reachable. After confirming, restart the SCW configuration.

Steps:

1. Open the Report Server Configuration Manager.
2. Start the SSRS and provide correct configuration details.
3. Recheck the ReportProvisioning.

Problem 31:
While installing System Configuration Wizard (SCW) Report Server in virtual machine, the following error is thrown - "Could not load file or assembly Micrososft.SqlServer.BatchParser,Version=12.0.0.0."



Possible Cause

If the assembly or its dependencies is not present in the VM.

Resolution

1. Install the Pre-Requisite installer given with the installation guide.

   or

2. Download and install the following packages manually:
   - SQLSysClrTypes.msi
   - SharedManagementObjects.msi
3. Install and check Report Server SCW setup.

Problem 32:
Cache rebuild does not run automatically.

Resolution

Run the cache rebuild manually. Refer to Installing Cache Rebuild.

Problem 33:
After updating the application from 2023.3 to 2023.4, if you are using WIS authentication, manually update the web.config file.

Resolution

- Take a backup of the existing file to ensure the configuration is preserved.
- After installation is completed, redo the updates to the **WIS web.config** as per the original configuration document, mainly pointing the **AssertionConsumerHandlerUrls** to the right addresses.

Problem 34:
Integration Service Server issues due to port change.

Resolution

As part of 2023.4 integration service is running as a separate service, hosted on port 1213 by default.

- If value for Integration Service Server is 80, change it to 1213.
- If the value for https port is 443, change it to something else.
- Ensure to have a separate port and not the same as other appdomains.

Problem 35:
License Server Landscape issues such as login issue with error message "Landscape is not defined in License Server" or "Unable to Load License".

Resolution

1. Run the following query:

```
SELECT * FROM .[dbo].[Frs_LS_LicenseServerLandscape]
```

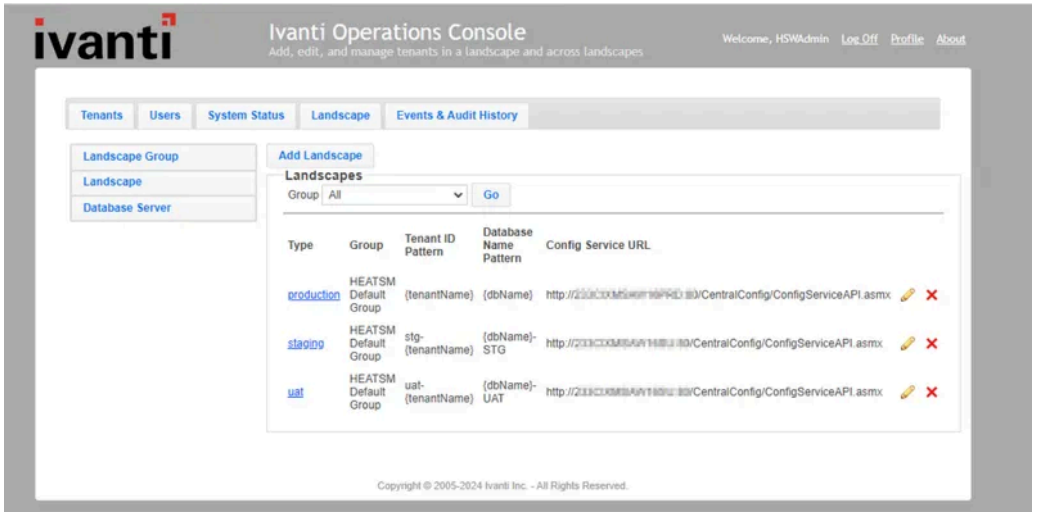
```
"select * from Landscape"
```

> Run both the above queries from the config tenant.

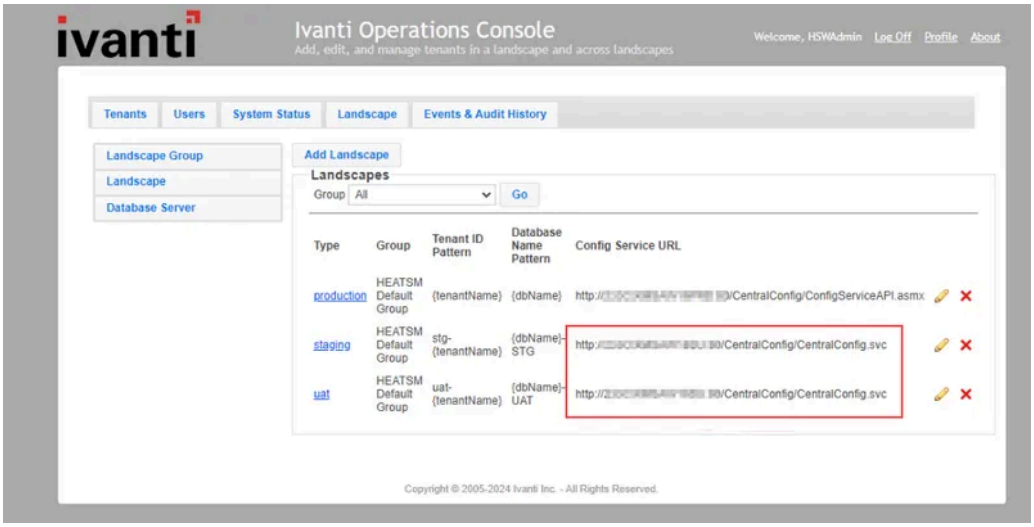
2. Ensure that the value in **ConfigServiceUrl** field is valid (check if the service definition loads when adding it in a browser). If not, update it to a valid central config url, for example, http://domainname/Centralconfig/CentralConfig.svc .

If using OpsConsole, follow the resolution below:

3. Check the landscape configuration from the DB or through ops console. If it is similar as shown the image below, that is if it has - /ConfigServiceAPI.asmx at the end, update those specific landscapes to have /Centralconfig/CentralConfig.svc route instead.



After update, the landscape configuration should be as shown the below image:



4. Restart IIS and login into both AppServer and LicenseManager.

> The **CentralConfig.svc** file should only be present only on the landscapes where the upgrade is executed from.

Problem 36: Cannot find APIKey in Header.

Resolution

The **central-config-api-key** should be the same in all **appdomains appsettings.json** file and in **Frs_data_api_key** table in the ConfigDB.

Update the central-config-api-key in the respective appdomains **appsettings.json** file.

Problem 37:
Issue with ActionServices.

Resolution

Refer to **ActionServices Troubleshooting** topic.

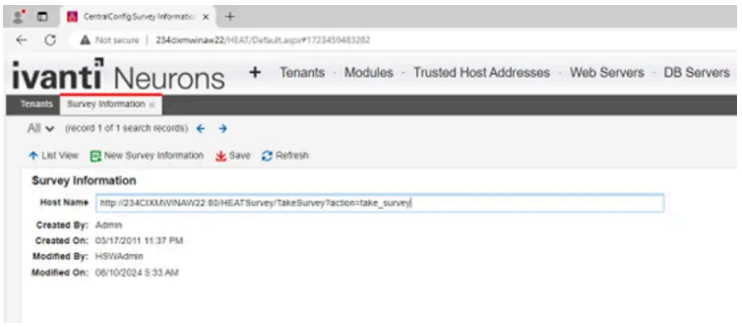
Problem 38:

Survey not opening - getting a 404 error when trying to open survey from incident or email.

Resolution

Open the Survey Information workspace from Config tenant.

Update URL in the Host Name field by removing "/Public" and ".aspx" as shown in the image below:



From {http/https}:///HEATSurvey/Public/TakeSurvey.aspx?action=take_survey to {http/https}:///HEATSurvey/TakeSurvey?action=take_survey

If the survey url opens from the machine where the application is hosted but not from other machines, make sure the {**hostname**} parameter has the fully qualified domain name instead of the machine name.

4.11. Chat Configuration

The administrator has to configure the following settings for the chat feature to work.

- Global Constant - In the Configuration module, under **Global Constants**, set the **ChatEnabled** field as **True**.
- User role - To make the chat icon available for a user, the administrator should enable chat for the user role.

  To enable the chat feature for a user role:

  a. Log in to Neurons for ITSM with the Administrator user role.

  b. From the Configuration console, select **Configure** > **Roles and Permissions**.

  c. Select the role for which you want to enable the chat feature.

  d. Ensure the **Overwrite default branding options with the options listed below** check box is selected.

  e. Select the **Enable Chat as Analyst** check box.

> - If you select the **Overwrite default branding options with the options listed below** check box, all check boxes under **Branding Options** will be disabled. Ensure to select them as needed.
> - Chat requests to the Service Desk console team get routed using the Round Robin algorithm based on the Service Desk Analyst's login state.

Configuring Chat

The chat configuration settings are pre-defined, however, you can change the settings as need be.

Session Configuration Settings:

- **Concurrent Chat Sessions per Analyst**: Number of chat requests that can be assigned to the Service Desk Analyst at a time.
- **View User Chat History for last Created Incident(s)**: Number of previous chat sessions' history that should be stored. History is stored only when an Incident is created in the chat session.
- **First Inactivity Timeout**: Timeout limit when a first warning message is sent to the Self Service Mobile portal user when inactive on a chat session.
- **Second Inactivity Timeout**: Timeout limit when a second warning message is sent to the Self Service Mobile user when inactive on a chat session.
- **Session Expiration Timeout**: Timeout limit when the Self Service Mobile portal user is logged out when inactive on a chat session.
- **Analyst Session Timeout**: Timeout limit when a Service Desk Analyst's status is turned **Offline** when no chat session is assigned.
- **User Timeout**: Timeout limit when a Self Service Mobile portal user is logged off from a chat session when unable to assign the chat session with the Service Desk Analyst.

Details:

- **Online Message Header**: Header displayed to the Self Service Mobile portal user when the Service Desk Analyst staff is online and ready to pick up chat sessions.

- **Offline Header**: Header displayed to the Self Service Mobile portal when the Service Desk Analyst is offline and not available to pick up chat sessions.
- **Hop Header**: Header displayed to the Self Service Mobile portal when the Service Desk Analyst staff is offline during non-hours-of-operation.
- **Join Header**: Header displayed to the to the Self Service Mobile portal when a Service Desk Analyst joins the chat session.
- **Fail Join Header**: Header text displayed to the to the Service Desk Analyst when unable to assign a Service Desk Analyst to the chat session.
- **Knowledge Message**: Link to knowledge base displayed to the Service Desk Analyst user when unable to assign a Service Desk Analyst to the chat session.
- **Warning Header**: First warning header displayed to the Self Service Mobile portal user when inactive on the chat for the set time.
- **Second Warning Header**: Second warning header displayed to the Self Service Mobile portal user when inactive on the chat for the set time.
- **Typing**: Typing indicator displayed in the chat window text box.
- **Send Message Place Holder**: Message displayed in the chat window text box.
- **Incident Created text**: Message displayed in the chat window when an incident is created from the chat session.
- **User Disconnected**: Message displayed to the Service Desk Analyst and the Self Service Mobile portal user when the chat session gets disconnected.
- **Online Message**: Message displayed to the Self Service Mobile portal user when the Service Desk Analyst staff is online.
- **Offline Message**: Message displayed to the Self Service Mobile portal user when the Service Desk Analyst staff is offline during hours-of-operation.
- **Hop Message**: Message displayed to the Self Service Mobile portal user when the Service Desk Analyst staff is offline during non-hours-of-operation.
- **Joining Message**: Message to the Self Service Mobile portal user when trying to assign a Service Desk Analyst to the chat session.
- **Join Message**: Message displayed to the Self Service Mobile portal user when a Service Desk Analyst joins the chat session.
- **Fail Join Message**: Message to the Self Service Mobile portal user when unable to assign a Service Desk Analyst to the chat session.
- **Knowledge Button**: Second line of the knowledge base link message when unable to patch a Service Desk Analyst to the chat session.
- **First Warning text**: First warning message displayed to the Self Service Mobile portal user when inactive on the chat for the set time.
- **Second Warning Text**: Second warning message displayed to the Self Service Mobile portal user when inactive on the chat for the set time.
- **Expired Warning Text**: Message displayed to the Self Service Mobile portal user just before logging off the chat session when the user is inactive on the chat for the set time.
- **Session Expired Text**: Message displayed to the Self Service Mobile portal user after logging off the chat session when the user was inactive on the chat for the set time.
- **User Wait Expired Text**: Message displayed to the Self Service Mobile portal user when unable to assign a Service Desk Analyst to the chat session.

When you make changes to the Chat Configuration, it is recommended that you clear cache for the changes to be effective. To do it, open the Configuration console > **Cache Management** > **Reset Chat Objects**.

Create Incident Quick Action

Service Desk Analyst can create an Incident from the chat session which is based on the **Create Incident** Quick Action added to the **ivnt_Chat Business Object**. This Quick Action can be customized as needed.

Customizing the Create Incident Quick Action

1. Log in to Neurons for ITSM with the Administrator user role.
2. Open the Configuration console.
3. Under **Build**, click **Business Objects** > search and open **ivnt_Chat Business Object**.
4. Click **Create Incident**.
5. Edit the data as needed and click **Save**.

4.11.1. Chatting with Self Service Mobile portal Users

If your administrator has assigned you as the chat analyst, you can pick up support chat requests to answer the Self Service Mobile portal users' queries or solve their issues without them having to raise an Incident.

In case the chat icon is not available for your user role, then you are not assigned the chat analyst role. For further assistance, reach out to your organization's administrator.

The number of chat requests you will be assigned at a time is setup by your administrator. A minimum of 1 and maximum of 20 chat requests can be assigned to you at a time. However, only when you accept the chat request the session starts.

To chat with the Self Service Mobile portal user

1. Log in to Neurons for ITSM.
2. Click the chat icon > **Turn On**.
3. Confirm by clicking **Turn On** in the prompt message.

   The chat icon turns green indicating you are **Online** and available to take up chat requests.

   > Unless your chat status is **Online**, you will not be assigned to any chat request.

4. The chat icon turns red indicating you have a chat request assigned to you. Click the chat icon.

   The name and department of the user requesting the chat is displayed.

5. Click **Start Chat** to accept and start chatting.
6. Respond by typing a message in the text box and click the **Send** icon.

   > If the chat icon turns red again, it indicates a new chat request. You can accept and chat with multiple users at a time. The number of chats you can be assigned on at a time is configured by your administrator.

7. Click the **Create an Incident Now** link in the chat window to create an Incident if you are unable to resolve the issue.
8. The username and other fields are auto-filled. Modify as need be and click **Ok**.

   An Incident will be created, and the Incident ID will be displayed on the chat window.

9. Click the **Incident ID** link from your chat window to open the Incident.

To skip a chat that is patched to you

- Do not click on the chat icon if do not want to accept a chat for some reason.

  The chat will be assigned to someone else while your status will still be **Online**.

  > However, if you do not accept chat requests subsequently for the number of times set as a limit by your administrator, your status will automatically turn **Offline**. Reach out to your administrator, to know the limit.

Additional actions you can perform from the chat window

- Drop-down arrow - displays the following options:

    - **Chat History** - displays the chat history from sessions that has Incident created. How many previous sessions' history should be stored is configured by your administrator.
    - **Full screen** - displays the chat window in full screen.
    - **Increase text Size** - increases the font size.
    - **View Profile** - displays the profile of the current user in the chat.
- **Copy** - copies the chat text.
- **Minimize** - minimized the chat window.
- **Close** - closes the chat window.

> - If an Incident was created in the previous chat session, the Incident ID will be displayed in the current chat window.
> - You cannot close an active chat window unless the Self Service Mobile portal user closes it.

5. Configuration Database

Use the Neurons for ITSM configuration database (ConfigDB) for the following:

- To manage different instances (such as production, staging, and UAT) of tenants.
- To update your environment. The information in the ConfigDB is originally populated by the System Configuration Wizard, which is used when installing Neurons for ITSM.

> This is the master configuration database for your Neurons for ITSM system. Do not make any changes here, except in the few areas described in the 'Workspaces that Can Be Used' section. Making unauthorized changes to the ConfigDB can lead to disastrous results and may be unrecoverable. Use the ConfigDB with **extreme** caution.

Intended Audience

This guide is intended for advanced on-premise Neurons for ITSM - Version 2025.3 administrators and Ivanti Software personnel.

Document Organization

This help lists the workspaces in the ConfigDB and what, if any, tasks can be performed. This document contains the following sections:

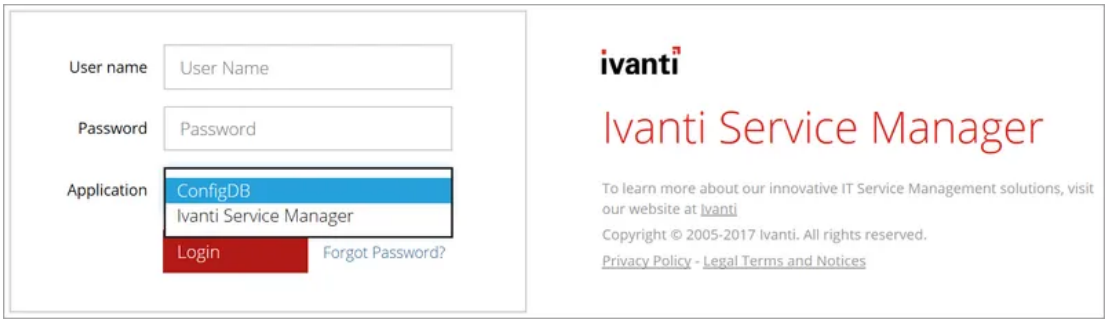
- Information about logging in. See **About Logging into the Configuration Database**.
- Workspaces that are also available in the Neurons for ITSM Service Desk Console or the Neurons for ITSM Configuration Console. See **Workspaces that Already Exist**.
- Workspaces that are for internal use only. See **Internal Use Only Workspaces**.
- Workspaces that are populated by the System Configuration Wizard. Do not make any changes to these workspaces. See **Workspaces Populated by the System Configuration Wizard**.
- Workspaces where you may make changes. See **Workspaces that Can Be Used**.

5.1. About Logging into the Configuration Database

To use the Neurons for ITSM configuration database (called ConfigDB), you must log in first. Only administrators can log into the ConfigDB.

By default, the system displays a drop-down menu allowing you to choose either ConfigDB (the Neurons for ITSM configuration database) or ISM (the Neurons for ITSM application database).

Logging in to the ConfigDB



If you do not see the **Application** drop-down menu, you must configure Neurons for ITSM to show it. For instructions on doing this, see the "Logging into Neurons for ITSM" section of the *Installation and Deployment Guide for* Neurons for ITSM .

5.2. Workspaces that Already Exist

The following workspaces found in the ConfigDB also exist in the Neurons for ITSM Console or the Neurons for ITSM Configuration Console. Therefore, they are not described in this document. For information about these workspaces, see the Neurons for ITSM online help within the application as well as at the Ivanti Product Documentation website. Click the "Service Manager" tile to see a list of the documents available.

- Employee
- All Logs
- Logging Configuration
- Notification
- Patch Log
- Encryption Key
- FRS Application Update Notes
- Ivanti Release Package
- Ivanti Release Project
- Ivanti Transaction Detail
- Ivanti Transaction Set
- Logon History
- Public Key
- Schedule Entry
- WS Mapping
- WS Proxy

5.3. Working With Features

- Adding a New Feature
- Changing the State of a Feature

Adding a New Feature

In Neurons for ITSM, you can add features. An example of a feature is Google Analytics. Follow these steps to add a feature:

1. Log in to the ConfigDB.
2. From the workspace selector bar, click More....
3. Select Feature. The system displays the **Feature** workspace.
4. Click New Feature.
5. In the **Feature Name** field, enter a name for the feature.
6. Click Save.

Changing the State of a Feature

Follow these steps to change the state of an existing feature.

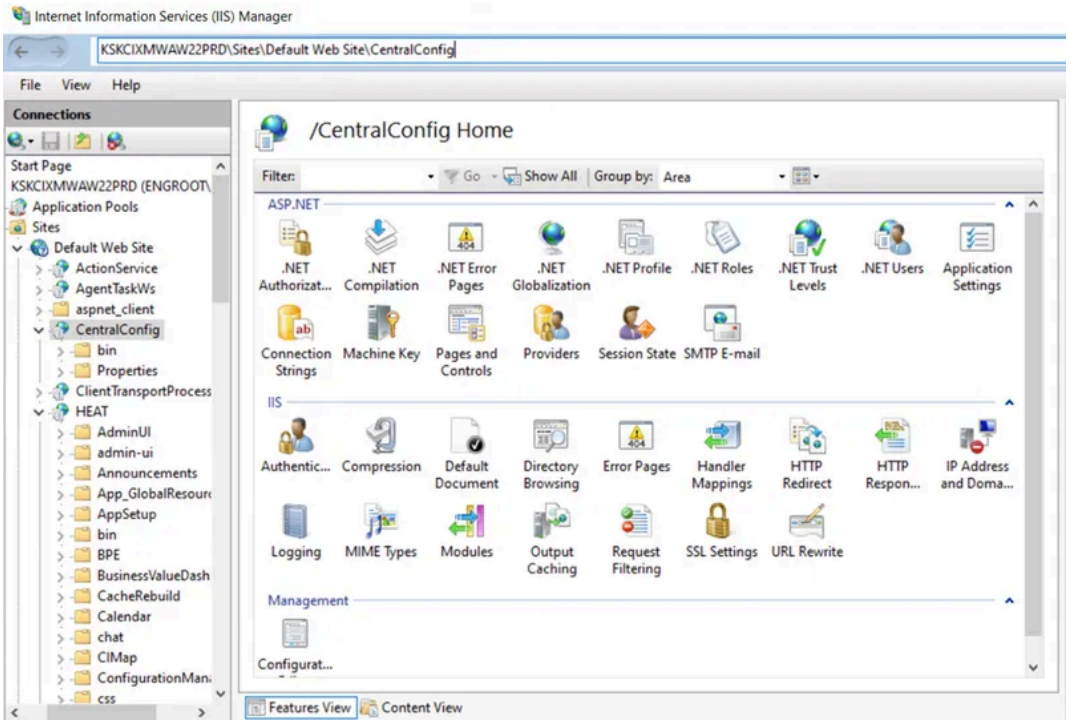
1. Log in to the ConfigDB.
2. From the workspace selector bar, click More....
3. Select Feature Management. The system displays the **Feature Management** workspace.
4. Click New Tenant Feature.
5. Enter information into the fields:
    - In the **Tenant** field, enter or select the tenant that this feature applies to.
    - In the **Feature** field, select the feature.
    - In the **State** field, select a state from the drop-down list. You can choose from the following states: Unconditionally on, Trial period, Trial Completed, Disable.
6. Click Save.

5.4. Securing IIS website

Neurons for ITSM on-premises installation allows installing ITSM modules on multiple machines and this requires securing the infrastructure within and outside the intranet. To ensure secure installation, administrators can define and manage rules that allow or deny access of specific IP addresses, range of IP addresses, or domain name(s) to the modules.

Add access restriction for CentralConfig application

1. Open the Internet Information Services (IIS) of the machine in which the **CentralConfig** is installed.
2. Select **CentralConfig** web application.
3. Select the **Features View** tab and select the **IP Address and Domain Restriction** option.



4. Alternatively, if **IP Address and Domain Restriction** is not available,

   a. Open **Server Manager** by clicking **Start** > **Administrative Tools** > **Server Manager**.

   b. Select **Role-based or Feature-based**, then select your server and click **Next**.

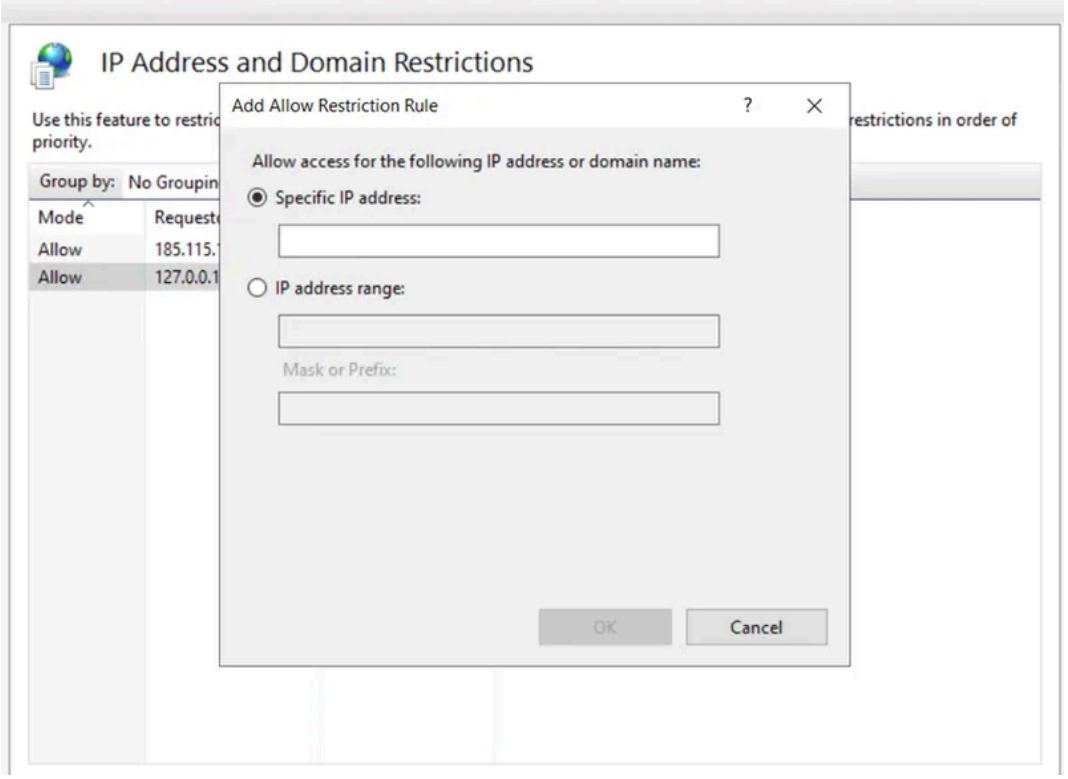
   c. Click **Add Role Services** to add the required role.

   d. From the **Select Role Services**, navigate to **Web Server (IIS)** > **Web Server** > **Security**.

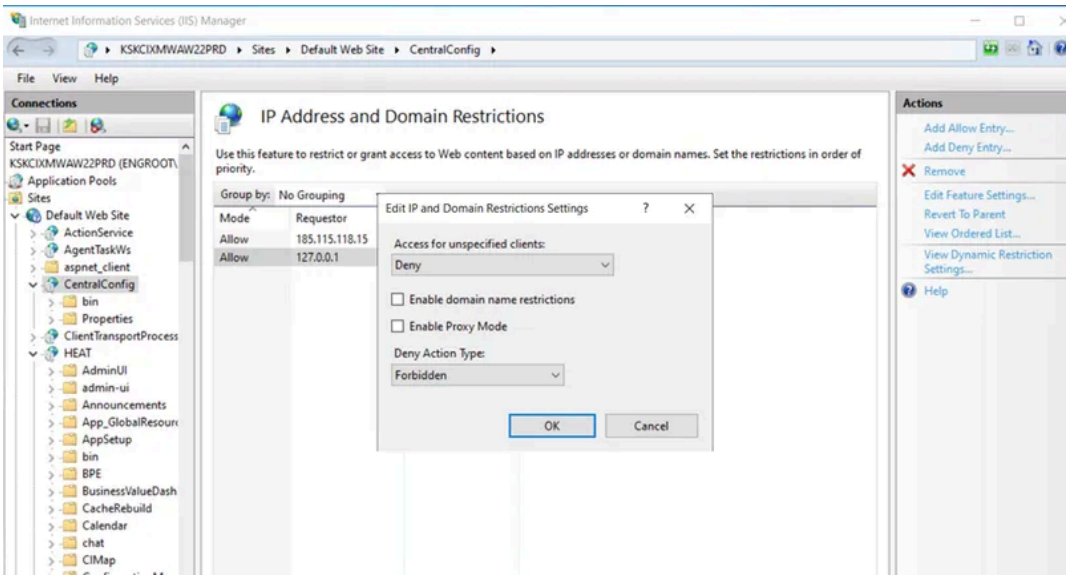
   e. Select the **IP and Domain Restrictions** checkbox and click **Next**.

   f. From the **Confirm Installation Selections** page, click **Install to add the IP and Domain Restrictions** role services.

5. Select **IP Addresses and Domain Restrictions** > **Add Allow Entries**.
6. Add the IP Addresses you want to allow.



7. Click **Edit Feature Setting** on the right panel.
8. Ensure the **Deny** checkbox is selected.



9. Add the IP Address of Central Config Machine.
10. Reset IIS.

> Repeat the process to add more IP Addresses if more services are installed on other different machines.

5.5. Internal Use Only Workspaces

The following workspaces in the ConfigDB contain information that you can view, but that you should never change. These workspaces are only used by Ivanti for internal system use.

> Do not change the values in any of these workspaces! Changing any values can lead to system degradation.

- CMDB Import History
- Feature Management
- IM Client Agent Version
- Installer Types
- Logging Server Configuration
- Modules
- Remote Host Blocked List
- Service Names
- User Feature Settings

5.6. Workspaces Populated by the System Configuration Wizard

The following workspaces found in the ConfigDB contain information that was populated based on information entered in the System Configuration Wizard when Neurons for ITSM was installed.

We recommend that you do not change any of the values in these workspaces.

For complete information about the System Configuration Wizard and the Neurons for ITSM installation, see the *Installation and Deployment Guide for* Neurons for ITSM 2025

- Database Servers
- Log Operations Locations
- Integration Services Configuration
- Metrics Server
- Report Server
- Survey Information
- Trusted IP Addresses
- Web Servers

5.7. Central Config API Authentication

APIs in the Central Config can be accessed only by known users by authenticating the API call. The API Header will pass the API Key which will be authenticated and passed by the Central Config server. This prevents un-authorised users from accessing the APIs.

Install/upgrade Ivanti Service Manager System Configuration Wizard

On installing or upgrading the Service Manager System Configuration Wizard, the API Key is auto-generated in the Central Config DB. This key is used to to authenticate the API calls to the Central Config Server. However, the following should be monitored while installing/upgrading.

Verify the webconfig file

Once the installation/upgradation is complete, verify the webconfig file has the following:

- In the config section:

  <section name="CentralConfigApiKey" type="System.Configuration.NameValueSectionHandler" />

- In the new protected section:

  <CentralConfigApiKey> <add key="CentralConfigApiKey" value="" /> </CentralConfigApiKey>

Verify the appsettings.json file

Once the installation/upgradation is complete, verify that the appsettings.json file has the following:

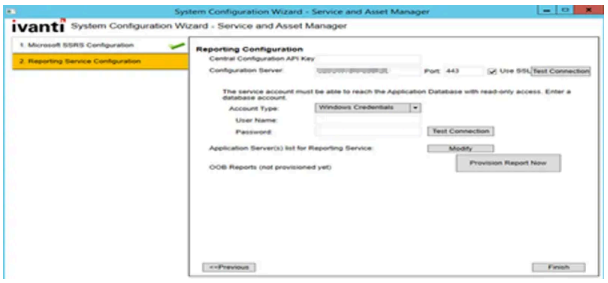
"central-config-api-key": "CAEFXXXX8C847D5XXXX698D8XXXXXX"

Update for Teams Bot Service

If you have installed Teams Bot Service, update the **CentralConfigApiKey** value in appsetting file. from C:\inetpub\TeamBotService/appsettings.json. Get the CentralConfigApiKey from the Central Config DB table (Frs_data_api_key).

Updates if Service Manager System Configuration Wizard and Service Manager Reporting Wizard are installed on different machines

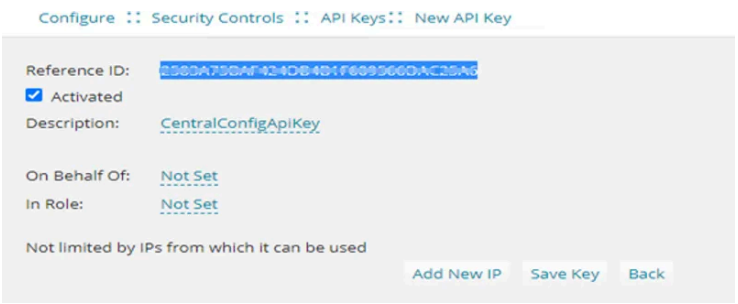
If the Service Manager System Configuration Wizard and Service Manager Reporting Wizard are installed on different machines, update the Centralconfig api key when installing the Reporting Wizard as shown in the below image. Get the CentralConfigApiKey from the Central Config DB table (Frs_data_api_key).



Updating the CentralConfigAPIKey

In case you wish to update/change the CentralConfigAPIKey, do so by following the below steps:

1. Stop all services that are referencing the centralconfig.
2. Log in to Service Manager with Admin credentials and open the Configuration console.
3. Select Security Controls > CentralCofigAPIKeyGroup and note down the description .
4. Delete the existing CentralConfigAPIKey and recreate a new key inside the CentralConfigAPiKeyGroup as shown in the below image.



5. Once the Key is created in the CentralConfig tenants, re-run the System Configuration Wizard to update the webconfig and appconfig files.
6. Restart your machine.

> The CentralConfigAPIKey should be unique, duplicate/multiple keys will break the feature.

Disabling the Centralconfig API Key Feature

Once the installation/upgradation is complete, verify the centralConfig appsettings.json file has: property “DisableApiKeyValidation”:”false” which need to be change to “True” as “DisableApiKeyValidation”:”true”.

1. Log in to the application with Admin credentials and open the Configuration console.
2. Select **Security Controls** > **API Key** and delete the **CentralConfigAPIKey**.

Ensure you **Delete** and not **De-activate** to disable.

Impacted webconfig file list:

AppDomains	Configuration Files
AppServer	Web.config
WorkFlow	WorkflowService.exe.config
Escalation	EscalationService.exe.config
email	EmailWindowsServiceapp.exe.config
bulkupload	KMBulkUploadTool.exe.config
integrationserver	IntegrationServiceHost.exe.config
systemconfigwizard	ServerConfigurationWizard.exe.config
release tool	ReleaseTool.exe.config
Frsloggingservice	FRSLoggingService.exe.config
metric	HEAT.MetricsServer.exe.config
message queue	MessageQueueServiceHost.exe.config
License service	LicenseServerWindowsService.exe.config
report provisioning	reportserver\ReportProvisioning.exe.config
Neurons	appsettings
Teams Bot Service	appsettings
DynamicWebServices	web.config
OpsConsole	web.config
FrsSurvey	web.config
FrsSurveyProxy	web.config
IM	IMServer\IMServices\AssetProcessor\Web.config
IM	IMServer\IMServices\DiscoProcessor\Web.config
IM	IMServer\IMServices\DiscoUtils\Web.config
IM	IMServer\IMServices\IMReadOnlyDataService\Web.config
IM	IMServer\IMServices\TaskProcessor\Web.config
IM	IMServer\WebServices\ClientTransportProcessor\Web.config
reports	C:\Program Files\Microsoft SQL Server\MSRS12.MSSQLSERVER\Reporting Services\ReportServer\Web.config

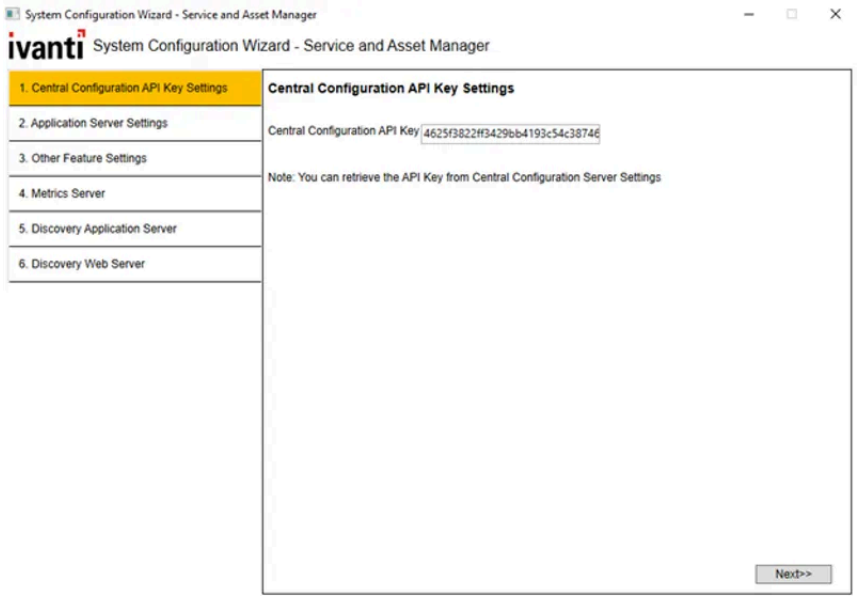
Impacted appsettings.json file of the appdomain

AppDomain
CCF
Appserver
EmailService
EscalationService
FRSSurvey
FRSSurveyProxy
InboundWebService
IntegrationServer
KMImportTool
LicenseManager
LoggingService
MessageQueue
MetricsServer
OpsConsole
ReleaseTool
SystemConfigurationWizard
WorkflowService
Reportserver
IMServer - IMServices
AssetProcessor
DiscoProcessor
DiscoUtils
IMReadOnlyDataService
IvantiCloudDataProcessor
TaskProcessor
WebServices - AgentTaskWs

Custom Installation of SCW

When customers try to install the Configuration Server in one machine and other components in another machine, user has to copy the Central Config API Key from Configuration Server and update the key in the **Central Configuration Settings** page as show in the following image.

Sample image of Custom Installation of SCW


System Configuration Wizard - Service and Asset Manager
ivanti System Configuration Wizard - Service and Asset Manager
1. Central Configuration API Key Settings
2. Application Server Settings
3. Other Feature Settings
4. Metrics Server
5. Discovery Application Server
6. Discovery Web Server
Central Configuration API Key Settings
Central Configuration API Key 4625f3822ff3429bb4193c54c38746
Note: You can retrieve the API Key from Central Configuration Server Settings
Next>>

5.8. Setting up the Teams Bot Integration

Setting up the Central Config API Key

1. Log in to the CentralConfig > **Configure Application** > **Security Controls** > **ApiKeys**.
2. Find the **CentralConfigApiKey** as shown below and copy the value.
3. Paste the copied value in the appsettings.json file for the CentralConfigApiKey. For example, *central-config-api-key"="xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"*

Executing the PowerShell Script (from version 2023.4 onwards)

1. Download the zip file to a relevant folder in your machine.
2. Extract the zip file.
3. Open PowerShell as Administrator.
4. Go to the location where the PowerShell script is located, that is, the location where you extracted the zip file.
5. Run the script - `.\install.ps1` .
6. Provide relevant details to the below prompts:
    - Is the Environment SSL Enabled?
    - Please enter the ISM Tenant Id [example: ismtenant.yourdomain.com] only if environment is SSL enabled, otherwise provide app server IP address or host/domain name.
    - Please enter the Central Config Tenant Id [example: configtenant.yourdomain.com] only if environment is SSL enabled, otherwise provide server IP address or host/domain name.

   The configuration is done.

Executing the PowerShell Script (before version 2023.4)

1. Download the zip file to a relevant folder in your machine.
2. Extract the zip file.
3. Open PowerShell as Administrator.
4. Go to the location where the PowerShell script is located, that is, the location where you extracted the zip file.
5. Run the script - `.\TeamsBot.ps1` .
6. You will be prompted for the Appserver as well as the Central Config ipaddress/ Machine name/ localhost name. name/ localhost. Enter the appropriate Address/Machine name/localhost.

   The configuration is done.

Troubleshooting

Incorrect ipaddress/machine name/local host name (version 2023.4 onwards)

If you provide incorrect IP address/machine name/local host name, open the Appsetting.json file from the folder inetpub >Teams Bot Service verify and replace the value for the following:

- AppServerUri
- CentralConfigUri
- ConfigServiceClientUri

Example of the AppSetting.json file:

{ "JsonMaxDepth": 128, "MicrosoftAppId": "", "MicrosoftAppPassword": "",

"IPCMServiceName": "/ServiceAPI/IPCMService.asmx", "LogLevel": "DEBUG", "CacheTimeout": "30",

"isOnPremise": true, "central-config-api-key": "", "AppServerUri": "https://ism-onprem15-

tenant1.ivanticlouddev.com/HEAT", "NoOfFAQ": 2, "NoOfKnowledgeBase": 3, "Soap": {

"OpenTimeout": "2", "CloseTimeout": "2", "SendTimeout": "2",

"ReceiveTimeout": "2" }, "CentralConfigUri": "https://cfg-onprem15-

ism.ivanticlouddev.com/CentralConfig", "BuildVersion": "", "ConfigServiceClientUri": "https://cfgonprem15-
ism.ivanticlouddev.com/CentralConfig", "ISM_ConfigServiceClientName":

"api/centralconfig", "ConfigServiceClientName": "api/centralconfig", "UseInMemoryState": true,

"SecretStorageUri": "", "RedisStorageConfiguration": { "EnableRedisFlag": true,

"RedisConnectionString": "redis.redis.svc.cluster.local",

"sendmessagetochatservice": "reveivemessagefromteams", "closesocketsignal":

"closesocketsignal" }, "IocServices": [{

"LifetimeType": "1", "AssemblyName": "ServiceManager.SimpleStorageStrategy",

"ClassName": "ServiceManager.SimpleStorageStrategy.AzureBlobStorage",

"InterfaceName": "ServiceManager.SimpleStorageStrategy.ISimpleStorageStrategy" },

{ "LifetimeType": "1", "AssemblyName":

"ServiceManager.SimpleStorageStrategy", "ClassName":

"ServiceManager.SimpleStorageStrategy.SecretStorage.AppSettingSecretKey",

"InterfaceName": "ServiceManager.SimpleStorageStrategy.SecretStorage.ISimpleSecretStorageStrategy"

}, { "LifetimeType": "1", "AssemblyName":

"ServiceManager.AppServer.WebReference", "ClassName":

"ServiceManager.AppServer.WebReference.IPCMService", "InterfaceName":

"ServiceManager.AppServer.WebReference.IIPCMService" }, {

"LifetimeType": "1", "AssemblyName": "ServiceManager.CentralConfig.WebReference",

"ClassName": "ServiceManager.CentralConfig.WebReference.Rest.ConfigServiceClientJson",

"InterfaceName": "CentralConfig.Contract.IConfigServiceAPI" }, {

"LifetimeType": "1", "AssemblyName": "ServiceManager.Cache",

"ClassName": "HEAT.Cache.InMemory.InMemoryCache", "InterfaceName":

"HEAT.Cache.Contract.ICache" }] }

Incorrect ipaddress/machine name/local host name (before version 2023.4)

If you provide incorrect IP address/machine name/local host name, open the Appsetting.json file from the folder inetpub >Teams Bot Service and replace the AppserverIPAddress with the correct one.

Example of the AppSetting.json file:

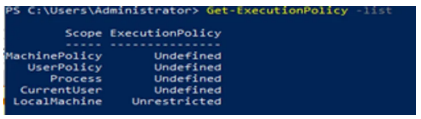
{"MicrosoftAppId"="";"MicrosoftAppPassword"="";"IPCMURL"= "http://$AppserverIPAddress/HEAT/ServiceApi/IPCMService.asmx"; "WorkflowURL"= "http://$AppserverIPAddress/HEAT/ServiceApi/WorkflowService.asmx"; "ConfigURL"= "http://$CentralConfigIPAddress/CentralConfig/ConfigServiceAPI.asmx"; "RetrieveTenantLogLevel_ws_url"= "http=//$CentralConfigIPAddress/CentralConfig/RetrieveTenantLogLevel.ashx"; "EnableCentralLogging"= false; "LoggingService_ws_url"= "http://$AppserverIPAddress/Heat.Logging.Service/api/LoggingService/HeatServiceManagementLogging"; "ElapsedSecondsToFlushLog"= 60; "ItemSizeToFlushLog"= 1000; "LogSettingCacheTimeoutInMinutes"= 5; "SendLogFileLocation"= "C:\logs"; "WriteLogFileLocation"= "C:\logs"; "SendLogWaitInterval"= 300; "WriteLogWaitInterval"= 300; "EnableLogging"= true; "CacheTimeout"= "30"; "isOnPremise": false; ”CentralConfigApiKey”=”” }

Make sure that Boolean values (true/false) don't not have double quotes.

To know the execution policy list

Run the command - `PS C: \Users\Administrator>Get-ExecutionPolicy - List`

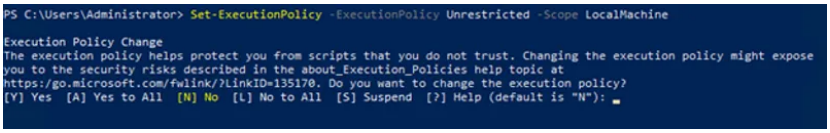
Sample image of the policy list



To set the execution policy

Once the PowerShell script is executed successfully, execute the command - `Set-ExecutionPolicy -ExecutionPolicy “PROVIDE PREVIOUS VALUE” -Scope LocalMachine` to set the execution policy.

Sample image of setting execution policy




```
Bash
```

5.9. Importing Packages with SQL

The feature flag **SQLPackageConfig** in the Feature Management section of the Configuration DB is set to `True` by default from version 2023.2.

When the feature flag is set to `True` - packages containing SQL queries will fail with warning that the SQL statements failed to run.

Solution:

- When importing Neurons for ITSM packages - set the feature flag **SQLPackageConfig** to `False` to avoid any failures. Once the import is done, ensure to set the feature flag **SQLPackageConfig** back to `True`.
- Modify all custom packages which contains SQL queries and remove all references to SQL statements.

Enabling or Disabling the Feature Flag for on-premise customers

1. Open the Configuration DB.
2. Navigate to Feature Management.
3. Search for the feature flag **SQLPackageConfig** and the respective Tenant.
4. Click on the respective record.
5. Toggle the status value to enable/disble the feature flag.

5.10. Enabling the Action Service

The Action Service can only be enabled per tenant.

Configuring Redis for Action Service

To configure Redis for the Action Service, you have to open **appsettings.json** in the Action Service and enter the following:

Bash

```
"RedisConnectionString": "<Redis_Server>:<Port_Number>"

"RedisSettings": {
"ExpirationTimespan": "0.00:15:00",
"ConnectionString": "<Redis_Server>:<Port_Number>,password=<PasswordOfRedisServer>,abortConnect=false"
}
```

For example :

Bash

```
"RedisConnectionString": "host.docker.internal:6379"

"RedisSettings": {
"ExpirationTimespan": "0.00:15:00",
"ConnectionString": "host.docker.internal:6379,password=Password@123,abortConnect=false"
}
```

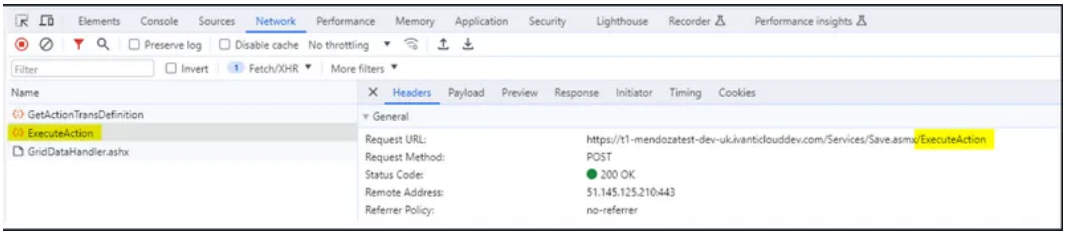
> A Redis server is required for the Action Service. For more details on installing Redis manually, refer **Setting up Redis**.

Enabling Action Service

- Set the **EnableActionService** option in Global Constants as `True`.
- Ensure the **ActionServiceUsage** option is `Disabled` in the Central Config. By default it is **Disabled**.
- Set the **DisableDirectActionServiceUsage** option as `Unconditionally ON` in the Central Config.

Verifying the functioning of Action Service

1. Log in to a Neurons for ITSM tenant.
2. Navigate to the **Incident** workspace and open an existing Incident.
3. Open the chrome console > **Network** > **Headers**.
4. Execute a converted quick action. For example, **OOTB** > **More** > **Form Actions** > **Assigned Incident to Me**.
5. Verify if the below endpoint is being called.



Troubleshooting

Problem

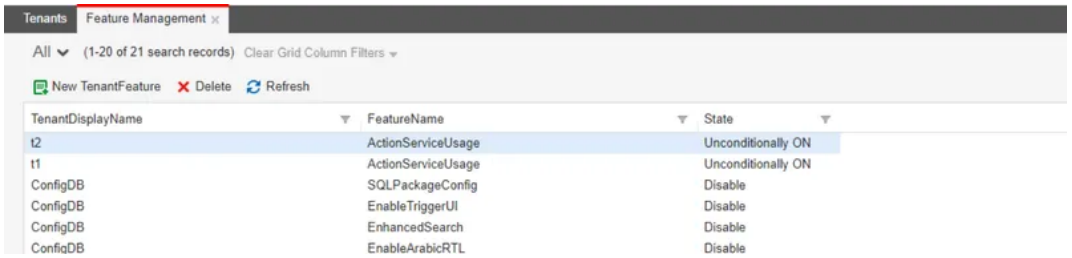
After Upgrading, the Action Service is not working when the installation is using Windows Authentication Option.

Resolution

1. Check **IIS settings** > **Application Pool** > **HeatActionServicePool**.
2. Check **Identity Column** to ensure the **Identify Property** has the same value as all other services.

Feature Management

- **ActionServiceUsage** feature is disabled by default. All non-complex type of actions will be executed within AppServer.
- **ActionServiceUsage** needs to be set as `Unconditionally ON`. All non-Deprecated actions will be executed within Action Service.



- **DisableDirectActionServiceUsage** can be set as `Unconditionally ON` to disable directly calling the Action Service, returning to the old approach of calling the ActionServiceClient through the Save.asmx service.
- **DisableActionServiceForAEM** can be set `Unconditionally ON` to prevent Advanced Email Monitoring from being executed via the Action Service. Actions will then be executed via AppServer.
- **EnableActionServiceForWorkflow** when set to `Unconditionally ON` executes workflow actions via the Action Service. Otherwise workflow actions will be executed via AppServer.

> Do not use this unless you have issues on CoreVM or local.

Condition	Endpoint
Feature flag does not exist in .	ActionService/ExecuteAction
Feature flag exists in ConfigDB but is not used in Feature Management. (Treated as system wide Unconditionally ON).	Services/Save.asmx/Execute Action
Feature flag is set as Disabled.	ActionService/ExecuteAction
Feature flag is set as TrialPeriod	ActionService/ExecuteAction
Feature flag is set as TrialCompleted	Services/Save.asmx/Execute Action
Feature flag is set as Unconditionally ON	Services/Save.asmx/Execute Action

On-premise Configuration

ActionServiceUsage Tenant Feature Management should be set to Disabled for OnPremise environment, until the FrameworkActionsRunner has been deployed.

Services	Settings	Notes
Action Service	ISM_RedisConnectionString	Connection String to Redis deployed on the environment. This should be the same as other applications, Example, localhost:6379,abortConnect=false
	AppServerUri	URI to AppServer deployed on the environment. This should be the same as other applications. Example, http://:/HEAT
	MessageQueueServiceUri	URI to message queue service deployed on the environment. This should be the same as other applications . Example, http://:
	InventoryServiceUri	URI to inventory service uri deployed on the environment. This should be the same as other applications. Example, http://:
	ConfigServiceClientUri	URI to CentralConfig deployed on the environment. This should be the same as other applications. Example, http://:/CentralConfig
	ConfigServiceClientName	CentralConfig client name. This should be the same as other applications. Example, api/centralconfig
	SaaSAppServerBaseURL	URI for SaaS AppServer Base URL deployed on the environment. This should be the same as other applications. Example, http://:/HEAT
	WorkflowServiceUri	URI for the workflow service deployed on the environment. This should be the same as other applications. Example, http://:/HEAT
	SaaSBaseURI	URI for SaaS Base Uri This should be the same as other applications e.g. http://:/HEAT
	ISimpleSecretStorageStrategy	
	central-config-api-key	The central config API key used to validate API calls in the config service.
AppServer.Core	appsettings.json ActionServiceUri web.config ActionServiceUri	http://:/ActionService/ActionService /ActionService/ActionService is required to be duplicated. The first part denotes the application name hosted in IIS, the second denotes the endpoint prefix in the service itself.
Email.Core	ActionServiceUri	http://:/ActionService/ActionService /ActionService/ActionService is required to be duplicated. The first part denotes the application name hosted in IIS, the second denotes the endpoint prefix in the service itself.
WorkflowService.Core	ActionServiceUri	http://:/ActionService/ActionService /ActionService/ActionService is required to be duplicated. The first part denotes the application name hosted in IIS, the second denotes the endpoint prefix in the service itself.
MetricsService	ActionServiceUri	http://:/ActionService/ActionService /ActionService/ActionService is required to be duplicated. The first part denotes the application name hosted in IIS, the second denotes the endpoint prefix in the service itself.

5.11. Workspaces that Can Be Used

There are very few workspaces in the ConfigDB that can be used by administrators. See the following topics for information:

- Configuring the Message Queue
- Configuring the Message Queue Handler
- Configuring Email
- Viewing All Logs
- Working With Customer Contacts
- Working With Tenants
- Configuring Manual Load Partitioning

5.11.1. Configuring the Message Queue

Use the **Message Queue Configuration** workspace to create and configure a message queue.

Follow these steps:

1. Log in to the ConfigDB.
2. From the workspace selector bar, select Message Queue Configuration. The system displays the **Message Queue Configuration** workspace.
3. Enter information into the fields.

Parameter	Description
Polling Interval (sec)	Specifies how often to check the message queue. The default value is 120.
Processor Threads	Specifies the number of threads. The value depends on the specifications (such as memory and CPU) of your system. The default value is 250.
Metadata Refresh Interval (min)	Specifies the interval for which to refresh the metadata pertaining to the tenant, which is the amount of time for which to retain the cache. After this amount of time, the system purges the cache and reloads it. The default value is 10.

4. Click Save.

5.11.2. Configuring the Message Queue Handler

Use the **Message Queue Handler** workspace to configure the message queue handler.

Follow these steps:

1. Log in to the ConfigDB.
2. From the workspace selector bar, select Message Queue Handler. The system displays the **Message Queue Handler** workspace.
3. Enter information into the fields.

Parameter	Description
Name	Name of the message queue handler web service.
Endpoint	The URL of the message queue handler web service to which queued tasks are dispatched. An example is http://IP_address/IntegrationService.svc.
Dispatch Method	The name of the web method that is implemented as part of the handler web service (which the message queue invokes to dispatch tasks). For example, the Neurons for ITSM XSLT email handler implements a web method called HandleMessage. NOTE: The name of the dispatch web method is case sensitive.
Batch Size	The maximum number of tasks that the message queue can dispatch to a message queue handler.
Active	Specifies if the message queue handler is on or off.
Priority	Specifies the order of this handler, in relation to any other handlers that are defined.
Process Timeout Interval (minutes)	Amount of time to wait for a response from the message queue handler before logging an error or before retrying, if you have not exceeded the number of attempts specified by the **Max number of attempts** parameter.
Include Data with Dispatch	Specifies whether to include the data when this dispatch web method is called. If you do not check this option, the message queue handler must call the dispatch web method to get the data.
Max number of Attempts	The maximum number of times to resend the data before logging an error.
Archive on Completion	Specifies whether to archive the message queue journal, which contains the tasks that the message queue handler has processed. Use this information for troubleshooting purposes. Enabled by default.
Purge Archive after (days)	Set to 30 days by default. The amount of days after which the archive is purged.
Description	A description of the message queue handler.

4. Click Save.

5.11.3. Configuring Email

Use the **Email Configuration** workspace to configure the inboxes for email.

Follow these steps:

1. Log in to the ConfigDB.
2. From the workspace selector bar, select Email Configuration. The system displays the **Service Provider Email Configuration** workspace.
3. Enter information into the fields.

Field	Description
Message Queue Handler Name	The name of the message queue handler to use. This must be the same name entered in the **Name** field in the **Message Queue Handler** workspace. See Configuring the Message Queue Handler.
Mailbox Poll Interval	The amount of time, in seconds, after which email configuration changes take affect.

4. Click Save.

5.11.4. Viewing All Logs

If there are any problems with the ConfigDB, such as email issues or workflow errors, they are reported in the logs. Review the logs to determine the underlying issue and to help resolve it.

- Setting the Log Level
- Viewing the Logs

Setting the Log Level

You can set the amount and detail of information in the logs by adjusting the Log Level. You can do this by going to the **Logging Configuration** workspace. See the Neurons for ITSM Online Help.

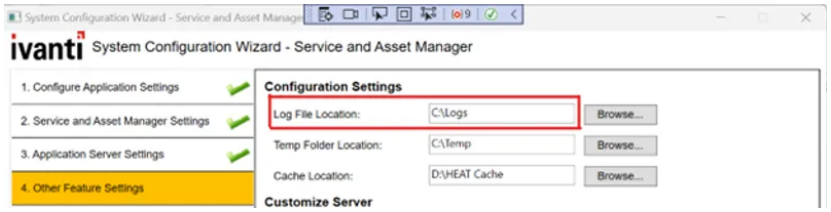
Configuring the Logs

All the logs related to the app domain will be stored in the following locations:

Configurable Logs

- In the step 4 of the System Configuration Wizard, enter the file location in the Log File Location.

  All the configurable logs pertaining to Configuration Wizard and AppServer will be stored in this location.



Non-Configurable Logs

These are logs that depend on the parameters specified in the **nlog.config** file or other AppDomain.

For example, in the **EscalationService appdomain**, go to the path - *HEAT Software|HEAT|EscalationService*, check the **nlog.config** file where the configuration of logs is done in the target node with the name - **logfile** as shown in the image below:

```
<nlog xmlns="http://www.nlog-project.org/schemas/NLog.xsd"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

  <targets>
    <target name="console" type="Console" layout="${longdate}|${level:uppercase=true}|${logger}|${message}" />
    <target name="debugger" type="Debugger" layout="${longdate}|${level:uppercase=true}|${logger}|${message}" />
    <target name="logfile" type="File" fileName="/ism/logs/escalationsvc/${machinename}.ism.esc.svc.${date:format=MM-dd-yyyy-hh}.log"
  </targets>
  <rules>
    <logger name="*" minlevel="Debug" writeTo="console,debugger,logfile" />
  </rules>
</nlog>
```

For all other AppDomains listed below, the logs will be available in the following format - **C:\ism\logs\esclationssvc**.

- ActionService
- CentralConfiguration
- EmailService
- EscalationService
- FRSSurveyProxy
- FRSSurvey

- OpsConsole
- IntegrationServer
- MessageQueue
- InboundWebService
- LoggingService
- MatricsServer
- WorkFlowService
- AssetProcessor
- DiscoProcessor
- IMReadOnlyDataService
- IvantiCloudDataProcessor
- TaskProcessor
- WebServices - AgentTaskWs

Viewing the Logs

Follow these steps to view all of the logs associated with the ConfigDB:

1. Log in to the ConfigDB.
2. From the workspace selector bar, select **All Logs**. The system displays the **LogsAll** workspace with the following information:

Field	Description
Log Entry Id	A unique ID for this log entry.
Log Time	The time at which the log entry was originally created on the server.
Log DB Time	The time when the log was uploaded into the database.
Client IP Address	The IP address of the service that generated the log, such as the workflow server.
SubSystem Id	The subsystem of the service that generated the log. (Each module has predefined subsystems. For example, for the email server, polling is a subsystem.)
Login Id	The login ID for the user who experienced the log event.
Tenant Id	The tenant instance that is experiencing the log event.
Error Code	The category of the error. An example is "trigger not found".
Current Role	The role assigned to the user if the error is generated by a user. If the error is generated by the system, this field is empty.
Host Name	The machine that is reporting the error.
Thread Name	Internal thread ID. Not applicable.

3. Double-click any log entry to view the log details.
4. To refresh the data, click **Refresh**.

Non-configurable logs that is logs that depend on the parameters specified in the **nlog.config** file are stored in the following path - HEAT Software\HEAT\EscalationService.

5.11.5. Working With Customer Contacts

This section contains the following topics:

- About this Workspace
- Viewing Customer Contacts
- Adding a Customer Contact
- Editing a Customer Contact
- Deleting a Customer Contact

About this Workspace

> This workspace is for deployments with multi-tenant configurations.

You can add, edit, or delete a primary contact for a particular tenant. This is the person you contact for all communications regarding the particular tenant.

This information is also listed in the **Account Information** workspace in the Neurons for ITSM Configuration Console.

Viewing Customer Contacts

Follow these steps to view information associated with customer contacts:

1. Log in to the ConfigDB.
2. From the workspace selector bar, select Customer Contacts. The system displays the **Contacts** workspace with the following information:

Field	Description
Login ID	The login ID for the customer contact.
Full Name	The full name of the customer contact.
Primary Email	The primary email address for the customer contact.
Primary Phone	The primary phone number for the customer contact.

3. Double-click any entry to view the details.
4. To refresh the data, click the refresh icon.

Adding a Customer Contact

1. Log in to the ConfigDB.
2. From the workspace selector bar, select Customer Contacts. The system displays the **Contacts** workspace.
3. Click New Contacts. The system displays the **Contacts** page.
4. Enter information into the fields.
5. Click Save.

Editing a Customer Contact

1. Log in to the ConfigDB.
2. From the workspace selector bar, select Customer Contacts. The system displays the **Contacts** workspace.
3. Double-click the entry to edit.
4. Change the entry.
5. Click Save.

Deleting a Customer Contact

1. Log in to the ConfigDB.
2. From the workspace selector bar, select Customer Contacts. The system displays the **Contacts** workspace.
3. Highlight the entry to delete.
4. If there are dependencies on other items, the system display a confirmation message. Click Continue.

5.11.6. Working With Tenants

About the Tenants Workspace

> The information on this page is populated by the System Configuration Wizard.

Although there is a lot of information in this workspace, you should only do the following tasks from within this workspace:

- Reset the cache for a tenant. See **Resetting the Cache for a Tenant**.
- Add a certificate. If you want to use SAML authentication, you must upload a certificate and password so that end users can download it. See **Adding a Certificate to a Tenant**.
- Capture audit files from ISM Discovery, to help with troubleshooting. See **Capturing Discovery Messages for Debugging**.
- Configure how users attach files. See **Configuring Attachment Options**.
- Set metadata to read-only. See **Setting Production Metadata to Read-Only**.
- Set the login URL. See **Setting the Login URL**.

Resetting the Cache for a Tenant

Neurons for ITSM caches information about each tenant in the ConfigDB. Because of this, changes to the tenant, such as updating the status, database information, and so on, are not immediately reflected in the corresponding tenant. This can lead to situations where Neurons for ITSM allows a user to log into a tenant, even though the status of the corresponding tenant in the ConfigDB is set to closed.

To manually reset the cache for a specific tenant, do the following:

1. Log in to the ConfigDB.
2. From the workspace selector bar, select Tenants. The system displays the **Tenants** workspace.
3. Highlight the tenant to reset the cache for.
4. Click Reset Tenant Cache.

Adding a Certificate to a Tenant

> If your implementation includes multiple tenants, the certificate should be for the domain and not the specific tenant. For example, the certificate for Cloud tenants is for *.saasit.com, not for a specific tenant such as *mytenant*.saasit.com.

1. Log in to the ConfigDB.
2. From the workspace selector bar, select Tenants. The system displays the **Tenants** workspace.
3. Double-click the tenant for which to add a certificate.
4. Click Add Certificate.
5. Navigate to the certificate location. Highlight it and click Open.
6. Enter the password for the certificate in the **Certificate Password** field.
7. Click Save.

Capturing Discovery Messages for Debugging

Perform the following procedure if you are using Discovery and you need to capture the audit files sent to the server from the Discovery client agents. After you capture the audit files, you can review them in a debug environment to find out more details about the problem. The logging can be very long, so be careful about using this feature. For more information, see the Neurons for ITSM online help.

8. Log in to the ConfigDB.
9. From the workspace selector bar, select Tenants. The system displays the **Tenants** workspace.
10. Double-click the tenant to configure logging for.
11. Check Log Failed IM Message.
12. Click Save.

Configuring Attachment Options

Perform the following procedure to configure how attachments are saved. You may need to change the format for storing files if users regularly store very large files, such as system log files. After you configure this, when a user saves an attachment to a record, the attachment is stored in the location that you specified here. For information about saving attachments as a file stream, see https://msdn.microsoft.com/en-us/library/gg471497.aspx.

We recommend that for system data, such as icons and images, you select Database for the **Attachment Save Type** field.

1. Log in to the ConfigDB.
2. From the workspace selector bar, select Tenants. The system displays the **Tenants** workspace.
3. Double-click the tenant for which to configure attachment options.
4. For the **Attachment Save Type** field, select a type: **Database**, **FILESTREAM**, **File system**, **Amazon S3**, or **Azure Blob**.
5. For the **Attachment Path** field, enter the default path where attachments are saved.
6. Click Save.

Setting Production Metadata to Read-Only

If a user makes a change to the metadata in the production instance of the tenant, such as when configuring a workflow for a request offering, but does not make the same change in the UAT or staging instance of the tenant and then pushes the data from the UAT or Staging instance of the tenant to the production instance of the tenant, the data becomes corrupt and the workflow fails.

To avoid this, set the metadata on the production instance of the tenant to read-only. Follow these steps:

1. Log in to the ConfigDB.
2. From the workspace selector bar, select Tenants. The system displays the **Tenants** workspace.
3. Double-click the tenant for which to set the production metadata to read-only.
4. Check **Is Production Metadata Read-only**.
5. Click Save.

If a user tries to edit the workflow for a request offering, the system displays a message stating that the production metadata is read only and cannot be edited. See "Working with Request Offerings" in the Neurons for ITSM online help for more information.

Setting the Login URL

Set the login URL in the following scenarios:

- When configuring Neurons for ITSM Active Directory authentication using Windows Integrated Security (WIS). This feature allows users to access Neurons for ITSM without entering their user name and password. This procedure is only valid for the on-premises version of Neurons for ITSM.
- When configuring the Neurons for ITSM on-premises URL for mobile users.

Follow these steps:

1. Log in to the ConfigDB.
2. From the workspace selector bar, select Tenants. The system displays the **Tenants** workspace.
3. Double-click the tenant for which to set the login URL.
4. In the **Login Url** field, enter the URL where users go to log in.
5. Click Save.

Setting the Alternate Login URL (Vanity URL) for an MSP

Besides setting the login URL as described above, you can also specify an alternate login URL, also known as a vanity URL, for a Managed Service Provider (MSP).

Follow these steps:

1. Log in to the ConfigDB.
2. From the workspace selector bar, select Tenants. The system displays the **Tenants** workspace.
3. Double-click the tenant for which to set the alternate login URL (vanity URL).
4. At the bottom, click the **Alternate Login URL** tab.
5. Click New TenantUrlAlias. The system displays the **New TenantUrlAlias** dialog box.
6. In the **Url** field, enter the alternate login URL (vanity URL).
7. Click Save.

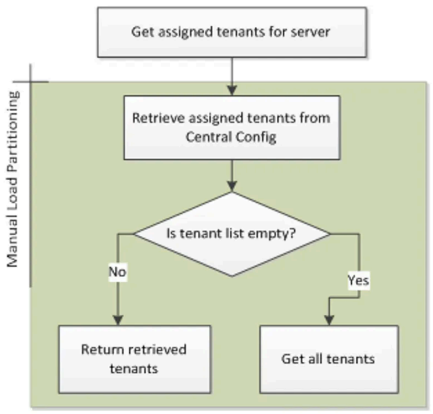
5.11.7. Configuring Manual Load Partitioning

About Manual Load Partitioning

Use the **Service Server** workspace to configure the manual load partitioning feature. The manual load partitioning feature allows you to allocate the load between workflow or email service instances based on your knowledge of the tenant load and server capacity.

When you start up the services, the workflow and email services retrieve the list of assigned tenants. The services only process the tasks for the assigned tenants. The system updates the list of assigned tenants in predefined time intervals. If the list of tenants is empty, the system falls back to the default behavior of processing all of the tenants.

Email Load Balancing Flowchart



> For the email service and workflow service, all load partitioning is done manually. Changes are configured in the **appsettings.json** file at the binary folder. By default **EnableLoadPartitioning** key remains false in the **appsettings.json**, enabling it disrupts the whole Workflow Service.

Assigning Tenants to the Workflow or Email Service

When you configure workflow or email service servers, ensure that at least one server is defined as the "catch all" server. A "catch all" server manages all of the live tenants, even if it does not have any tenants associated with it. Each host should have at least one "catch all" server. This server processes workflow and email events for all tenants.

For each server, you must enable either the email service, the workflow service, or both. If you do not enable a service on the server but the service is running, the service gets an empty tenant list and it runs as a "catch all" server.

If a service instance crashes or stops, the system deletes the partition details from the database and no other instance will pick these tenants except for the "catch all" server.

1. Log in to the ConfigDB.
2. From the workspace selector bar, click More....
3. Select Service Server. The system displays the **Service Server** workspace.
4. Click New Service Server.
5. Enter the host name in the **Host Name** field.
6. If the email service will run on this server, check Is Email Service Enabled.

7. If the workflow service will run on this server, check Is Workflow Service Enabled.
8. If the server is a catch all server, check Is Catch All.
9. If the server is not a catch all server, do the following:

   a. Click the **Tenants** tab.

   b. Click Link.

   c. Highlight a tenant from the list and click Select.
10. Click Save.

Viewing the Results of the Manual Load Partitioning

You can view the results of the manual load partitioning in one of two ways:

- Viewing the INFO log for a service. See **Setting the Log Level for the Workflow or Email Service** and **Viewing the Workflow or Email Manual Load Partitioning**.
- Viewing a report of the load partitioning through Microsoft SQL. See **Viewing a Report of the Manual Load Partitioning**.

Setting the Log Level for the Workflow or Email Service

To configure the logs, do the following:

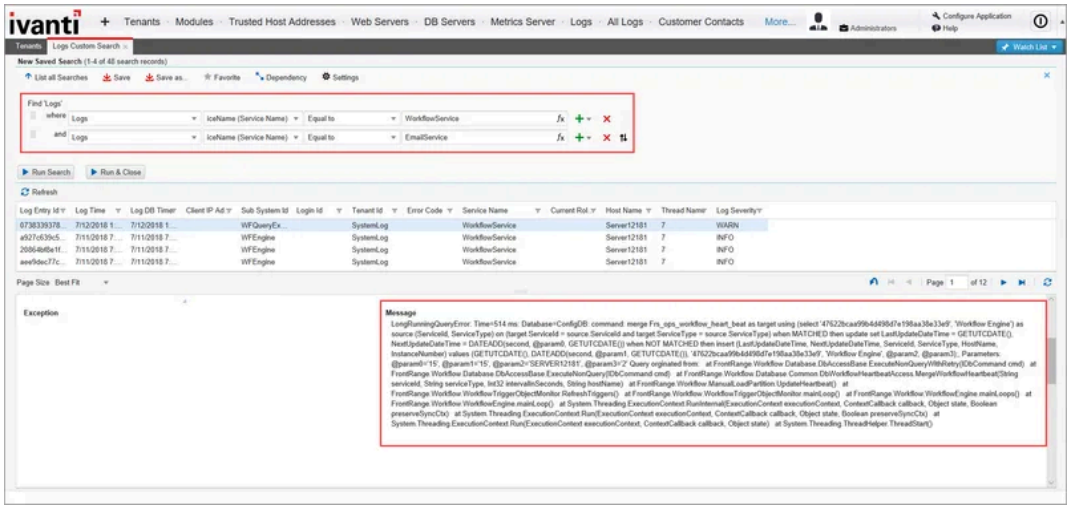
1. Log in to the ConfigDB.
2. Open the **Logging Configuration** workspace.
3. Open the **WorkflowService** service.
4. Change the value of the **Log Level** field to INFO.
5. Open the **EmailService** service.
6. Change the value of the **Log Level** field to INFO.
7. Click Save.

Viewing the Workflow or Email Manual Load Partitioning

After you configure the logs, to view the results, do the following:

1. Log into the ConfigDB.
2. Open the **Logs** workspace.
3. Use a saved search to show all logs with a service name of WorkflowService or a service name of EmailService.

   Logs Saved Search



4. Verify that the tenants listed under the current assigned tenant list match the linked tenants for this server.

Viewing a Report of the Manual Load Partitioning

The workflow or email service retrieves the assigned tenants and updates the **Frs_ops_workflow_heart_beat** and **Frs_ops_workflow_partition** tables in the ConfigDB.

Before you begin, start the workflow or email service. Keep the current session.

Run the following query in Microsoft SQL on the ConfigDB to see the details of the services that are assigned to each tenant:

SELECT ServiceType, HostName, InstanceNumber, TenantId

FROM Frs_ops_workflow_heart_beat heartbeat

JOIN Frs_ops_workflow_partition partition

ON heartbeat.ServiceId = partition.ServiceId

The following is an example of the partitioning details:

Service Type	Host Name	Instance Number	Tenant ID
Workflow Engine	CA-L02552	1	ConfigDB, ITSM_7_Daily_Build
Email Engine	CA-L02552	1	ConfigDB, ITSM_7_Daily_Build

5.11.8. Firebase Configuration for Mobile Push Notification

A new workspace - **FCM Configuration** is added to ConfigDB. The following are its sections:

- **Project Id**: Project Id of the firebase which we going to configure in future step. This Id is used by the server to send the push notification.
- **Server Key**: Unique key that is present inside the firebase project. This key is used by the Server to send the push notification.

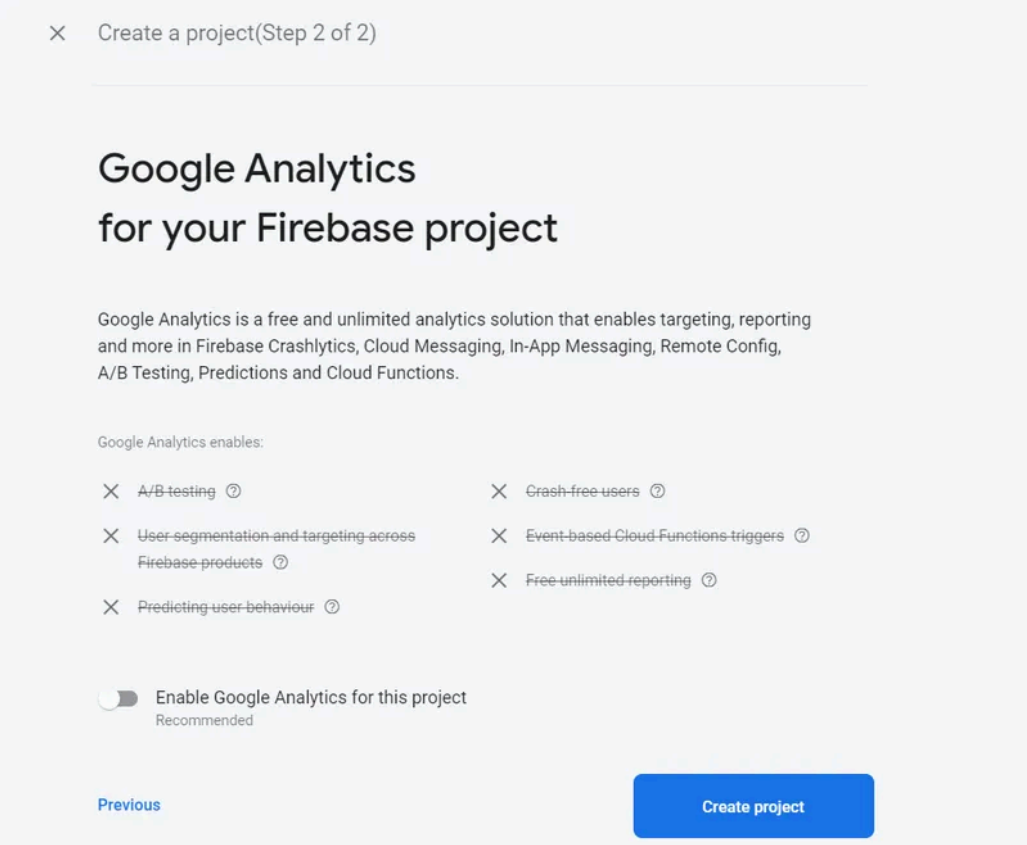
  > This section is deprecated and will be removed in the upcoming release. Therefore, meantime, we recommend giving a dummy value. For example, 100021050040.

- **Client configuration for Android**: json configuration which is used by android mobile to generate FCM token. This configuration will be present in Firebase project we need to add in config Database.
- **Client configuration for iOS**: Configuration which used by iOS device to generate FCM token.
- **Server Settings**: Use to authenticate firebase service account.

The following steps should be performed for Mobile Push Notification to work:

Creating a new Project in Firebase

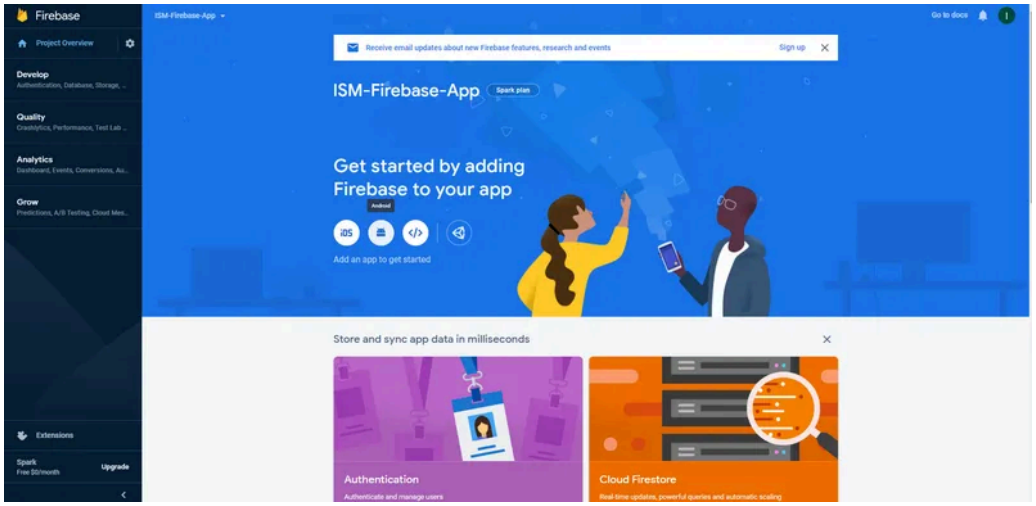
1. Navigate to https://firebase.google.com/ to create a new project. Since Firebase is from Google, you need to sign into your Gmail account so the Firebase project is associated with your Gmail account.
2. Navigate to firebase.com and click **Go to console** on the top right corner of the page.
3. Click the **Create a project** tile (for new Users) or **Add Project** (for existing users) in the Firebase.
4. Type the project name, select the **I accept the firebase terms** check box and click **Continue**.
5. Disable Google Analytics and click **Create Project**.



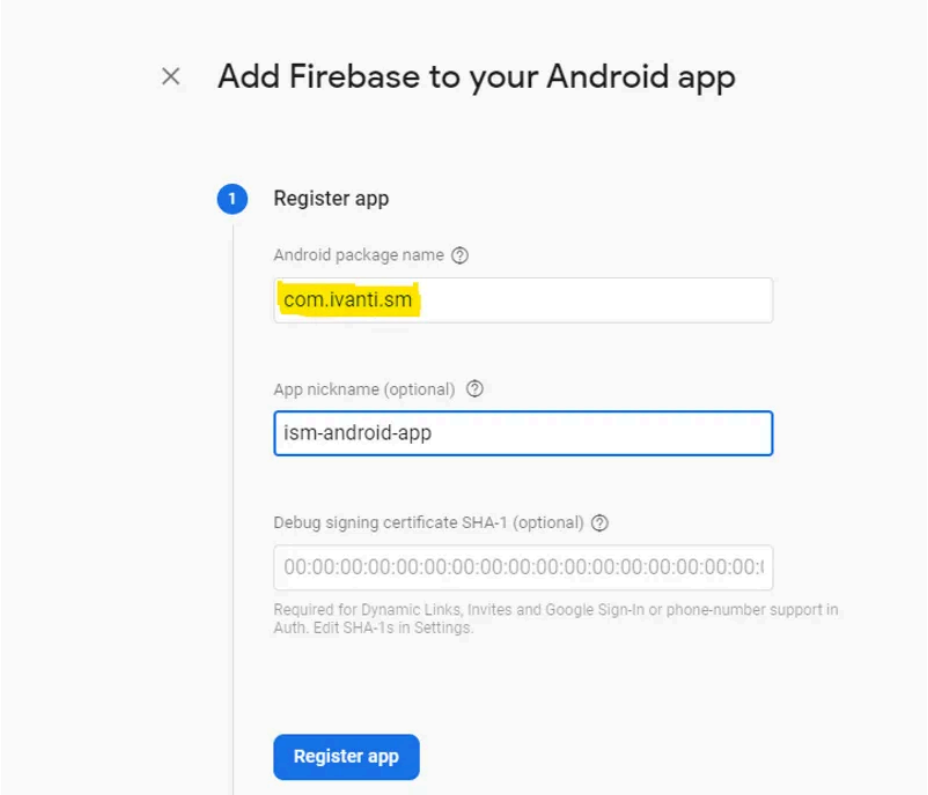
Configure iOS and Android app in Firebase

In this step, you will create iOS and Android app in the firebase project.

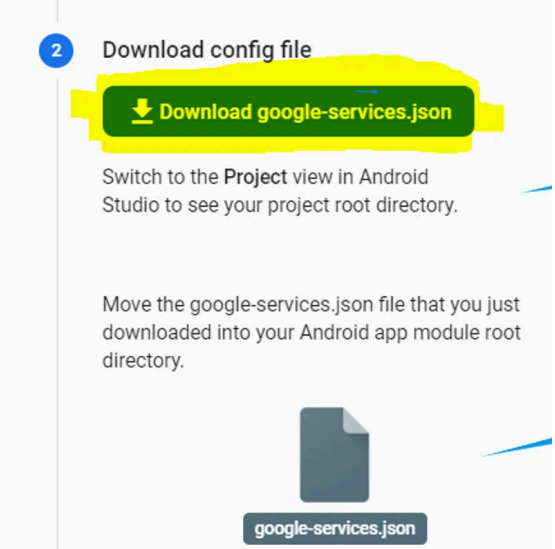
1. Navigate to the newly created project.
2. Click on the **Android** icon to create new Android app.



3. Enter the name of the package - it should always be **com.ivanti.sm**" in the **Android Package name** and enter any name in **App Nickname**. Ensure the package name is right and click on **Register app**.

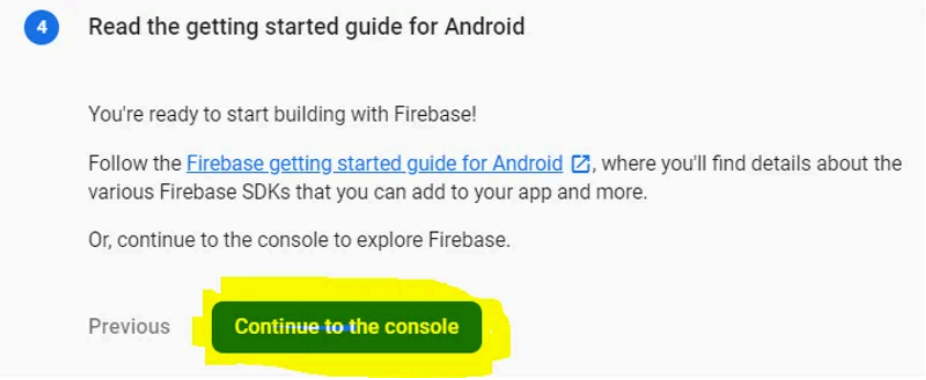


4. Click on **download google.service.json** which is json configuration for Android. Download this file and click **Next**.



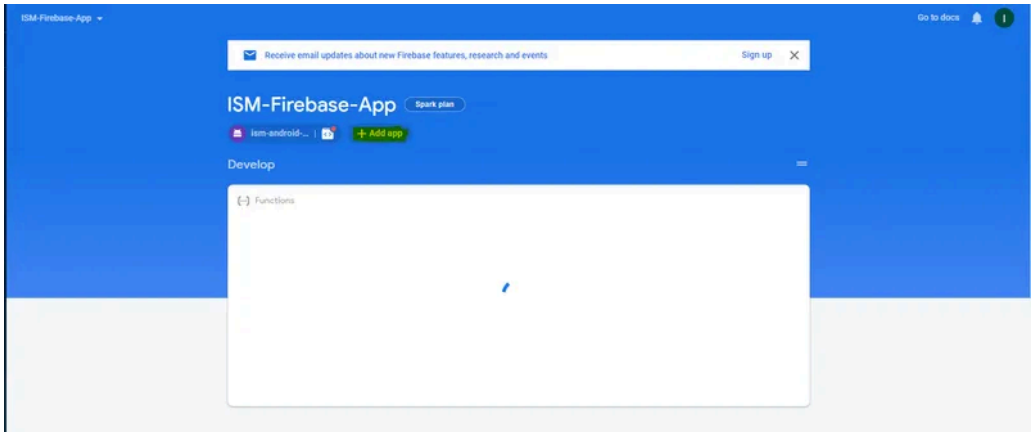
5. Skip Step 3 by clicking on **Next – (Add Firebase SDK)**
6. Click **Continue to the console**.

> If you see "**Run your app to verify installation**" step then you can click on "**Skip this step**" and then click **Continue to console**.

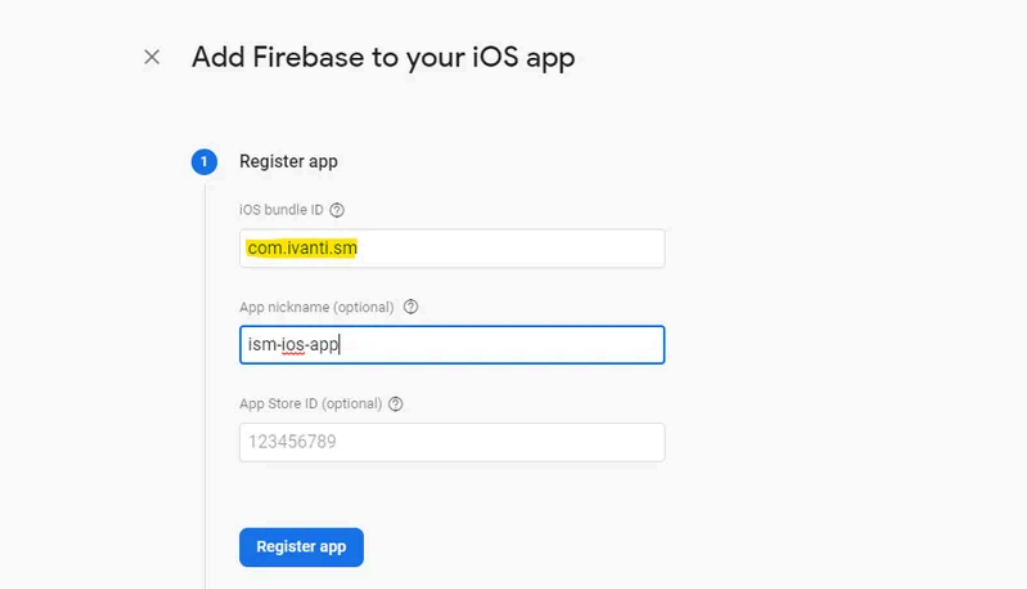


Once all these steps are completed, you have the Android app ready in the Firebase project.

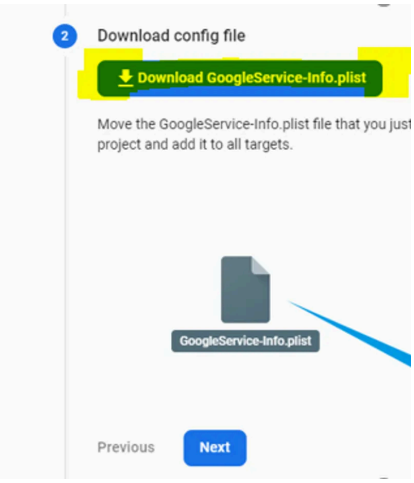
7. Now, do the same for iOS.
8. Click on **Add app** again in the Firebase project and select iOS.



9. Specify the same package name **com.ivanti.sm in** the iOS bundle ID, enter the **App nickname** and click on **Register app**.



10. Click on **Download GoogleServiceInfo.plist** which contains the configuration for iOS device. and click **Next**.



11. Skip Step 3 by clicking on **Next – (Add Firebase SDK)**.
12. Skip Step 4 by clicking on **Next – (Add Initialization code)**.
13. Click **Continue to Console** in step 5.

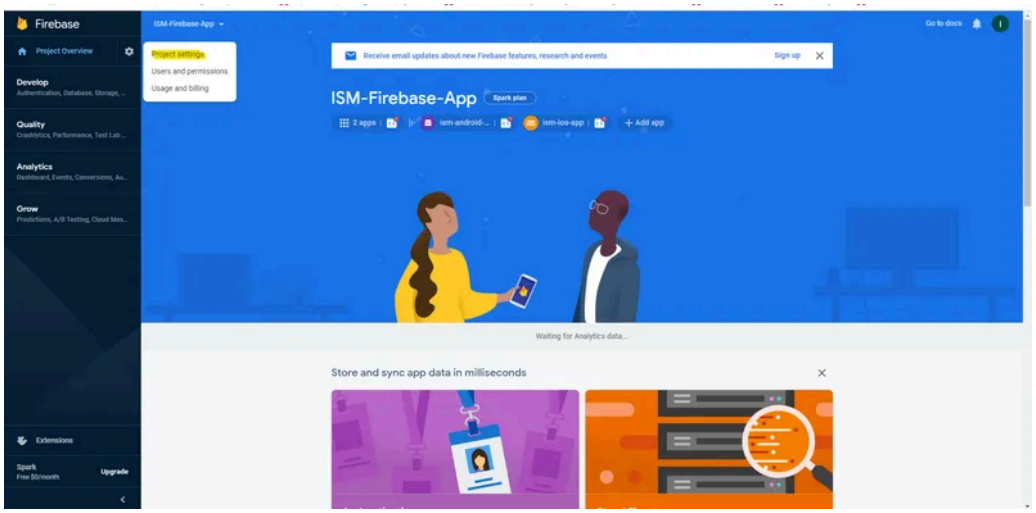
> If you see Run your app to verify installation step, click **Skip this step** and then click **Continue to console**.

Once these steps are completed, you have iOS and Android app configuration is ready in the Firebase.

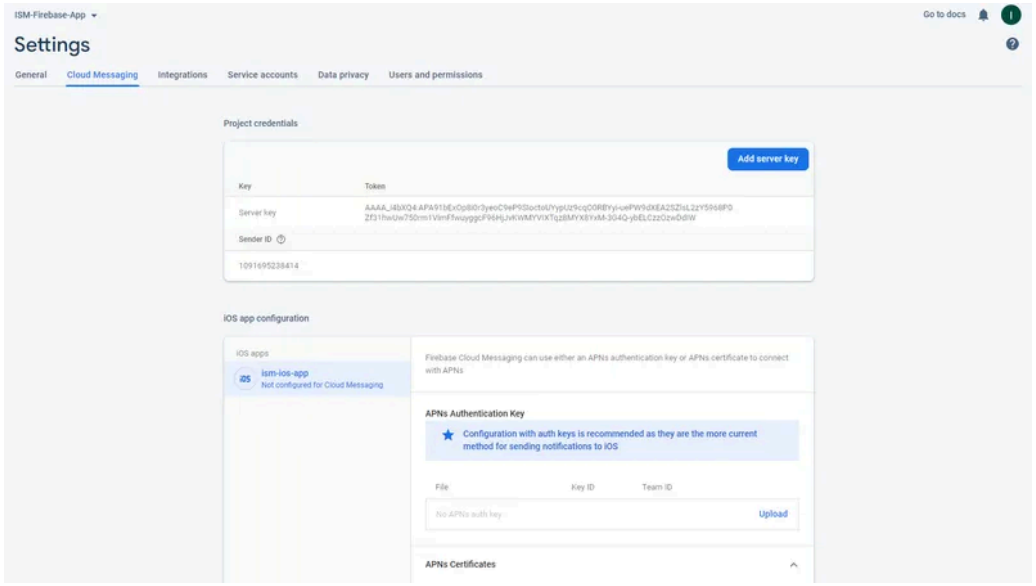
Apply iOS Certificate for iOS app

Ivanti will provide the APNs Certificate for iOS app. You can find the certificate at Ivanti Market Place.

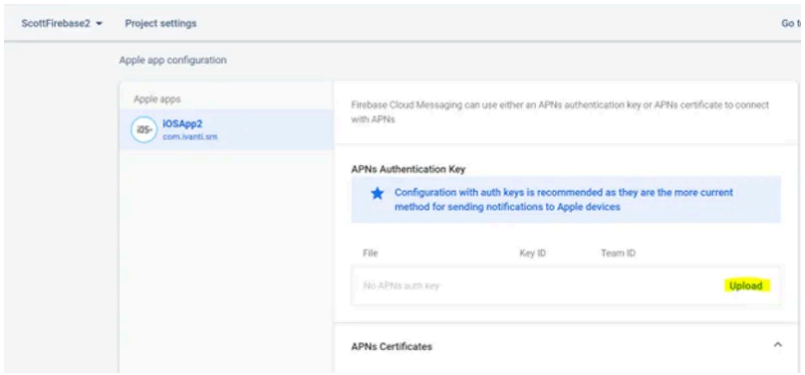
1. Click the **Setting** button from top left and click on **Project settings**.



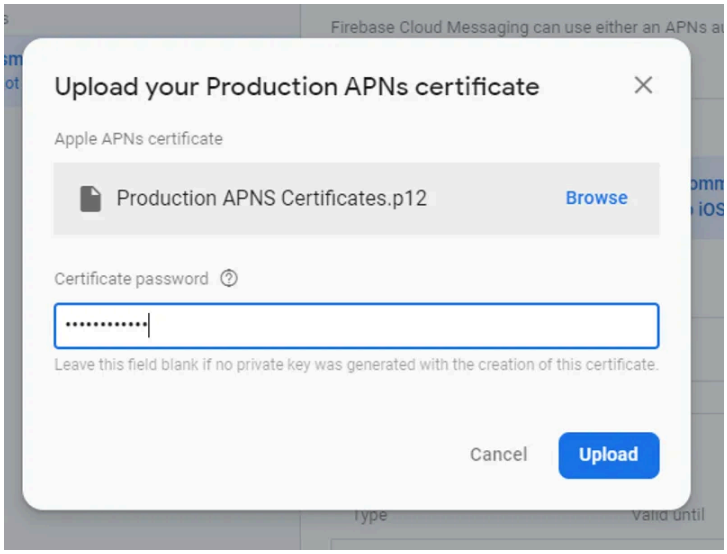
2. Click the **Cloud Messaging** tab.



3. Scroll down to the **APNS Authentication Key** section and click the **Upload** button.



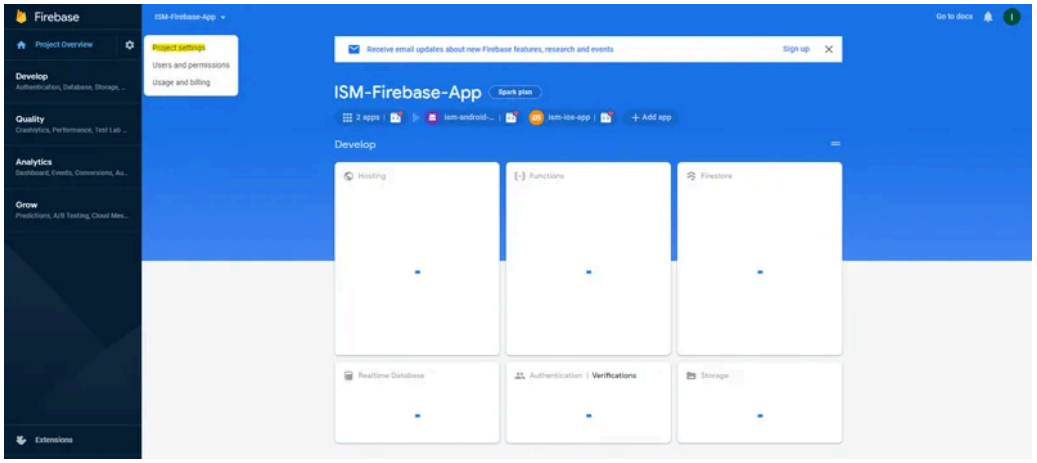
4. Browse to the AuthKey_V65XKX7498.p8 file.



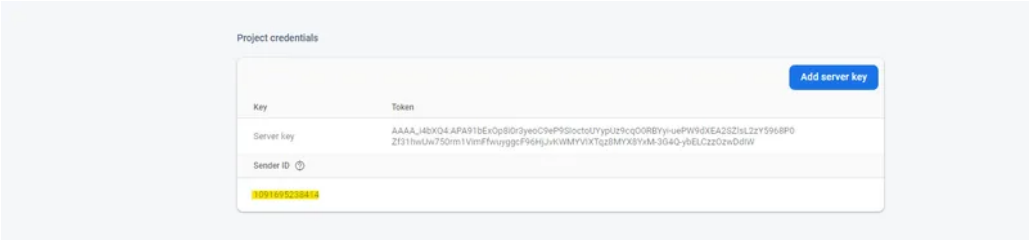
5. Enter the following details:
    - Key ID: V65XKX7498
    - Team ID: B7FW4Y3UP6

Configure Firebase in Config Database of ISM

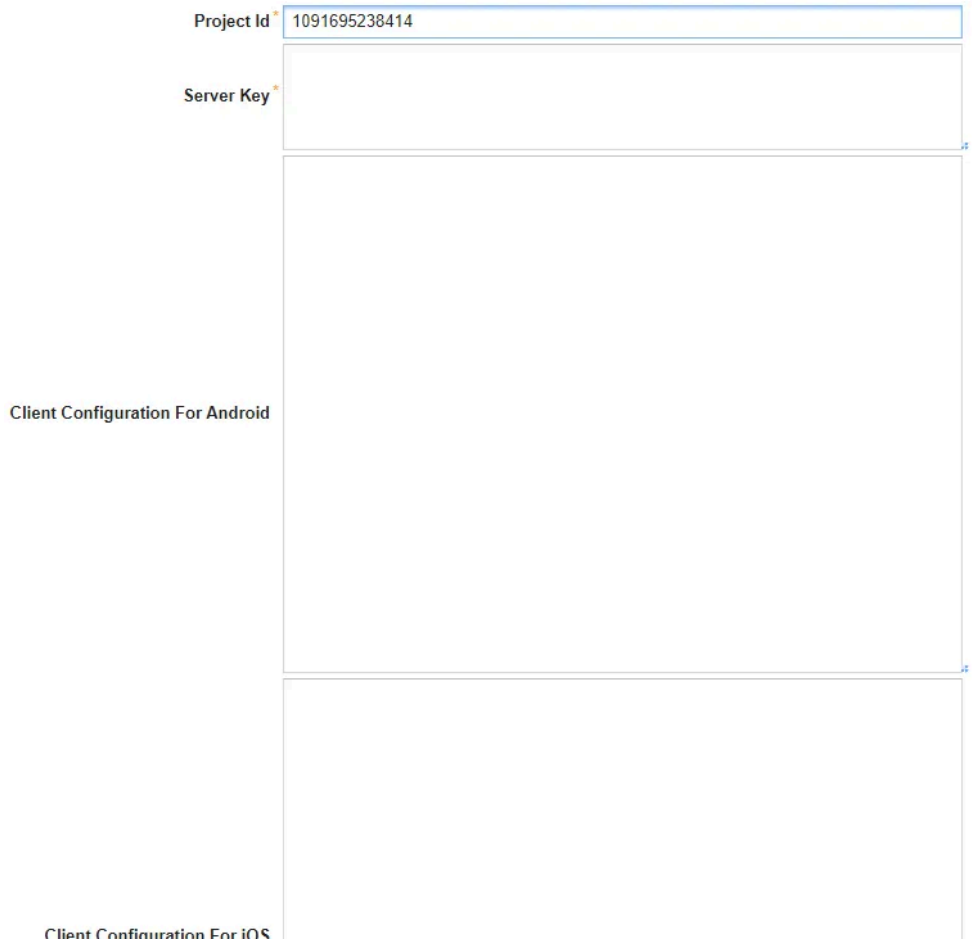
1. Navigate to **Project settings** from the main page of Firebase project.



2. Navigate Cloud Messaging Copy the Sender ID.

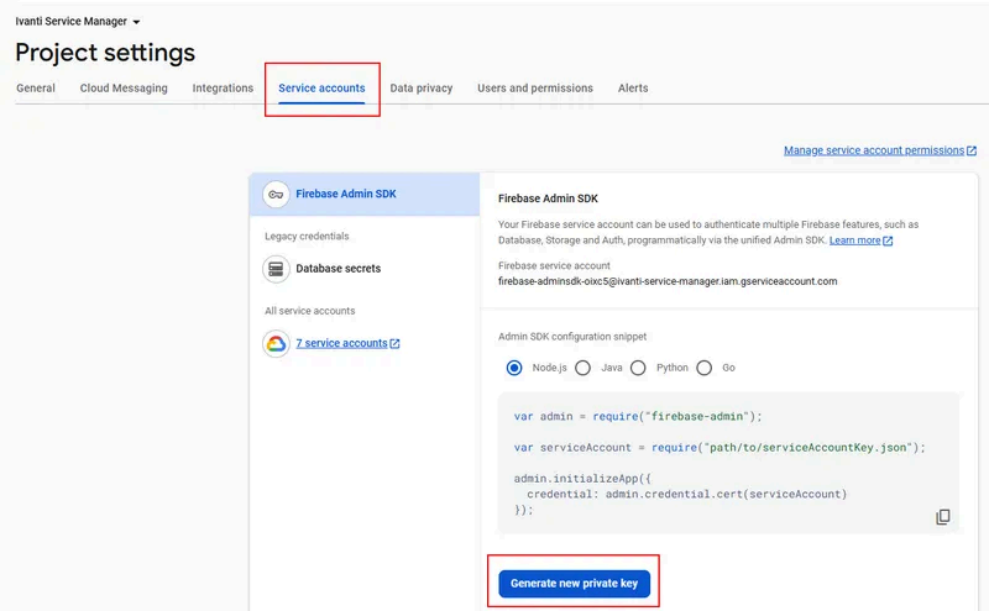


3. Go to **ISM CentralConfig**, open **FCM Configuration** workspace and paste **Sender ID** which is copied to the **Project ID** in workspace.



4. Get **Server Settings** from the **Service accounts** tab.

   a. Under the **Project settings** section, go to **Service accounts** > **Firebase Admin SDK**, and click **Generate new private key**.

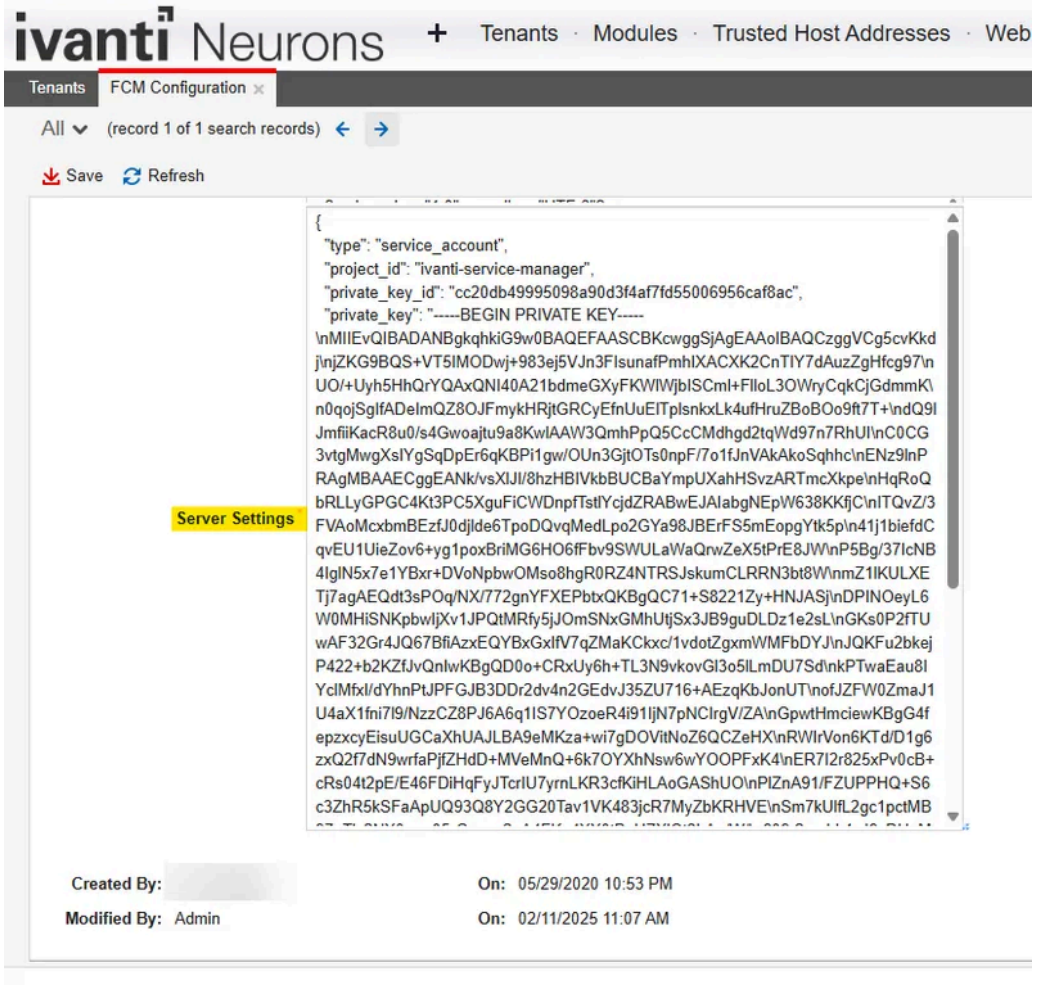


A dialog box appears.

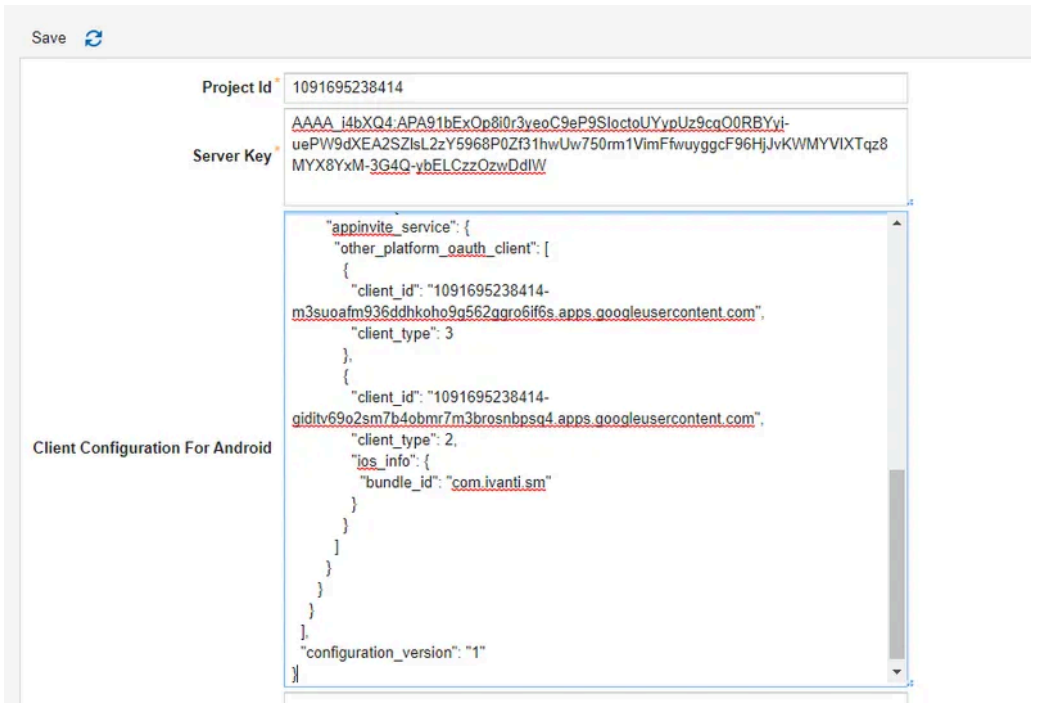
b. Click **Generate key**.

It will download a JSON file.

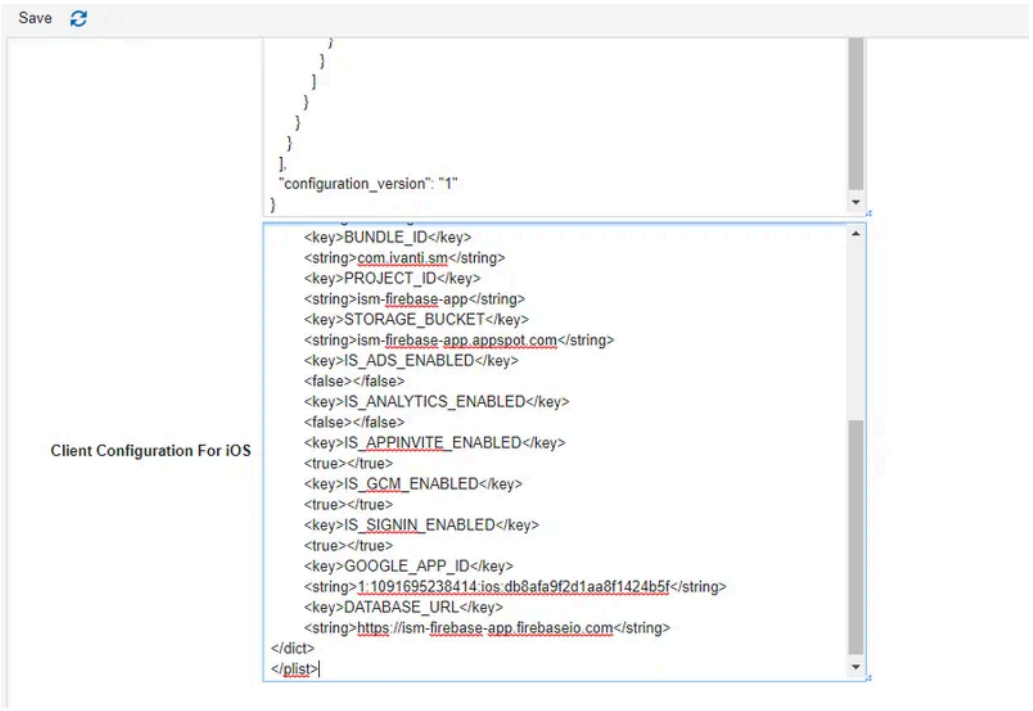
5. Copy the server setting from the JSON file and paste it in the **Server Settings** box in the **FCM Configuration** workspace.



6. Copy the entire content from **google-service.json** which is downloaded from Step – 3 and paste in the **Client Configuration** for Android.



7. Finally, you need to add the **iOS FCM configuration** in **config DB**. Copy the entire content from **GoogleService-Info.plist** which is downloaded in step 3 and paste it in the **Client Configuration for iOS** and **Save**.



5.11.9. Decommissioning FTP and File on Shared folder Integration

Currently, ITSM customers can import data from FTP and Shared Path locations. Since these sources pose a security threat to the cloud environment, restrictions are added to the use of these options. These restrictions are feature flagged which can enable or disable the options available to an end user either while importing or exporting data to ITSM.

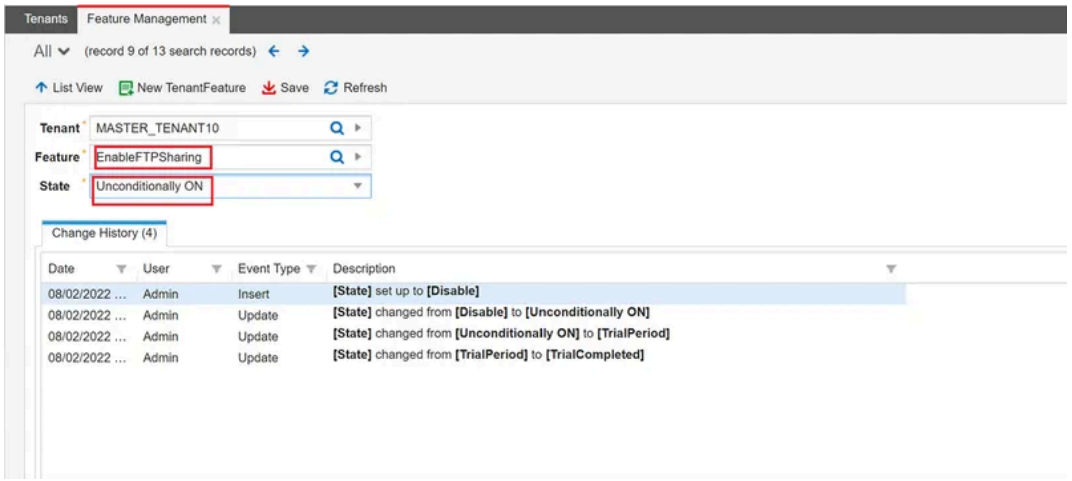
- Both FTP and File on Shared Folder controls are disabled using the feature flag from 2022.3 on wards. These feature flags can be turned off by the Cloud Ops in the Config DB under Feature Management workspace.
- The feature flag is not turned off to make sure both FTP and File on Shared Folder controls are still available and customer can migrate to Secure FTP without impacting their automation. However, it would be temporary.
- It's highly recommended to move your file upload related automation from FTP and File on Shared Folder location to Secure FTP on priority. Refer SecureFTP configuration documentation further.

Steps to enable or disable the feature flag

Navigate to the **Feature Management** Business Object in the Config DB and search for your tenant feature.

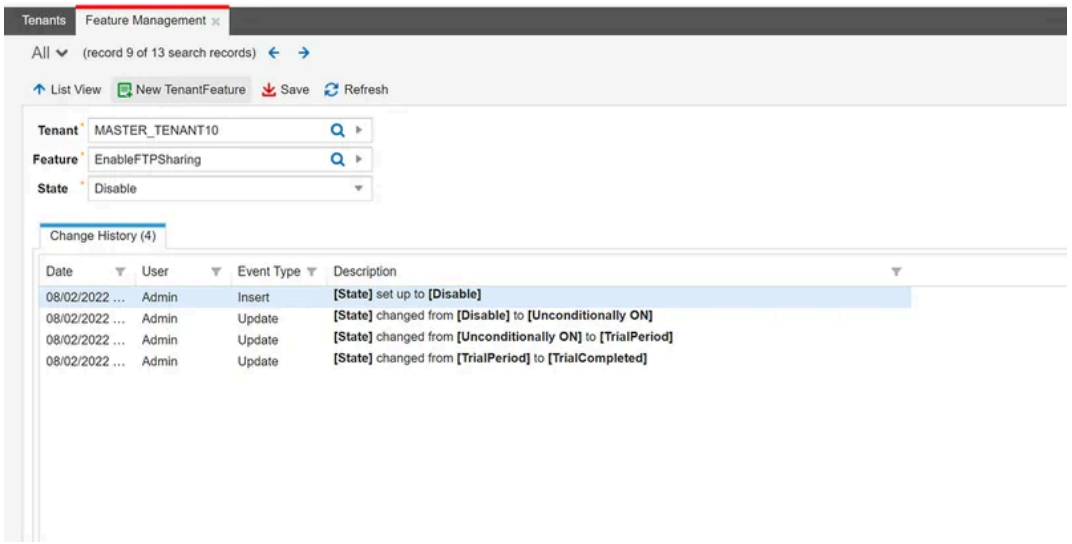
1. Once you open the feature, set the **State** as **Unconditionally ON** to enable the feature.

   Feature flag on



2. Set the **State** as **Disable** or **Trial Completed**, this disables the feature.

   Feature flag off



Post Feature Configuration

1. From the Configuration console navigate to **Data Import Connections** under **Integration tools**.
2. Click **Add new** and select the **Connection Type** field.
    - When the feature is enabled - in the **Connection Type** field drop-down list, **FTP** and **File on Shared Folder** options are available.
    - When the feature is disabled - in the **Connection Type** field drop-down list, **FTP** and **File on Shared Folder** options are not available.
3. Similarly, the **Location** field in the **Data Export Connections**,
    - When the feature is enabled, the **Location** will show **FTP** and **Secure FTP**.
    - When the feature is disabled, the **Location** will show only **Secure FTP**.

6. Setting up the Teams Bot Integration

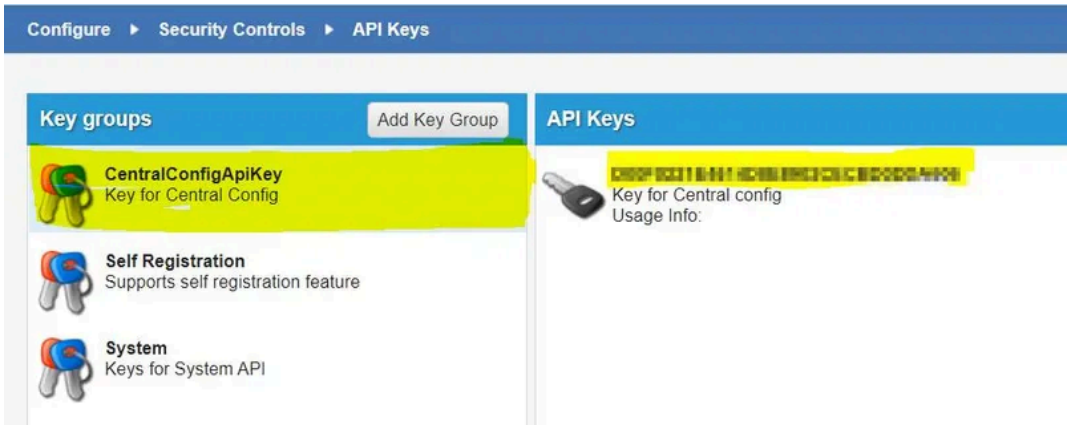
For integrating Microsoft Teams with Ivanti Neurons for ITSM, on-premise users should first setup the environment by downloading the package and executing PowerShell script.

What the Powershell Script does?

- Creates the directory in the **inetpub** folder.
- Creates a new application pool and sets item property for it.
- Creates a new web application and sets item property for it.
- Silently installs the required application, as in you might not get any prompts to proceed with the installation.
- Copies the required binaries to the destination folder.
- Creates the appsetting.json once the ipaddress/machine name/local host name is provided.

Setting up the Central Config API Key

1. Log in to the CentralConfig > **Configure Application** > **Security Controls** > **ApiKeys**.
2. Find the **CentralConfigApiKey** as shown below and copy the value.



3. Paste the copied value in the appsettings.json file for the CentralConfigApiKey. For example, *central-config-api-key"="xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"*

Executing the PowerShell Script

Download the zip file to a relevant folder in your machine.

4. Extract the zip file.
5. Open PowerShell as Administrator.
6. Go to the location where the PowerShell script is located, that is, the location where you extracted the zip file.
7. Run the script - `.\TeamsBot.ps1` .
8. You will be prompted for the Appserver as well as the Central Config ipaddress/ Machine name/ localhost name. name/ localhost. Enter the appropriate Address/Machine name/localhost.

   The configuration is done.

Troubleshooting

Incorrect ipaddress/machine name/local host name.

If you provide incorrect IP address/machine name/local host name, open the Appsetting.json file from the folder inetpub >Teams Bot Service and replace the AppserverIPAddress with the correct one.

Example of the AppSetting.json file:

{"MicrosoftAppId"="";"MicrosoftAppPassword"="";"IPCMURL"= "http://$AppserverIPAddress/HEAT/ServiceApi/IPCMService.asmx"; "WorkflowURL"= "http://$AppserverIPAddress/HEAT/ServiceApi/WorkflowService.asmx"; "ConfigURL"= "http://$CentralConfigIPAddress/CentralConfig/ConfigServiceAPI.asmx"; "RetrieveTenantLogLevel_ws_url"= "http=//$CentralConfigIPAddress/CentralConfig/RetrieveTenantLogLevel.ashx"; "EnableCentralLogging"= false; "LoggingService_ws_url"= "http://$AppserverIPAddress/Heat.Logging.Service/api/LoggingService/HeatServiceManagementLogging"; "ElapsedSecondsToFlushLog"= 60; "ItemSizeToFlushLog"= 1000; "LogSettingCacheTimeoutInMinutes"= 5; "SendLogFileLocation"= "C:\logs"; "WriteLogFileLocation"= "C:\logs"; "SendLogWaitInterval"= 300; "WriteLogWaitInterval"= 300; "EnableLogging"= true; "CacheTimeout"= "30"; "isOnPremise": false; ”CentralConfigApiKey”=”” }

> Make sure that Boolean values (true/false) don't not have double quotes.

To know the execution policy list

Run the command - `PS C: \Users\Administrator>Get-ExecutionPolicy - List`

Sample image of the policy list

To set the execution policy

Once the PowerShell script is executed successfully, execute the command - `Set-ExecutionPolicy -ExecutionPolicy “PROVIDE PREVIOUS VALUE” -Scope LocalMachine` to set the execution policy.

Sample image of setting execution policy

```
Bash
```

7. Performance Tuning

The *Performance Tuning Guide for* Neurons for ITSM describes how to configure Neurons for ITSM including the platform, the Neurons for ITSM application server, and the Neurons for ITSM application database, to optimize performance for Neurons for ITSM .

Intended Audience

This document is intended for system administrators.

To perform the procedures described in this document you should have:

- Expert knowledge of Neurons for ITSM, platform, business objects, definition sets, and the Ivanti Service Manager Configuration Console. The Configuration Console is the interface where you apply changes to the databases to give structure to Ivanti Service Manager.
- Working knowledge of database structure and terminology.
- Working knowledge of the Microsoft Windows operating system.
- Working knowledge of the Microsoft .NET architecture.
- Working knowledge of Microsoft SQL Server Management Studio.

List of Topics

This document contains the following sections:

- **Server and Connection Tuning**: Describes how to configure servers and the connections between them in Ivanti Service Manager to optimize performance.
- **Platform and Application Tuning**: Describes how to configure Ivanti Service Manager and its platform to optimize performance.
- **Database Tuning**: Describes how to tune the Ivanti Service Manager databases for optimal performance.
- **Performance Monitoring**: Lists common performance monitoring parameters.
- **Troubleshooting - Performance Tuning** : Describes how to diagnose and fix common performance issues.
- **Deleting Inactive Tables and Records**: Describes how to delete inactive tables currently residing in the Ivanti Service Manager databases.

7.1. Server and Connection Tuning

This section describes how to configure servers and the connections between them in Ivanti Service Manager to optimize performance.

- Ensuring that Hardware and Software Prerequisites are Met
- Optimizing Network Utilization and Bandwidth
- Deactivating Unused Web Servers in the Ivanti Service Manager Configuration Database
- Working with Log Files and Logging Levels
- Configuring the Application Pool Recycling Schedule

Ensuring that Hardware and Software Prerequisites are Met

Hardware and software requirements for Ivanti Service Manager are listed in the *System Requirements and Compatibility Matrix for Ivanti Service Manager*. Verify that all components in your environment meet those requirements.

Optimizing Network Utilization and Bandwidth

Recommendations for optimal Ivanti Service Manager performance are:

- Latency of 110 ms or less. See Checking Database Latency and Moving Indexes to a Separate Disk for an example Microsoft SQL script that you can execute to check latency.
- A minimum of 1.5m bits/second in bandwidth between the Ivanti Service Manager application server and remotely located client computers.

Deactivating Unused Web Servers in the Ivanti Service Manager Configuration Database

To improve performance, deactivate any unused web servers from the Ivanti Service Manager configuration database. Follow these steps:

1. Log into the Ivanti Service Manager configuration database.
2. Open the **Web Servers** workspace.
3. Click **List View** to see a list of all of the web servers.
4. Open the web server to deactivate.
5. Uncheck Server is Active.
6. Click Save.

7.1.1. Working with Log Files and Logging Levels

Log files for all services, such as workflow and escalation, can provide valuable information for performance tuning.

We recommend that you set the log levels to error. You can set the log level to debug for troubleshooting, but it is very verbose so we do not recommend using that level on a regular basis.

Follow these steps to change the log level for the **WorkflowService** and **EscalationService** logs:

1. Log in to Ivanti Service Manager.
2. Open the **Logging Configuration** workspace.
3. Open an entry.
4. From the **Log Level** field, select a log level.
5. Click Save.

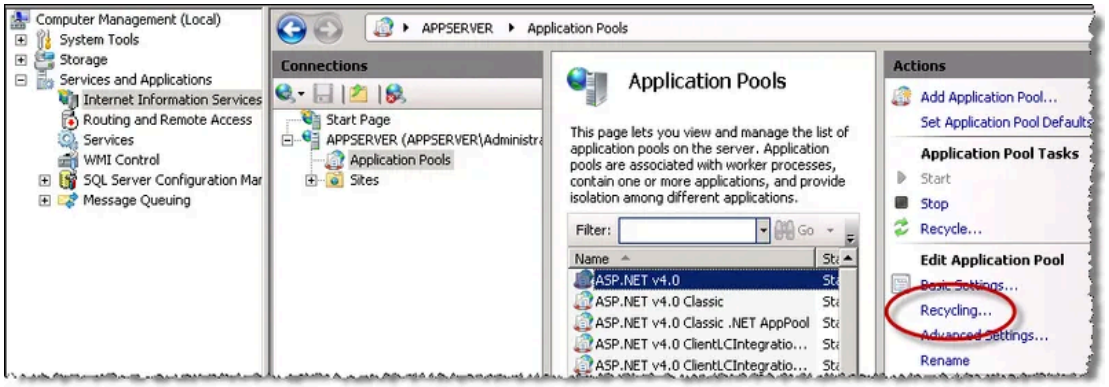
7.1.2. Configuring the Application Pool Recycling Schedule

We recommend that you schedule the application pool recycling to occur at a specific time rather than at an interval. Setting a specific time lets you schedule the recycling for off-peak times when performance is least likely to be impacted.

To configure the application pool recycling schedule, do the following:

1. Open the **Computer Management** panel by going to **Start > Control Panel > Administrative Tools > Computer Management**.
2. In the **Computer Management** panel, double-click **Services and Applications** and double-click **Internet Information Services (IIS)**.
3. In the **Connections** panel, double-click Application Server and double-click **Application Pools**.
4. In the list of application pools, highlight an application pool and click **Recycling...** located in the **Edit Application Pool** area.

Configuring the Application Pool Recycling Schedule



5. In the **Recycling Conditions** dialog box, check **Specific time(s)** and specify an off-hours time for the recycling to occur.
6. Repeat step 4 and step 5 to specify recycling schedules for the other application pools.

7.2. Platform and Application Tuning

This section describes how to configure Ivanti Neurons for ITSM and its platform to optimize performance.

- About the WAN Environment
- Planning and Scheduling Maintenance
- About Reviewing, Closing, and Deleting Records
- Tuning Business Rules and Workflows
- Best Practices for Dashboards and Layouts
- Improving Back-End Processing
- Using Event and Error Log Files for Troubleshooting
- About Using Combo Boxes Sparingly
- Evaluating Slow Running Queries
- About Data and Log Files
- About Snapshot Isolation Levels

About the WAN Environment

Applications built on the Ivanti Neurons for ITSM platform are commonly used over a WAN link. Whether deployed in a remote data center or used from branch offices, users expect performance reasonably similar to that of the LAN environment. Following the practices outlined in this guide result in a well performing application over the WAN that closely resembles performance on the local network.

To address poor performance over a WAN, it is important to clearly understand the problem. Though low bandwidth can have an impact, the foremost issue affecting WAN performance is latency, the time it takes data to travel from one endpoint to another. The higher the latency of the WAN link, the longer it takes data to traverse the link. High latency results in delays that can cause users to wait minutes for simple operations to complete.

To combat the higher latency of a WAN environment, Ivanti Neurons for ITSM was designed to make fewer “chunky” calls, rather than frequent smaller “chatty” calls. In other words, instead of retrieving little bits of data in individual requests, the system retrieves larger data sets in a single request. This well-established design principle reduces the overhead of traversing high-latency links, resulting in much improved application performance.

There are certain configurations and features within Ivanti Neurons for ITSM that can affect the behavior of the application, causing it to be more or less chatty, specifically, caching business objects and definitions and using script expressions. This guide covers these areas in detail and provides guidelines for keeping these areas in check.

This section does not describe how to fix network or WAN problems. Rather, this section focuses on ensuring that Ivanti Neurons for ITSM is configured to behave in a manner that produces acceptable performance on a well performing WAN.

7.2.1. Planning and Scheduling Maintenance

Create and follow a regular maintenance plan. Performing regular, ongoing maintenance is crucial to ensuring optimal system performance. At a minimum, a maintenance plan should include the following:

- A schedule for deleting records and tables. See **Deleting Inactive Tables and Records**.
- Running a tuning advisor tool to determine index tuning actions and then performing those actions. See **Database Tuning**.
- Scanning for duplicate indexes and then removing the duplicates. See **Checking for Duplicate Indexes**.
- Configuring database auto-growth. See **Setting the Value for the Database Auto-Growth Parameter**.

Depending on your system requirements, it is likely that you will perform additional activities as part of a maintenance plan.

The following is an example maintenance plan:

Month	Maintenance Task
January	• Evaluate the current archival plan and modify it as necessary. • Record performance metrics to establish a baseline. • Run the Microsoft SQL tuning advisor to determine recommended index changes. • Apply recommended index changes. • Scan for and delete duplicate indexes. • Check for license renewal requirements.
February	• Evaluate the current archival plan and modify it as necessary. • Troubleshoot errors reported in the event log and message queuing.
March	• Evaluate the current archival plan and modify it as necessary. • Audit log level and location settings. • Review database auto-growth and if necessary reset it to an appropriate static value.
April	• Evaluate the current archival plan and modify it as necessary. • Run the Microsoft SQL tuning advisor to determine recommended index changes. • Apply recommended index changes. • Scan for and delete duplicate indexes.
May	• Evaluate the current archival plan and modify it as necessary. • Troubleshoot errors reported in the event log and message queuing.
June	• Evaluate the current archival plan and modify it as necessary. • Record performance metrics and compare them with the recorded baseline.
July	• Evaluate the current archival plan and modify it as necessary. • Run the Microsoft SQL tuning advisor to determine recommended index changes. • Apply recommended index changes. • Scan for and delete duplicate indexes.
August	• Evaluate the current archival plan and modify it as necessary. • Troubleshoot errors reported in the event log and message queuing.
September	• Evaluate the current archival plan and modify it as necessary. • Review database auto-growth and if necessary reset it to an appropriate static value.
October	• Evaluate the current archival plan and modify it as necessary. • Run the Microsoft SQL tuning advisor to determine recommended index changes. • Apply recommended index changes. • Scan for and delete duplicate indexes.
November	• Evaluate the current archival plan and modify it as necessary. • Troubleshoot errors reported in the event log and message queuing.
December	• Evaluate the current archival plan and modify it as necessary. • Record performance metrics and compare them with the recorded baseline.

Month	Maintenance Task
	• Update infrastructure design plans and other documents. • Update system growth expectations.

About Reviewing, Closing, and Deleting Records

It is not uncommon for systems to have hundreds of thousands of active incident records that no longer need to be open and active. For example, if an incident has not been modified for several months, it can probably be closed.

Dashboard parts and saved searches may query these inactive records daily. Closing records that are older than a specified age (for example, three months) prevents them from being queried by dashboard parts and saved searches, which can significantly improve dashboard and search loading performance. Deleting closed records can also reduce the load on database server infrastructure. Additionally, deleting workflow history and instance tables prevents the workflow engine from having to query them.

We highly recommend that you implement a regularly occurring process to identify records that can be closed and deleted.

You can use service level agreements to automatically trigger a notification when an active record has been open for a specified amount of time. To do this, assign a default service level agreement to all services that do not have an active one. In the service level agreement, set the escalation to notify the owner when an active incident record has been open and active for a certain number of days.

Deleting closed and inactive records can reduce the load on database sever infrastructure. See **Deleting Inactive Tables and Records**.

7.2.2. Tuning Business Rules and Workflows

The following sections describe how to tune business rules and workflows to maximize performance. Implementing these practices can significantly improve overall system performance and reduce deadlocks and timeouts caused by the workflow engine.

- Disabling Inactive Business Rules
- Disabling Inactive Workflows
- Deleting Workflow Tables
- Validating Email Addresses in Notification Workflows

Disabling Inactive Business Rules

Review all active business rules. If a business rule is no longer used, disable it by removing its trigger and updating the rule description to disabled.

Disabling Inactive Workflows

Whenever a workflow is triggered, the system adds it to the event queue where it waits to be processed. Reducing the number of workflows waiting to be processed in the event queue improves performance. Review all active workflows. If a workflow is not used, disable it and update is description to disabled.

Deleting Workflow Tables

It is not uncommon for workflow history tables to contain hundreds of thousands of records. Workflow instance tables can also contain large numbers of records. Deleting workflow history and instance tables prevents the workflow engine from having to query them. Deleting workflow history and instance tables can significantly improve event table performance.

For details about how to deleting tables, see Deleting Inactive Tables and Records.

We recommend that you delete workflow tables on a regular basis (such as one or more times per week, if feasible).

Validating Email Addresses in Notification Workflows

If a notification workflow containing an invalid email address is triggered, event queue resources are wasted while the invalid email address is evaluated and processed. To prevent this situation, add a condition to notification workflows so that email addresses are validated before the workflow is triggered.

7.2.3. Best Practices for Dashboards and Layouts

You can improve Ivanti Service Manager performance by following the dashboard and layout best practices described in these sections:

- Dashboard Best Practices
- Layout Best Practices

Dashboard Best Practices

Because dashboards are created from multiple dashboard parts, they usually execute multiple saved searches when they are opened or updated. You can significantly improve performance by following these best practices:

- Ensure that dashboards display only active records, and that searches configured in dashboard parts do not search for closed or completed records.
- Disable or increase the duration of automatic refresh on all active dashboard parts.
- Use the fewest possible dashboard parts to create a dashboard (no more than four whenever possible).
- Wherever possible, add date filters and remove the all option.
- Wherever possible, remove memo fields from grids.
- Enable auditing only when necessary.

Layout Best Practices

Ensure that tabs in a workspace exist only if they are absolutely necessary. Workspace tabs that contain counters can consume significant resources. If a tab contains a counter, Ivanti Service Manager executes a query to return the number of records available on that tab. Each tab that generates a query adds to the time it takes for the parent workspace to load.

Improving Back-End Processing

This section describes how to tune stored procedures, triggers, and file import processes to improve performance.

- Stored Procedures and Triggers
- File Import Processes

Stored Procedures and Triggers

We recommend that you configure stored procedures and triggers outside of Ivanti Service Manager to avoid any unnecessary problems.

Stored procedures often consist of complex searches that query multiple database tables. A deadlock can result if a stored procedure is scheduled to run too frequently. Wherever possible, minimize the use of frequently scheduled stored procedures and triggers that query primary business objects such as incident, problem, change, journal, task, and other primary business objects. If possible, do not run these procedures at intervals of less than five minutes. If a frequently scheduled stored procedure is necessary, we recommend that you run it outside of normal business hours.

File Import Processes

If there are too many files to process or import in too short of a time (which might occur if the creation of a new record generates multiple queries), system performance can be significantly impacted.

Wherever possible, schedule the file import process to run outside of normal business hours.

Using Event and Error Log Files for Troubleshooting

Microsoft event logs and message queuing error logs are often helpful in determining common configuration errors, especially with regard to the email listener and file import process. We recommend that you review these logs when you troubleshoot performance issues.

7.2.4. About Using Combo Boxes Sparingly

- About Using Combo Boxes
- Creating a Pick List

About Using Combo Boxes

Do not use combo boxes for validating business objects that have a large number of rows. The maximum number of rows a combo box can display is 200. If there are more than 200 rows, the system does not return the excess rows. If a business object has more than 200 rows, we recommend using a text box with a **Search** dialog box.

To effectively validate business objects with more than 200 rows, perform one of the following:

- Change the business object from a combo box to an associated item selector. Access this function from the field control toolbar.
- Constrain the combo box field to validate using another field to reduce the number of rows returned. For example, constrain the **Employee** field to be constrained by the **Department** field, so that when validating the **Department** field, only employees from the department selected are shown.

Creating a Pick List

1. From the Configuration Console, click **Build > Pick Lists** to open the **Pick Lists** workspace. The system displays a list of pick lists.
2. Click **Add New...**. The system displays the **Add New Validation List** page.
3. Enter information into the fields.

Field	Description
List name	The name of the pick list to be stored.
Display name	The name of the pick list to be displayed in Ivanti Service Manager.
Get list data from	Check Business Object and select a business object from the drop-down list.
Display field	The name of the field to display in the pick list.
Value field	A stored field name. Select from the drop-down list.
Sort by field	The field by which to sort. Select a sorting value, either ascending or descending.

4. Add constraints to the pick list to restrict the data presented. Click the **add new** icon to add a constraint formula. This formula is based on the parameter selected in the **Get list data from** field.
5. Click Save.

See the Ivanti Service Manager online help for additional information about creating and modifying a pick list and its values.

7.2.5. Evaluating Slow Running Queries

Queries that take a long time to execute can manifest themselves just about anywhere in Ivanti Service Manager. Dashboards, saved searches, and viewing a business object are all areas impacted by slow running queries. With some familiarity with the tools available, tracking down and fixing these problems is a relatively straight forward task.

- Identifying Problem Queries
- Improving Saved Search Query Performance

Identifying Problem Queries

If queries run slowly, use the Microsoft SQL Server profiling tools to find out which queries are the problem. See Tuning Indexes for more information.

Improving Saved Search Query Performance

Saved searches are used in many places throughout Ivanti Service Manager, especially in dashboards and workspaces. Saved search queries can be complicated because they can involve related data and typically include criteria for more than just a single field. For this reason, dealing with performance problems for a saved search can be a bit more involved. Constructing saved searches in the most efficient way possible can significantly improve performance.

You can create a saved search, modify a saved search, delete a saved search, add a saved search as a favorite, and find saved searches. For more information about using the saved search feature, see the “Using Saved Search” topic in the Ivanti Service Manager online help.

> You can use the CONTAIN operator and LIKE (SQL format) operator in your search query, but this can result in slower query performance. The position of the wild card characters in the query determines the performance and index usage. Hence, your search query should have the wild card characters positioned correctly to improve the query performance.

We recommend the following:

- Using “Begin with”, “Not begin with” and “Not Null” in saved searches by adding Index to fields used in Saved Search. See " About Creating Indexes Using the Configuration Console, Not Third Party Tool " on page 25 for information on how to create indexes using the Ivanti Service Manager database.
- Minimizing sorting and grouping in grids.
- Creating index fields for tables frequently used in saved searches to reduce database access time. Use the Microsoft SQL Server profiling tools to find problem queries and to help identify indexes that can improve performance. See the Ivanti Service Manager online help for information about creating indexes.

About Data and Log Files

Microsoft SQL Server databases generally contain two types of files:

- Data files, which usually use the .mdf extension. These files contain the database data.
- Log files, which usually use the .ldf file extension. These files contain log data which can be verbose.

As a best practice, we recommend that you use house these files on different volumes.

About Snapshot Isolation Levels

- About Snapshot Isolation Levels
- Using Read Committed Snapshot Isolation (RCSI)

About Snapshot Isolation Levels

Isolation levels control how two or more transactions running simultaneously are isolated from each other in terms of locking and blocking resources. The isolation level determines the level of concurrency and data consistency. Ivanti Service Manager uses the following isolation levels:

- Read committed: This is the default transaction isolation level in Microsoft SQL Server. This isolation level prevents "dirty reads" which occur when a transaction is allowed to read data from a row that was modified by another transaction that has not yet been committed. This level acquires shared locks to prevent other transactions from modifying the data during a read operation by that transaction. A shared lock can be acquired only if there is no exclusive lock (required for data modification) by other transactions; this ensures it reads only committed data.
- Read committed snapshot isolation (RCSI): An extension of read committed but with increased concurrency. With this level, the system uses the last committed version before the statement starts, regardless of when the transaction starts.

We recommend that you use RCSI if your system meets any of the following criteria:

- It has a high number of transactions.
- If you encounter a high number of locks in your database.
- Your system has multiple back-end servers, because there could be multiple workflow, escalation, and other tasks that are triggered after the system creates or updates a record.

Note that using RCSI generally increases the CPU utilization due to the additional processing, but lowers the CPU utilization due to a reduction in locks and deadlocks. The net result is lower CPU utilization and therefore, we recommend using RCSI.

Using Read Committed Snapshot Isolation (RCSI)

For the Ivanti Service Manager database, ensure that the following two parameters are set to ON:

- Allow Snapshot Isolation
- Is Read Committed Snapshot On

To set these parameters to on, do the following:

1. Open Microsoft SQL Server Management Studio.
2. Navigate to and highlight the Ivanti Service Manager database to set the properties for.
3. Right-click and select Properties.
4. On the left, select Options.
5. Under **Miscellaneous**, set **Allow Snapshot Isolation** to True and set **Is Read Committed Snapshot On** to True.
6. Click OK.

7.3. Database Tuning

This section describes how to tune the Ivanti Service Manager database for optimal performance. If the system responds slowly while running queries, loading dashboards, or performing other actions, implement the recommendations described here. Information includes:

- Performance Enhancement Recommendations
- Tuning Indexes
- About Using Full Text Search

Performance Enhancement Recommendations

The following sections describe recommendations and caveats to keep in mind when you tune the Ivanti Service Manager database.

- About Tuning the Indexes on a Regular Basis
- Deleting Inactive Tables and Records
- Checking for Under-Utilized Indexes
- Turning on JSON Compression in Microsoft Internet Information Services (IIS)
- Checking Database Latency and Moving Indexes to a Separate Disk
- About Creating Indexes Using the Configuration Console, Not Third-Party Tools
- Checking for Duplicate Indexes
- System Tables to Not Index
- Not Using Include Fields in Indexes
- Ignoring “Create Statistics” Index Results
- Setting the Value for the Database Auto-Growth Parameter
- Monitoring Database Growth and Resetting the Auto-Growth Parameter as Necessary

About Tuning the Indexes on a Regular Basis

Evaluate on an on-going basis the need to perform the re-indexing procedure described in Tuning Indexes. If necessary, you can perform the re-indexing procedure as part of normal maintenance. It does not require hardware or other infrastructure changes.

About Deleting Inactive Tables on a Regular Basis

Whenever possible, delete inactive tables in the Ivanti Service Manager database to keep the database size and complexity to a minimum. See Deleting Inactive Tables and Records for details about deleting.

Checking for Under-Utilized Indexes

Ivanti Service Manager uses indexes to quickly access data. This is defined as reading from the index. Every time that you add a new record, you must also add the new record information to all of the indexes that it applies to. This is defined as writing to an index. While using an index saves time, adding information (writing) to the index takes time.

We recommend that you check your indexes to ensure that they are used frequently. If an index is not used frequently, the performance impact of writing to it initially might not justify the benefits that it provides. One way to check for under-utilization is to

determine the ratio of an index of reads to writes. If you have a table that you are writing records to frequently, but you are not reading using this index very often, it takes more effort to add the new record to the index than it is probably worth.

Calculate the read:write ratio for an index to determine if it is under utilized. If an index has a read:write ratio of 10:1 or less (that is, if total reads to the index does not equal at least 10 times the number of total writes to the index), you should consider deleting the index.

Run the following Microsoft SQL script on the Ivanti Service Manager database to check how often data in an index is read and how often it is written.

Bash

```
-- This script checks for under-utilized indexes.
-- Indexes in which the number of "writes" outnumbers
-- the number of "reads" by far, e.g., 90%, are
-- considered possibly under-utilized indexes.

SELECT OBJECT_NAME(s.object_id)AS 'Table Name',
i.name AS 'Index Name',
i.index_id,
user_updates AS 'Total Writes',
user_seeks + user_scans + user_lookups AS 'Total Reads',
user_updates - (user_seeks + user_scans + user_lookups)
AS 'Difference'

FROM sys.dm_db_index_usage_stats AS s WITH (NOLOCK)
INNER JOIN sys.indexes AS i WITH (NOLOCK)ON s.object_id
= i.object_id
AND i.index_id = s.index_id

WHERE OBJECTPROPERTY(s.object_id,'IsUserTable')= 1
AND s.database_id =DB_ID()
AND user_updates > (user_seeks + user_scans + user_lookups)
AND i.index_id > 1

ORDER BY 'Difference' DESC,
'Total Writes' DESC,
'Total Reads' ASC;
```

Turning on JSON Compression in Microsoft Internet Information Services (IIS)

Follow these steps to improve performance by implementing JSON compression:

1. Open the **Computer Management** panel by going to **Start > Control Panel > Administrative Tools > Computer Management**.
2. In the **Computer Management** panel, double-click **Services and Applications** and double-click **Internet Information Services (IIS)**.
3. On the page for the server (not the site) click Configuration Editor.
4. Expand system.webServer and click httpCompression.
5. Click dynamicTypes to open it.
6. Click mimeType to open the **Properties** dialog box.
7. For the **mimeType** entry, enter application/json, and for the **enabled** entry, enter True.
8. Click Apply.

9. Restart your Microsoft IIS server to load the new module.

Checking Database Latency and Moving Indexes to a Separate Disk

When data and indexes are stored on the same physical disk, resource contention might occur when the system looks up an index and retrieves the corresponding data. Measure the database latency to determine whether resource contention is affecting performance.

Run the following script on the Ivanti Service Manager database to check database latency. If database latency significantly exceeds 110 ms, move indexes to a physical disk that is separate from the disk that stores the Ivanti Service Manager database.

Bash

```
-- Description: This script checks the latency on the
-- different Ivanti databases. Ideally the latency figure
-- should be a single digit value.

SELECT
--virtual file latency
[ReadLatency] =
CASE WHEN [num_of_reads] = 0
THEN 0 ELSE ([io_stall_read_ms] / [num_of_reads]) END,
[WriteLatency] =
CASE WHEN [num_of_writes] = 0
THEN 0 ELSE ([io_stall_write_ms] / [num_of_writes]) END,
[Latency] =
CASE WHEN ([num_of_reads] = 0 AND [num_of_writes] = 0)
THEN 0 ELSE ([io_stall] / ([num_of_reads] + [num_of_writes]))            END,
--avg bytes per IOP
[AvgBPerRead] =
CASE WHEN [num_of_reads] = 0
THEN 0 ELSE ([num_of_bytes_read] / [num_of_reads]) END,
[AvgBPerWrite] =
CASE WHEN [io_stall_write_ms] = 0
THEN 0 ELSE ([num_of_bytes_written] / [num_of_writes]) END,
[AvgBPerTransfer] =
CASE WHEN ([num_of_reads] = 0 AND [num_of_writes] = 0)
THEN 0 ELSE
(([num_of_bytes_read] + [num_of_bytes_written])/
([num_of_reads] + [num_of_writes])) END,
LEFT ([mf].[physical_name], 2) AS [Drive],
DB_NAME ([vfs].[database_id]) AS [DB],
-- [vfs].*,
[mf].[physical_name]
FROM
sys.dm_io_virtual_file_stats (NULL,NULL) AS [vfs]
JOIN sys.master_files AS [mf]
ON [vfs].[database_id] = [mf].[database_id]
AND [vfs].[file_id] = [mf].[file_id]

WHERE DB_NAME ([vfs].[database_id]) like '%HEAT%'

-- and [vfs].[file_id] = 2 -- log files

-- ORDER BY [Latency] DESC
-- ORDER BY [ReadLatency] DESC
ORDER BY [WriteLatency] DESC;

GO
```

About Creating Indexes Using the Configuration Console, Not Third-Party Tools

Do not use third-party tools (such as Microsoft SQL Server Management Studio) to update indexes in the Ivanti Service Manager database. Instead, use the Configuration Console to create new indexes. We recommend this because the Configuration Console creates and maintains index information as Ivanti Service Manager metadata. Ivanti Service Manager uses this metadata whenever tables or indexes in the Ivanti Service Manager database need to be recreated.

Checking for Duplicate Indexes

Check for duplicate indexes in these situations:

- Before creating indexes in the Ivanti Service Manager database, make sure that the table for which you are creating an index does not already have an index containing the same fields.
- If you suspect that duplicate indexes already exist and are impacting performance, find and delete them.

The following sections describe how to check a specific table to determine whether it is indexed, and how to query the Ivanti Service Manager database to identify all duplicate indexes.

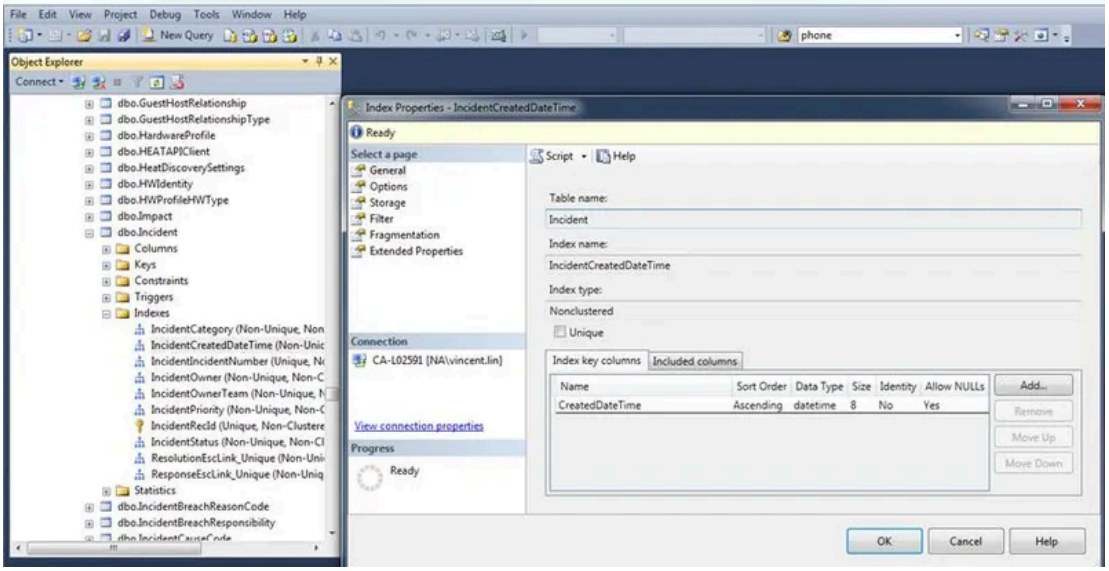
- Checking a Specific Table for Indexes
- Checking for Duplicate Indexes Using a Microsoft SQL Script

Checking a Specific Table for Indexes

Follow these steps:

1. Open Microsoft SQL Server Management Studio.
2. In the **Object Explorer** panel, navigate to the Ivanti Service Manager database.
3. Navigate to **Tables > System Tables > *table_name* > Indexes**.
4. Double-click an index to open it in the **Index Properties** window. Indexed fields are listed in the **Index key columns** area.

Checking for Fields that are Already Indexed



Checking for Duplicate Indexes Using a Microsoft SQL Script

Run the following Microsoft SQL script on the Ivanti Service Manager database to identify duplicate indexes:

Bash

```
-- This script checks for duplicate indexes.

with indexcols AS
(
SELECT object_id AS id,
index_id AS indid, name,
(SELECT case keyno WHEN 0 THEN NULL ELSE colid END AS [data()]

FROM sys.sysindexkeys AS k
where k.id = i.object_id
and k.indid = i.index_id
ORDER BY keyno, colid
for xml path('')) AS cols,

(SELECT case keyno when 0 then colid else NULL end AS [data()]
from sys.sysindexkeys AS k
where k.id = i.object_id
and k.indid = i.index_id

ORDER BY colid
for xml path ('')) AS inc

FROM sys.indexes AS i
)

SELECT object_schema_name(c1.id) + '.' + object_name(c1.id) AS 'table',
c1.name as 'index',
c2.name as 'exactduplicate'

FROM indexcols as c1
JOIN indexcols as c2
ON c1.id = c2.id
AND c1.indid < c2.indid
AND c1.cols = c2.cols
AND c1.inc = c2.inc;
```

System Tables to Not Index

We recommend that you do not index tables that begin with FRS_. These are system tables containing default indexes that are required for Ivanti Service Manager to function properly.

Not Using Include Fields in Indexes

See **Best Practices for Creating New Indexes** for more information.

Ignoring “Create Statistics” Index Results

When creating indexes, ignore the “Create Statistics” results returned by performance tuning tools. See **Best Practices for Creating New Indexes** for more information.

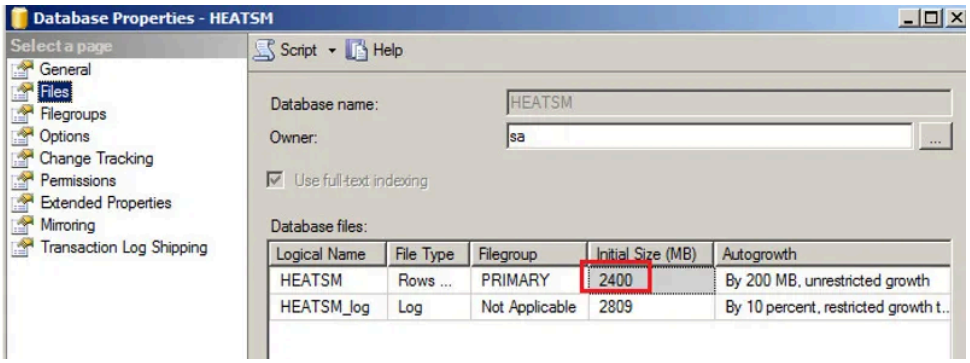
Setting the Value for the Database Auto-Growth Parameter

By default, the auto-growth size of a new database is 1 MB. This is too small for the Ivanti Service Manager database, so you must increase it.

To estimate and configure the appropriate auto-growth size for your Ivanti Service Manager database, do the following:

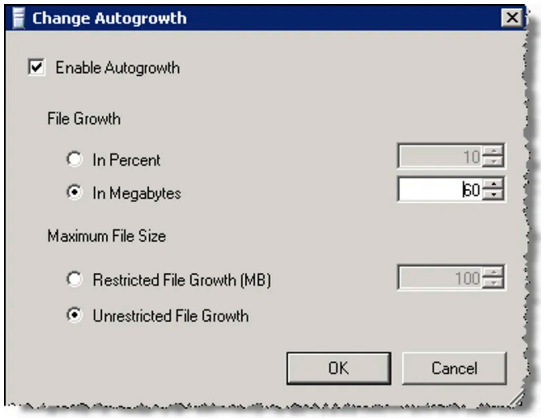
1. Estimate the total size of your initial Ivanti Service Manager database based on expected transactions for incident, service request, and other primary business objects.
2. Open Microsoft SQL Server Management Studio.
3. In the **Object Explorer** panel, navigate to the Ivanti Service Manager database.
4. Right click the Ivanti Service Manager database and select **Properties**.
5. In the **Select a page** panel, click **Files**.
6. In the **Initial Size (MB)** column of the Ivanti Service Manager database line item, set the initial database size based on your estimate from step 1.

Specifying the Initial Size of the Ivanti Service Manager Database



7. Click the **details** icon in the Ivanti Service Manager database line item to open the **Change Autogrowth for Ivanti Service Manager** dialog box.
8. Check Enable Autogrowth, select In Megabytes, and enter a number that is 5 to 10 percent of the initial database size. We recommend that you set the auto-growth parameter using a static value instead of a percentage.

Setting the Auto-Growth Parameter



9. Click **OK**.

Monitoring Database Growth and Resetting the Auto-Growth Parameter as Necessary

We recommend that you periodically evaluate whether the value for the **Enable Autogrowth** parameter is appropriate and reset it as necessary.

After you create the initial Ivanti Service Manager database, monitor its size every month for the first few months to estimate its growth rate. If necessary, reset the value for the **Enable Autogrowth** parameter based on the observed growth rate. After you determine the long-term growth pattern, you can usually wait six months or longer before checking and adjusting the value of the **Enable Autogrowth** parameter.

7.3.1. Tuning Indexes

Perform these steps to re-index the Ivanti Service Manager database:

- Capturing a Database Trace
- Getting Index Tuning Recommendations
- About Creating New Indexes

Capturing a Database Trace

Use the Microsoft SQL Server Profiler in Microsoft SQL Server Management Studio to capture a database trace. You analyze the information from this trace file in Getting Index Tuning Recommendations.

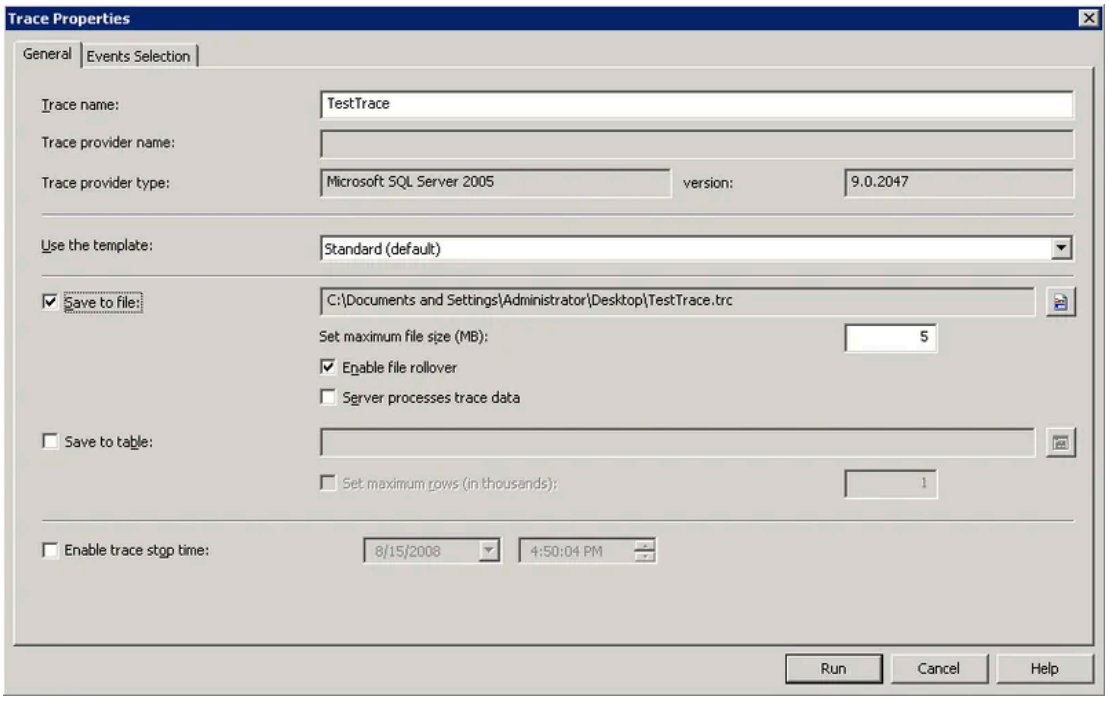
1. Open Microsoft SQL Server Management Studio by selecting **Start > All Programs > Microsoft SQL Server 2008 > SQL Server Management Studio**.
2. Highlight the server that hosts the Ivanti Service Manager database.
3. From the **Tools** menu, select **SQL Server Profiler**.
4. Start a new trace:
    - If you are prompted for login credentials, specify them for Ivanti Service Manager and click **Connect**. The system displays the **Trace Properties** window.
    - If you are not prompted for login credentials, select **File > New Trace** in the **SQL Server Profiler** window. The system prompts you for login credentials. Specify credentials for Ivanti Service Manager and click **Connect**. The system displays the **Trace Properties** window.
5. On the **General** tab, do the following:

    a. In the **Trace name** field, enter a name for the trace.

    b. In the **Use the template** field, select **Standard (default)**.

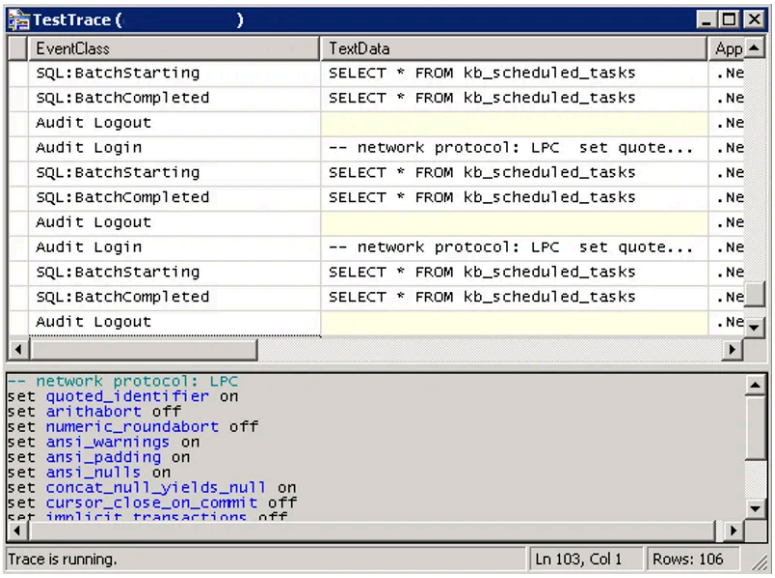
    c. Check **Save to file** and designate a file location for the trace.

Running a Trace From the Trace Properties Window



6. Click **Run** to initiate the trace. The system displays trace information while the trace runs.

A Running Trace



7. When the trace has collected enough information to analyze as described in the next section, select **Pause Trace** or **Stop Trace** from the **File** menu.

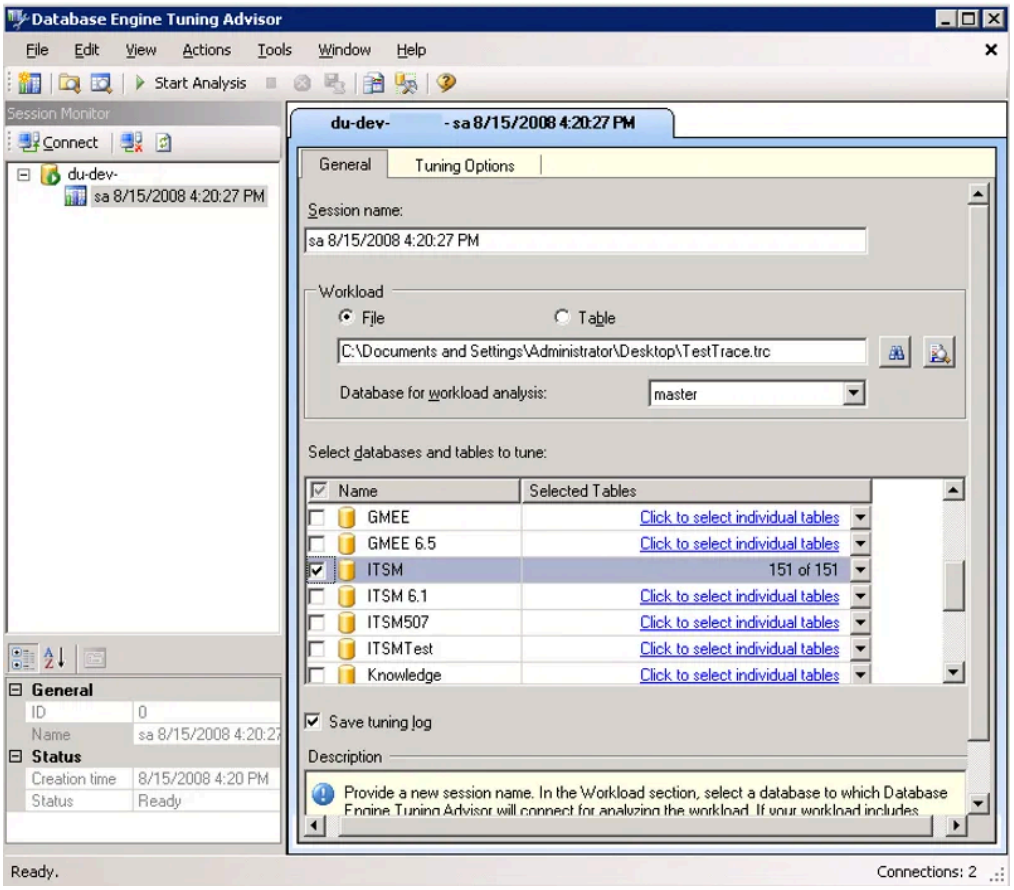
Getting Index Tuning Recommendations

Use the Index Tuning Advisor in Microsoft SQL Server Management Studio to get index tuning recommendations.

1. Open Microsoft SQL Server Management Studio by selecting **Start > All Programs > Microsoft SQL Server 2008 > SQL Server Management Studio**.
2. Highlight the server that hosts the Ivanti Service Manager database.

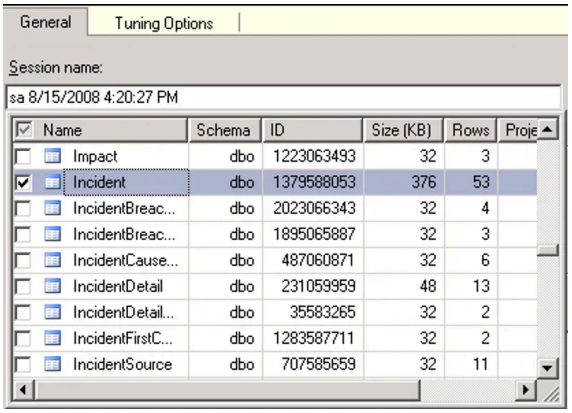
3. From the **Tools** menu, select **Database Engine Tuning Advisor**.
4. Start a new tuning session:
   - If you are prompted for login credentials, specify the credentials for Ivanti Service Manager and click **Connect**. The system displays the **Database Engine Tuning Advisor** window containing a new open session.
   - If you are not prompted for login credentials, select **File > New Session** in the **Database Engine Tuning Advisor** window. The system prompts you for login credentials. Specify credentials for Ivanti Service Manager and click **Connect**. The system opens a new session in the **Database Engine Tuning Advisor** window.
5. On the **General** tab, in the **Workload** area, browse to the trace file that you created in Capturing a Database Trace and click **Open**.
6. In the **Select databases and tables to tune** area, highlight the Ivanti Service Manager database.

Selecting the Ivanti Service Manager Database



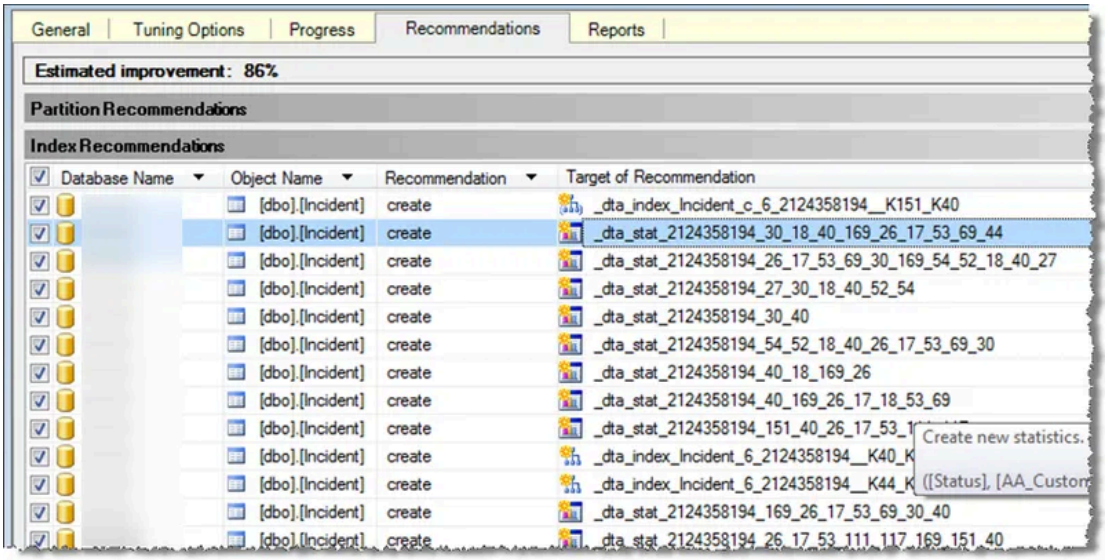
7. From the drop-down menu next to the selected database in the **Selected Tables** column, choose which tables to analyze.

Selecting Tables



8. Click the **Tuning Options** tab.
9. In the **Physical Design Structures (PDS) to use in database** section, select **Indexes**.
10. In the **Partitioning strategy to employ** section, select **No partitioning**.
11. In the **Physical Design Structures (PDS) to keep in database** section, select **Keep all existing PDS**.
12. Click **Start Analysis**.
13. To view progress, select the **Progress** tab.
14. To view recommended indexing actions, click the **Recommendations** tab and go to the **Index Recommendations** section.

Index Recommendations



1. Verify that the indexes recommended for creation do not already exist in the Ivanti Service Manager database as described in [Checking for Duplicate Indexes](#).
2. To view the tuning summary, click the **Reports** tab. You can view reports in the **Tuning Reports** drop-down list beneath the tuning summary.

About Creating New Indexes

After you verify that the indexes recommended for creation do not already exist in the Ivanti Service Manager database, use the Configuration Console to create the indexes.

- Best Practices for Creating New Indexes
- Creating New Indexes

Best Practices for Creating New Indexes

Follow these best practices when you create new indexes:

- Ivanti Service Manager does not create clustered indexes. Disregard any clustered indexes created outside of Ivanti Service Manager.
- Ivanti Service Manager does not create or use include fields or included indexes. You must create and maintain any included indexes that you use outside of the Ivanti Service Manager database. We recommend that you evaluate on a case by case basis the trade off between performance gains and maintenance overhead.
- Ivanti Service Manager maintains statistics only for indexes created in Ivanti Service Manager. Disregard any statistics returned by the create statistics operations in database performance tuning tools.

Creating New Indexes

Perform these steps to create new indexes using the Configuration Console:

1. From the Configuration Console, click **Build > Business Objects**. The system displays the **Business Objects** workspace.
2. Select a business object. The system displays the **Business Object** page.
3. Click the **Indexes** tab. The system displays the **Indexes** workspace with a list of indexes for the business object.
4. To add a new index, click **Add New...**. The system displays index fields associated with this business object.
5. Select the field to index for this business object. You can add more fields later. The system displays the **System Properties** page.
6. Enter information into the fields.

Field	Description
Index Name	A unique name for this index.
Description	A useful description for this index.
Unique	Designates this value as unique, preventing duplicate entries in the index and its backing table. You can create uniqueness by using multiple fields. This option automatically creates an unique index of this field. Unlike the primary key field: • You can have multiple unique fields in the business object. • This field is allowed to contain NULL values (although the NULL values are not indexed).
Clustered	Checks if the business object is part of a clustered index at the database level. Clustered indexes can greatly increase overall speed of retrieval, but usually only where data is accessed sequentially in the same or reverse order of the clustered index, or when a range of items is selected. This is the same as a clustered index function for a Microsoft SQL database. A clustered index cannot contain included fields (it cannot be a cover index). This checkbox is disabled for index fields that have an increasing value, such as GUID.
Key Fields	See **Viewing, Adding, and Deleting Key Fields**
Included Fields	See **Viewing, Adding, and Deleting Included Fields** .
Filter Expression	Optional. You can apply a filter to the index that does not include NULL values. Click the **add new** icon +, then choose a field from the list to create a filter. Repeat as needed. • Operation: Choose an operation from the drop-down list for this filter. • **NotNull**: The filter does not index NULL values. • Value: Enter a string value or expression for this filter.

7. Click Add this Index. The system adds the index to the business object.

Viewing, Adding, and Deleting Key Fields

The system displays the running size, in bytes, of the index for the key fields. The limit is 900 bytes.

1. From the Configuration Console, click **Build > Business Objects**. The system displays the **Business Objects** workspace.
2. Select a business object. The system displays the **Business Object** page.
3. Click the **Indexes** tab. The system displays the **Indexes** workspace with a list of indexes for the business object.
4. Click an index name to open the record.
5. To add a key field, do the following:

   a. Click the **add** icon at the end of the row.



b. Click **not set** to select a field from the drop-down list.

c. Click **Ascending** to change the order. You can select either ascending or descending.

6. To delete a key field, click the **delete** icon ✕ at the end of the row.
7. Click **Save**.

Viewing, Adding, and Deleting Included Fields

1. From the Configuration Console, click **Build > Business Objects**. The system displays the **Business Objects** workspace.
2. Select a business object. The **Business Object** page appears.
3. Click the **Indexes** tab. The system displays the **Indexes** workspace with a list of indexes for the business object.
4. Click an index name to open the record.
5. To add a key field, do the following:

a. Click the **add** icon at the end of the row.



b. Click **not set** to select a field from the drop-down list.

c. Click **Ascending** to change the order. You can select either ascending or descending.

6. To delete a key field, click the **delete** icon ✕ at the end of the row.
7. Click Save.

For additional information about adding indexes, see the Ivanti Service Manager online help. (For information about accessing the Ivanti Service Manager documentation.

7.3.2. About Using Full Text Search

Full text search support was added to enhance the Ivanti Service Manager database performance. This feature is most helpful when you need to perform similar searches against large text fields such as the **Incident.Symptom** field.

7.4. Performance Monitoring

This section describes how to monitor the performance of Ivanti Service Manager.

- Monitoring the Performance of Servers
- Monitoring the Processing Services

Monitoring the Performance of Servers

If you experience performance issues, such as long delays or timeouts, start diagnostics by monitoring the resources used on your system. The following performance monitors provide metrics that you can use to determine if system resources are over utilized.

- On the Ivanti Service Manager application server, monitor the CPU, memory, IO, and network.
- On the Ivanti Service Manager database server, monitor the CPU, memory, IO, and network plus database statistics.
- Also monitor resource utilization. For example, if the average CPU utilization is above 60%, consider adding more CPU on the Ivanti Service Manager application server.

The following are the performance counters on the Ivanti Service Manager application servers. The system samples these counters every five minutes and sends the results:

- Memory:
  - Available MBytes
- Paging file:
  - Percent usage
- Physical disk:
  - Average disk reads per second
  - Average disk writes per second
  - Disk reads per second
  - Disk writes per second
- Processor:
  - Percent processor time (CPU)
- System:
  - Processor queue length
  - Network utilization total, sent, and received per second

The following are the performance counters on the Ivanti Service Manager database server:

- Memory:
  - Available MBytes
- Paging file:
  - Percent usage
- Physical disk:
  - Average disk reads per second
  - Average disk writes per second

    - Disk reads per second
    - Disk writes per second
- Processor:
    - Percent of processor time
- Microsoft SQL Server:
    - Logins per second
    - Logouts per second
    - Lock waits per second
    - Total latch wait time (ms)
    - Total latch waits per second
    - Buffer manager page life expectancy
    - General statistics – user connections
    - Memory manager – memory grants pending
    - Microsoft SQL statistics – batch requests per second
    - Microsoft SQL statistics – compilations per second
    - Microsoft SQL statistics – recompilations per second
- System:
    - Processor queue length
    - Average wait time
    - Number of deadlocks per second
    - Lock timeouts per second

Monitoring the Processing Services

- About Monitoring the Processing Services
- Workflow Service
- Email Service
- Escalation Service
- Message Queue

About Monitoring the Processing Services

You also should monitor the processing services. The Ivanti Service Manager platform includes built-in performance counters for the Ivanti Service Manager message queue, the email service, the escalation service, and the workflow service.

To manage the processing services more effectively, the system implements counters. These counters are implemented per landscape, rather than per service across multiple landscapes. Each of the counters indicates the output across all servers.

For example, the system monitors the counter called “number of tenants” for all servers in the landscape where the service is running. This provides a holistic view of all of the servers and how the service is distributed and consumed. For example, with the workflow service, it shows how the tenants are partitioned versus the non-partitioned servers that act as a catch-all entity.

Workflow Service

The workflow service has these monitors:

Counter	Description
Events handled per poll	Represents the number of events read and handled, per tenant, per poll. If a tenant does not have any events queued, this is zero.
Events handled per second	The number of events read and handled, per tenant, per second.
Events handled versus read per poll percentage	The number of events handled, versus read, per tenant, per poll percentage.
Events read per poll	The number of events read, per tenant, per poll.
Events read per second	The number of events read, per tenant, per second.
Number of busy processor threads	Based on the configured number of processors of the workflow service on a particular server, in the WorkflowService.exe.config file, the counter shows the number of busy processors available per server. The “number of free processor threads” plus the “number of busy processor threads” should equal the total number of processors in the configuration file.
Number of free processor threads	Based on the configured number of processors of the workflow service on a particular server, in the WorkflowService.exe.config file, the counter shows the number of free processors available per server. The “number of free processor threads” plus the “number of busy processor threads” should equal the total number of processors in the configuration file.
Number of outstanding web service calls	The number of outstanding web service calls.
Number of tenants	The number of tenants assigned to the workflow service, per server. Based on partitioning, the number should indicate the total number of active tenants on the landscape.

Email Service

The email service has these monitors:

Counter	Description
Number of busy mailbox poll threads	Number of busy (consumed) mailbox polling threads, per server. The “number of free mailbox poll threads” plus the “number of busy mailbox poll threads” should equal the total number of mailbox poll threads as configured in the configuration file.
Number of busy message processing threads	Number of busy (consumed) message processing threads available, per server. The “number of free message processing threads” plus the “number of busy message processing threads” should equal the total number of message processing threads as configured in the configuration file.
Number of emails read	The number of emails read from customer mailboxes, per tenant.
Number of free mailbox poll threads	Number of free mailbox polling threads available, per server. The “number of free mailbox poll threads” plus the “number of busy mailbox poll threads” should equal the total number of mailbox poll threads as configured in the configuration file.
Number of free message processing threads	Number of free message processing threads available, per server. The “number of free message processing threads” plus the “number of busy message processing threads” should equal the total number of message processing threads as configured in the configuration file.
Number of messages read	The number of email messages read from the database, per tenant.
Number of outstanding web service calls	The number of outstanding web service calls.
Number of polled mailboxes	The number of mailboxes polled.
Number of tenants	The number of tenants assigned to the email service, per server. Based on partitioning, the number should indicate the total number of active tenants on the landscape.

Escalation Service

The escalation service has these monitors:

Counter	Description
Events handled per poll	Represents the number of events read and handled, per tenant, per poll. If a tenant does not have any events queued, this is zero.
Events handled per second	The number of events read and handled, per tenant, per second.
Events handled versus read per poll percentage	The number of events handled, versus read, per tenant, per poll percentage.
Events read per poll	The number of events read, per tenant, per poll.
Events read per second	The number of events read, per tenant, per second.
Number of busy processor threads	Based on the configured number of processors of the escalation service on a particular server, in the EscalationService.exe.config file, the counter shows the number of busy processors available per server. The “number of free processor threads” plus the “number of busy processor threads” should equal the total number of processors in the configuration file.
Number of free processor threads	Based on the configured number of processors of the escalation service on a particular server, in the EscalationService.exe.config file, the counter shows the number of free processors available per server. The “number of free processor threads” plus the “number of busy processor threads” should equal the total number of processors in the configuration file.
Number of outstanding web service calls	The number of outstanding web service calls.
Number of tenants	The number of tenants assigned to the escalation service, per server.

Message Queue

The Ivanti Service Manager message queue has these performance monitors:

Counter	Description
Average message processing time	The average time for a message to be processed by a handler. This does not account for the time that the message spends in the message queue waiting to be dispatched.
Average message size	The average size of the messages received.
Messages created per second	The number of messages received, per second.
Messages processed per second	The number of messages processed, per second by all handlers.
Number of concurrent handlers	The number of handlers that are busy working on items at any given time.
Number of concurrent messages	The number of messages that are being processed (dispatched to its handler) at any given time.
Number of outstanding web service calls	The number of outstanding web service calls.
Number of tenants	The number of tenants assigned to the message queue server.

7.5. Troubleshooting - Performance Tuning

This section describes how to diagnose and fix common performance issues.

Issues occur occasionally in Ivanti Service Manager. This does not necessarily indicate a malfunction. We recommend that you evaluate each issue on a case by case basis.

Symptom	Possible Cause	Action
Dashboards and searches are slow to load.	• Incidents and other records that are no longer active still have open status.	• Close and delete inactive incidents as described in **About Reviewing, Closing, and Deleting Records** and **Deleting Inactive Tables and Records**.
Issues and timeouts that are traceable to the workflow engine occur.	• Workflows that are inactive are still enabled. • Notification workflows do not validate email addresses prior to running.	• Disable inactive workflows as described in **Disabling Inactive Workflows**. • Validate email addresses in workflows as described in **Validating Email Addresses in Notification Workflows**.
Database command timeout errors occur.	• You need to re-index the Ivanti Service Manager database. • In general, when there is a timeout, the environment has insufficient hardware.	• Re-index the Ivanti Service Manager database as described in **Tuning Indexes**. • Verify that your system meets the hardware requirements described in the *System Requirements and Compatibility Matrix for Ivanti Service Manager*.
Workflows take a very long time to execute.		• Delete workflow-related tables as described in **Deleting Inactive Tables and Records** • Increase the processor count in the file called **WorkflowService.exe.config** to 40 as described in **Tuning Business Rules and Workflows**.

7.6. Deleting Inactive Tables and Records

This section describes how to delete inactive tables and records in the Ivanti Service Manager database.

- Best Practices for Deleting Tables and Records
- Deleting Records and Tables from the Ivanti Service Manager Databases

7.6.1. Best Practices for Deleting Tables and Records

These sections describe best practices and general recommendations to consider prior to deleting database tables:

- Identifying the Tables to Delete
- Identifying Criteria for Deleting Tables and Records

After you are familiar with these practices and have followed the described recommendations, see the "About Archiving and Deleting Data from the Ivanti Databases" topic in the in the Ivanti Service Manager online help.

Identifying the Tables to Delete

Identifying which tables to delete depends on customer-specific criteria. In general, deleting primary tables (for example, the **Incident**, **Task**, **Journal**, **Delete**, or other tables) results in the biggest performance improvement. Some tables are candidates for deleting in most situations.

Table Name	Primary Key	Foreign Key	Description
Audit_*Business_object_name*		ParentLink_RecID	Multiple tables that contain audit information generated by Ivanti Service Manager. The information is used by Self Service, the workflow engine, and the escalation engine. An example table name is **Audit_Incident**.
Frs_data_escalation_watch		ParentLink_RecID	Contains escalation tracking information.
Frs_data_workflow_instance	RecID	ParentLink_RecID	Contains workflow execution progress information.
Frs_data_workflow_history		WorkflowInstanceLink_RecID	Contains workflow execution history information.

The most commonly deleted transactional data tables are listed in the following table.

Parent business objects have primary keys. Child business objects have foreign keys. Business objects that can be both parent and child have primary and foreign keys.

Table Name	Primary Key	Foreign Key	Description
Attachment	RecID	ParentLink_RecID	Contains attachments displayed in the **Attachments** tab for major business objects (such as incident). Displays and manages all documents or files that are related to a particular business object. Contains the file name, file path, creator's name, and creation date.
FusionAttachments		RecID	A system business object stored inside an attachment containing the actual attachment block.
FRS_MyItem		SecondParentLink_RecID	Contains **My Item** records. The records contain references to incident, service request, and approval requests displayed in the **My Items** tab in Self Service.
Journal		ParentLink_RecID	Contains notes, email messages, and voice activities displayed in the **Activity History** tab for major business objects (such as incident).
Task	RecID	ParentLink_RecID	Contains task information that is displayed in the **Task** tab for major business objects (such as incident). Information pertains to general, computer provisioning, and software installation tasks.

Identifying Criteria for Deleting Tables and Records

You should only delete inactive records. Do not delete active records that contain data that can be referenced.

There are two ways to determine whether a record is inactive:

- **Based on time only**: Select a cutoff date or time, and if the record was not created or modified after the cutoff date or time, consider it inactive. This method is commonly used for records that do not have a lifecycle or different possible statuses (such as the **Audit History** business object). With this method, you specify a date and time to determine if the record is active.
- **Based on time and status**: Select a cutoff date or time and one or more statuses (or lifecycle stages). If the record was not created or modified after the cutoff date or time, or if the record meets the status criteria that you specify, consider it inactive. This method is commonly used for records that have a lifecycle or different possible statuses (such as the **Task** business object). With this method, you specify both a date and time and a status to determine if the record is active.

Deleting Records and Tables from the Ivanti Service Manager Databases

For information about deleting records and tables from the Ivanti Service Manager database, see the "About Archiving and Deleting Data from the Ivanti Databases" topic in the in the Ivanti Service Manager online help.

8. Operations Console

Use the information in this guide in conjunction with the *Installation and Deployment Guide for* Neurons for ITSM. That guide contains step-by-step instructions for installing and configuring your deployment, while this guide only describes procedures related to using the Operations Console.

> For a complete overview of all types of deployments and step-by-step instructions for installing and configuring each deployment, refer to the *"Installation and Deployment Guide for* Neurons for ITSM - *Version* 2025.3*"*. For information about using Ivanti Neurons for ITSM, including using packages and projects, see the Ivanti Neurons for ITSM online help.

Intended Audience

This document is intended for system administrators, including both Ivanti personnel and members of other organizations, who will be installing and using the Operations Console to manage tenants and tenant instances and to migrate data.

This guide is aimed at the following:

- The system administrator for an on-premise organization that has purchased, installed, and configured Ivanti Service Manager and who needs to migrate data between tenant instances. An on-premise system administrator works only on the Ivanti Service Manager system for his organization.
- The MSP (managed service provider) system administrator for an organization that manages the Ivanti Service Manager systems of several organizations. An MSP system administrator manages the Operations Consoles and migrates data for several organizations. This customer is also called a Software partner.
- Ivanti Software employees called customer service managers who manage tenants and data migration for customers using Ivanti Service Manager Cloud.

> This guide uses the term "customer." For MSP system administrators and Ivanti Software employees, the term customer refers to the actual customer (users from a different organization), but for on-premise system administrators, the term customer refers to users within the same organization.

List of Topics

This document contains the following sections:

- **Operations Console Overview**: Describes the Operations Console. Includes a terminology list and instructions for installing and logging in to the Operations Console.
- **Before You Begin Migrating Data**: Lists the four different ways to migrate data and contains conceptual information.
- **Starting the Data Migration**: Contains information such as how to create a tenant and what you must do before you migrate data for every migration method.
- **Migrating Data using Heat Development Project**: Describes how to update an existing tenant instance by migrating data between tenant instances by using the Ivanti Neurons for ITSM development project.
- **Troubleshooting and Known Issues**: Lists troubleshooting tips and known issues with the Operations Console.
- **Appendix A: Using the Operations Console to Perform Administrative Tasks**: Explains how to use the Operations Console to configure landscapes and tenants, as well as how to work with users, system settings, and the audit history.

- **Appendix B: Migrating Data from One Tenant Instance to Another Using Simple Mode or Advanced Mode**: Describes how to update an existing tenant instance by migrating data between tenant instances by using Simple Mode or Advanced Mode.
- **Appendix C: Using the ISM Development Project to Migrate Data**: Contains the information from the Ivanti Service Manager online help that describes how to use the Configuration Console to migrate data using the Ivanti Neurons for ITSM development project.
- **Appendix D: Working with Packages from the App Store**: Describes how to migrate data that came from the App Store.

8.1. Operations Console Overview

This section contains the following:

- A brief description of the Operations Console features. See **About the Operations Console**.
- Instructions for navigating the Operations Console interface. See **Operations Console Interface**.
- A summary of the actions that you can perform from each area of the Operations Console. See **About the Different Workspaces in the Operations Console**.
- Terminology used in this guide, including definitions of data types. See **Terminology**.
- Instructions for installing and logging in to the Operations Console. See **Installing the Operations Console** and **Accessing the Operations Console**.

About the Operations Console

The Operations Console provides an interface in which you can make changes to your Ivanti Service Manager system (usually in the staging instance of the tenant), test the changes (usually in the UAT instance of the tenant), and then deploy them (to the production instance of the tenant). You can manage tenant instances in the production (live system), staging (development system where initial changes are made), UAT (preproduction system where changes are tested), pilot, and other landscapes. For example, in any landscape you can:

- Create tenant instances.
- Edit the properties of tenant instances.
- Lock a tenant instance so that its metadata cannot be modified.
- Deactivate and delete tenant instances.
- Update tenant instances by migrating data from one tenant instance to another.
- Control the timing of tenant development and deployment.

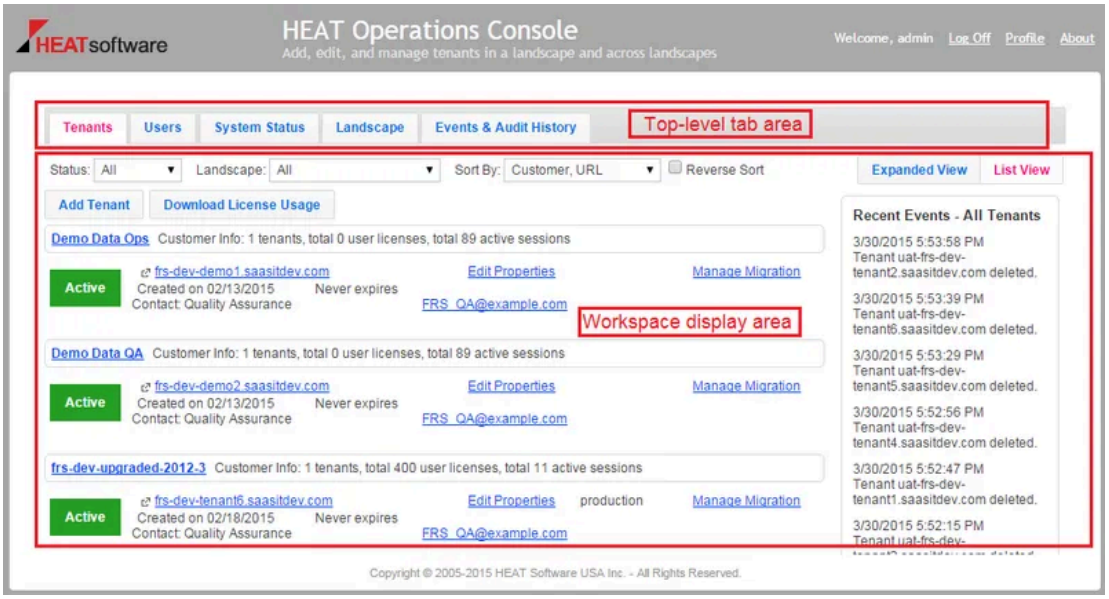
The Operations Console supports these capabilities by connecting to the Ivanti Service Manager configuration server in each landscape. The Operations Console has its own login credentials and database, which provide logging and other tenant management history capabilities.

The Operations Console also provides you with a direct connection to the Ivanti Service Manager configuration database in each landscape, allowing you to manage the Ivanti Service Manager configuration database for that landscape without going through the Ivanti Service Manager configuration server.

The Operations Console requires a database connection to the Ivanti Service Manager configuration database, a database connection to the Ivanti Service Manager application database, and a web services connection to the Ivanti Service Manager application database.

8.1.1. Operations Console Interface

Operations Console Main View (Tenants Workspace Displayed)



The Operations Console, shown in **Operations Console Main View (Tenants Workspace Displayed)**, contains these areas:

- A top-level tab area that you use to select the workspace to display. Click any tab to open it in the workspace display area. When you select a tab, the system displays the tab name in a pink font.
- A workspace display area containing the controls and detailed information for the workspace that opens when you select a top-level tab.

About the Different Workspaces in the Operations Console

Click a top-level tab to open it in the workspace display area. The following sections describe the actions that you can perform in each workspace.

- **About the Tenants Workspace**
- **About the Users Workspace**
- **About the System Status Workspace**
- **About the Landscape Workspace**
- **About the Events & Audit History Workspace**

About the Tenants Workspace

Use the **Tenants** workspace to perform a variety of tenant management activities, including creating, locking, migrating, and deleting tenants and tenant instances. See **Working with Tenants and Tenant Instances**.

About the Users Workspace

Use the **Users** workspace to view and edit information about the users who can access the Operations Console. See **Working With Operations Console Users**.

About the System Status Workspace

Use the **System Status** workspace to view status information about the entire Ivanti Service Manager system. See **Viewing System Information**.

About the Landscape Workspace

Use the **Landscape** workspace to view and edit information about landscapes. See **Configuring Landscapes**.

About the Events & Audit History Workspace

Use the **Events & Audit History** workspace to view information about each action that has occurred on a tenant or tenant instance. See **Viewing Events and Audit History**.

8.1.2. Terminology

This section defines key terms pertaining to the Operations Console and the tenant creation and data migration process.

- General Terms
- Types of Data
- Landscapes
- Tenant and Tenant Instance Types

General Terms

- Operations Console
- Customer Success Manager (CSM)
- Service Level Agreement (SLA)
- Backup Types

Operations Console

The tool used to create tenants and tenant instances, move data from one tenant instance to another (for example, from the staging instance of a tenant to the production instance), and perform various administrative tasks.

Customer Success Manager (CSM)

An Ivanti Software staff member who is the primary contact for a customer, providing support in managing the customer's Ivanti Service Manager system. A CSM establishes a long-term relationship with a customer while also supporting the Ivanti Software Professional Services Organization and other Ivanti Software teams throughout the customer's engagement with Ivanti Software.

A CSM works with a customer to do the following:

- Understand the customer's business requirements.
- Help the customer understand the roles of other Ivanti Service Manager implementation team members.
- Work with the customer to understand the customer's predefined metrics of success and ensure that those metrics are tracked and met during the customer's relationship with Ivanti Software.
- Ensure that the customer's questions and issues are acted upon quickly by the appropriate implementation team members.
- Provide the customer with information and access to training resources to ensure that the Ivanti Service Manager system is successful. Information and training resources are provided over the duration of the customer's relationship with Ivanti Software.

Service Level Agreement (SLA)

A customer's service level agreement defines the response and resolution target times for services provided by Ivanti Software. In the context of Ivanti Service Manager, a service level agreement determines the details that govern the process to provision, configure, and upgrade the tenant and its instances.

A support service is offered through the CSM organization. Incidents and service requests are covered under the service delivery agreements, meaning that they are handled and completed within agreed time to the satisfaction of the customer within the boundaries of the contractual agreements.

Backup Types

There are three steps to the process of backing up a database:

1. Back up the database on the source system.
2. Copy the database from the source system to the target system.
3. Restore the database on the target system.

You can either use Microsoft SQL or LiteSpeed to perform the backup. Use Microsoft SQL to perform the backup. There are several backup types you can choose from when using the Operations Console:

- **From last LiteSpeed backup**: Copies the source database from the source database backup location to the target database backup location and restores the database from there.
- **From live LiteSpeed backup**: Performs a live LiteSpeed backup of the source database and places the backup in the source database backup location. Copies the source database backup to the target database backup location and restores the database from there.
- **From copied LiteSpeed backup**: The source database has already been manually copied to the target database backup location. If you choose this option, the system adds a field called **Backup Filename** where you enter the name of the LiteSpeed backup.
- **From last MSSQL backup**: Copies the source database from the source database backup location to the target database backup location and restores the database from there.
- **From live MSSQL backup**: Performs a live Microsoft SQL backup of the source database and places the backup in the source database backup location. Copies the source database backup to the target database backup location and restores the database from there.
- **From copied MSSQL backup**: The source database has already been manually copied to the target database backup location. If you choose this option, the system adds a field called **Backup Filename** where you enter the name of the Microsoft SQL backup.
- **Database already restored**: The database was already restored manually outside of the Operations Console (for example, through Microsoft SQL Server Management Studio). Choosing this option means the database is not backed up or restored.

Types of Data

- Configuration Data (Settings)
- Master Data
- Metadata
- Transactional Data
- Validation Data

Configuration Data (Settings)

Configuration data, also called settings, is defined as core configuration attributes that define Ivanti Service Manager.

The only configuration data (settings) that the system migrates when you migrate configuration data (settings), as described in Updating Configuration Data (Copy Settings), are the following:

- Allowed attachments
- Authentication providers such as LDAP, SAML, OpenId, and Windows Integrated Security

- Bulk upload profiles
- Dynamic styles
- Global constants
- Integration settings
- LDAP configurations
- Object matching lists
- Password policies
- Voice integrations
- Watch list items
- Email configurations (including approval status keywords, email status update mapping settings, ignore message with subject settings, object mappings for email, task assignment status keywords, truncate body settings, and trusted sender domains)

You cannot migrate other types of configuration data (settings) using the Operations Console. To migrate this data, do so manually outside of the Operations Console by using Microsoft SQL commands.

Master Data

Master data is defined as data sets that are used across an organization and are typically unique to that organization. User and role definitions, role assignments, and categories are examples of master data. Customers typically make their own changes to their master data, although the implementation team can also change master data for a customer.

Master data is migrated together with metadata so that it is consistent with the business object definitions. There are four types of master data and you can migrate any or all of them:

- User data, except for user email addresses because they may have been scrubbed.
- User-related data, such as contact group, standard user team, organizational unit, and so on.
- Service request data, such as request offerings, request parameters, and so on.
- Escalation schedule.

You can now track these types of master data:

- Analytic metrics dashboards and configuration
- Schedule entries
- Escalation settings and schedules
- Escalation clock control
- Service level packages
- Service level targets

To see a list of master data for a tenant, if any has been defined, place your mouse over the destination tenant.

The only way to migrate master data is by enabling the Ivanti Service Manager development project and using the push link. See **The Four Ways to Migrate Data** for more information about this method of migrating data and see **Example: Migrating Service Level Data** for an example.

Metadata

Configured business objects, business object attributes, forms, grids, quick actions, searches, and so on. When you reconfigure your Ivanti Service Manager system, most of the changes are implemented by editing metadata.

The types of metadata are as follows:

- Business objects
- Counters
- Forms
- Grids
- Layouts
- Quick actions
- Rules
- Rule overrides
- Branding
- Roles
- Validation (pick list) definitions
- Business rules
- Searches (namely saved searches)
- Dashboards
- Chart-part metadata (one type of dashboard)
- Grid-part metadata (one type of dashboard)
- Special-part metadata (one type of dashboard)
- View-part metadata (one type of dashboard)
- Workflows
- Workflow types
- Reports (subscription, schedule, and distribution are not migrated from source to target)
- Validation data (of pure validation business objects and custom data)
- Images in the frs_def_images table

You can lock all of the metadata, except for quick actions, dashboards, searches, and reports. Those four types of metadata cannot be locked and are always open to be edited in the production instance of the tenant. To lock the other types of metadata, set the metadata to read-only. Each tenant instance has a **Metadata Locked** field that lets you know if the metadata is locked for an instance of a tenant.

Transactional Data

A customer's data records, including incidents, changes, service requests, and so on.

The only way to migrate transactional data is by enabling the Ivanti Service Manager development project and using the push link. See **The Four Ways to Migrate Data** for more information about this method of migrating data and see **Example: Migrating Service Level Data** for an example.

Validation Data

Validation data objects defined by customers. You must migrate all validation data together at the same time and you must migrate metadata and validation data together in the same migration. This guarantees that when metadata is migrated, it includes all pick list definitions that are associated with validation objects.

Validation data is any data that is configured as part of a *validation* business object. A *validation* business object and its associated data (for example, a pick list and the data in the list) are both considered validation data. But a *standard* business object and its associated data (for example, a pick list and the data in the list) are not both validation data. If the business object is a *standard* business object, then that is validation data, but the associated data is not considered validation data.

Business Object	Is it Validation Data?
Standard business object	Yes
Validation business object	Yes
Data associated with a standard business object	No
Data associated with a validation business object	Yes

Landscapes

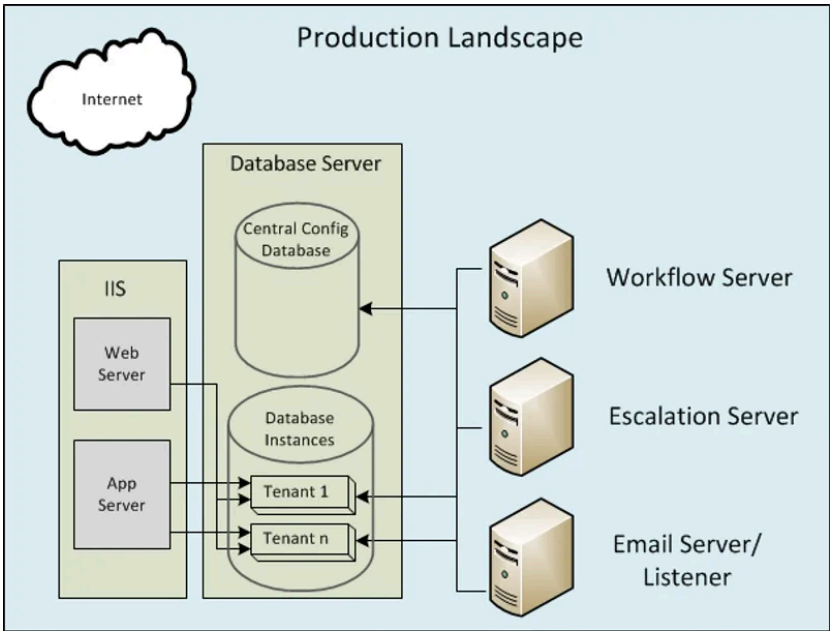
- **Landscape Definition**
- **Landscape Components**
- **Types of Landscapes**
- **About the Pilot Landscape Lifecycle**

Landscape Definition

A landscape is a set of IT infrastructure components that support tenant instances of the same type (where type refers to production, staging, and UAT). The production landscape only contains the production instances of tenants, the staging landscape only contains the staging instances of tenants, and the UAT landscape only contains the UAT instances of tenants.

Typical Production Landscape shows a typical configuration for a production landscape. Components and their locations vary from customer to customer. For example, on-premise customers probably only have one production tenant in their production landscape.

Typical Production Landscape



Landscape Components

At a minimum, a landscape contains these components:

- An individual, isolated Microsoft SQL Ivanti Service Manager application database (called IvantiSM) in each tenant (each tenant instance has its own database, but not its own database instance).
- A central Ivanti Service Manager configuration database (called ConfigDB) containing information about all tenant instances in the landscape.
- A shared pool of Ivanti Service Manager application servers for all Ivanti Service Manager instances in the landscape.
- A shared database cluster for the Ivanti Service Manager configuration and Ivanti Service Manager application databases.
- A shared pool of workflow servers used by all Ivanti Service Manager instances in the landscape.
- A shared pool of escalation engines used by all Ivanti Service Manager instances in the landscape. This is sometimes combined with the workflow server.
- A shared pool of email listeners used by all Ivanti Service Manager instances in the landscape.
- A shared pool of integration servers per function, such as survey, Discovery, Desktop Server and Management (DSM), Report Server, License Server, and so on.

Types of Landscapes

There are six types of landscapes:

- Production landscape: Hosts the production instances of tenants. Used during the data migration lifecycle. This is the environment where the product is deployed.
- Staging landscape: Hosts the staging instances of tenants. Used during the data migration lifecycle. This is the environment where development work is done.
- UAT landscape: Hosts the UAT instances of tenants. Used during the data migration lifecycle. This is the environment where testing is done.
- **Pilot landscape**: Optional. A standalone landscape that hosts the pilot tenant to provide customers with early access to a new Ivanti Service Manager point release. See **About the Pilot Landscape Lifecycle**.

- Development landscape: Optional. A standalone landscape that hosts tenant templates for demos, test drives, training, and so on.
- Other landscape: Optional. A standalone landscape type that can be used for any purpose.

About the Pilot Landscape Lifecycle

The pilot landscape hosts the pilot tenant which provides customers with early access to a new Ivanti Service Manager point release. Participating in the pilot release is optional. If you want to participate, request information from your Solutions/Implementation Manager. Ivanti Software accepts pilot customers on a first-come, first-served basis.

Approximately three weeks before a release, Ivanti Software makes the pilot landscape available to customers who have requested access. This is a temporary landscape that is only available during communicated periods.

A pilot landscape functions very similar to an alpha release. You can expect several updates and frequent changes in this landscape. We announce these frequent redeployments using system notifications only, and not through email.

Approximately a week after the Ivanti Service Manager pilot landscape release, Ivanti Service Manager moves into the staging landscape. When the system is in the staging landscape, it is available to all customers. During this time, no content moves from the staging landscape to the production landscape.

Ivanti Software posts the staging landscape update schedule on the web portal. During the staging landscape, there are approximately five updates. Each update is accompanied by a release note. You can request these staging release notes from support or your CSM.

If you find a problem, submit feedback by the end of the first week in order for it to potentially be fixed in the production landscape.

During the second week of the staging landscape, there is usually only one update and that is usually on Wednesday. This gives Ivanti Software a minimum of two days to do final testing on the system before it moves to the production landscape.

Tenant and Tenant Instance Types

A tenant is a Ivanti Service Manager system for a particular customer or application.

An instance of the Ivanti Service Manager tenant within a landscape. By default, each customer receives a production instance of the tenant and a staging instance of the tenant. Additional optional tenant instances are also available. The name and URL for each tenant instance must be unique.

> The term "tenant instance" is NOT related to the term "database instance". The use of the word instance, when referring to a tenant, simply means one copy or version of it.

The available tenant instances are:

- **Production instance of a tenant**: A tenant instance containing a customer's your production database (including all configuration and transactional data), managed under a service level agreement. All customers have a production instance of their tenant. The production instance of the tenant resides in a hardened environment in the production landscape. We do not recommend making metadata or configuration changes directly to the production instance of the tenant during production usage. Instead, make those changes in the staging instance of the tenant, push them to the UAT instance of the tenant for testing, and then push them to the production instance of the tenant.

  You can make some changes to master data (such as adding employees, role assignments, and so on) directly to the production instance of the tenant.

The production instance of the tenant is the only tenant instance covered under a service level agreement.

- **Staging instance of a tenant**: A tenant instance for developing and configuring a customer's your Ivanti Service Manager system and business processes. This tenant instance is generally used for configuration and metadata changes (but not transactional data changes). It undergoes a baselining process, in which it is updated periodically or by customer request from the production instance of the tenant to ensure that it is in sync with the production instance of the tenant prior to configuration and metadata changes. It can also be used to test new releases, new capabilities, and for training. All customers have a staging instance of their tenant; however, the staging tenant is not covered under a service level agreement.
- **UAT instance of a tenant**: A temporary tenant instance used for configuration testing and validation before the new configuration is pushed to an existing production tenant instance. The UAT instance of the tenant is required when reconfiguring an existing production instance of the tenant. It is optional when provisioning the production instance of the tenant for the first time or when upgrading to a new Ivanti Service Manager release.

On-premise customers may or may not use a UAT instance of a tenant.

- **Pilot instance of a tenant**: During the upgrade process to a new release, an optional pilot instance of the tenant is made available to customers to allow early access to a new point release. The pilot instance of the tenant resides in the pilot landscape.
- **Development instance of a tenant**: An optional development instance of the tenant for testing or development. The development instance of the tenant resides in the development landscape.
- **Other instance of a tenant**: An optional instance of the tenant for testing or development. The other instance of the tenant resides in the other landscape.
- **Tenant environment**: Infrastructure that ties together the production, staging, and UAT instances of a tenant. This infrastructure allows a complete change management process to manage and control the development and deployment lifecycles between these instances of the tenant.
- **Tenant template**: A preconfigured ("out-of-the-box") tenant that is typically used to set up a new tenant. A template implements best practices according to ITIL standards, and optimizes processes to allow fast implementation.

8.1.3. Installing the Operations Console

- Installing the Operations Console
- Accessing the Operations Console

Installing the Operations Console

You install the Operations Console when you install Ivanti Service Manager. You can find complete instructions for installing Ivanti Service Manager in the *Installation and Deployment Guide for Ivanti Service Manager*.

We highly recommend that you only install the Operations Console in the staging landscape, and not in any other landscape. For migrating data, you only need to install the Operations Console in one landscape; you should not install it in more than one landscape.

The following is a summary of the installation steps:

1. Access the installation folder on the Ivanti Software product CD or download folder and run IvantiServiceManager.exe. The system displays the **Welcome** dialog box.
2. Click **Next**. The system displays the **License Agreement** dialog box.
3. Select **I accept the terms in the license agreement** and click **Next**. The system displays the **Destination Folder** dialog box.
4. Click **Next** to accept the default installation folder. The system displays the **Setup Type** dialog box.
5. Do one of the following:
    - Select **Complete** for the installation type. The installation program installs all of the Ivanti Service Manager components including the Operations Console.
    - Select **Custom** for the installation type. Choose the components to install but ensure that you select Operations Console under Setup/Tools.
6. Click **Install** in the **Ready to Install the Program** dialog box.

The installation begins. The system displays a status dialog box, showing the installation progress of each component over the next few minutes.

7. If prompted, click Run to run the file.

Accessing the Operations Console

The Operations Console has the following default login credentials:

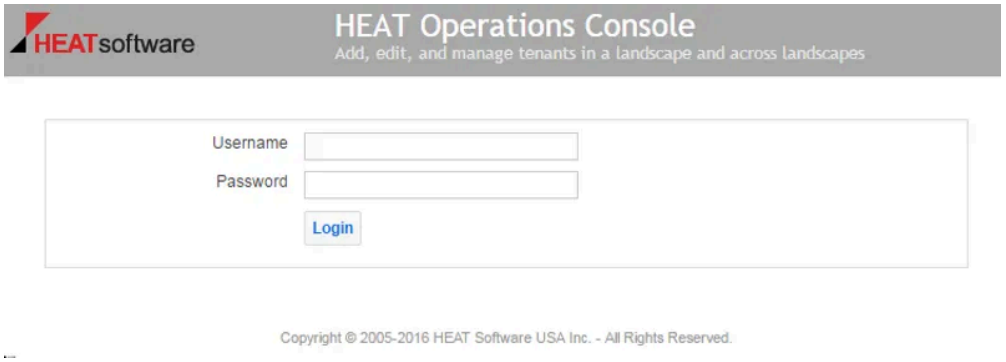
- User name: admin
- Password: manage

We highly recommend that you change the administrator password the first time that you access the system.

After installing Ivanti Service Manager, do the following to access the Operations Console:

1. Either click Start > All Programs > HEAT Software > HEAT > Ops Console or access the Operations Console from a URL in a browser. The system displays the Operations Console login page. See Operations Console Login Page.

    Operations Console Login Page



2. Enter the Operations Console user name and password.
3. Click Login.

8.2. Before You Begin Migrating Data

- The Four Ways to Migrate Data
- About Data Migration
- Conceptual Information: Migration Process Overview
- Items Excluded from Migration

The Four Ways to Migrate Data

Before you begin migrating data, it is very important that you understand the different ways there are to migrate data. There are four ways that you can migrate data:

- Enabling the Ivanti Service Manager development project and applying a package. With a package, you can migrate a very specific set of configuration changes. For example, you can add service request data to a package and migrate it. See Appendix C: Using the ISM Development Project to Migrate Data.
- Enabling the Ivanti Service Manager development project and using the push link. With this method, you can migrate master data, but not configuration data (because that is moved using packages). See Appendix C: Using the ISM Development Project to Migrate Data.
- Disabling the Ivanti Service Manager development project and using Simple Mode. With Simple Mode, you can only copy service request data and escalation schedules. When you use Simple Mode, you copy all of the configuration data from instance of a tenant to another. See How to Use Simple Mode to Migrate Data.
- Disabling the Ivanti Service Manager development project and using Advanced Mode. With Advanced Mode, you can choose which items to copy, such as all business objects but not workflows. However, you cannot specify which business objects (or whichever item you migrate). You can choose the type of items to migrate but not the specific items within that type. With Advanced Mode, you can also specify if you want to migrate master data, which you can use to copy the contents of a business object. See Using Advanced Mode to Migrate Data.

> We plan to remove Simple Mode and Advanced Mode in an upcoming release. The Ivanti Service Manager development project will then be the only data migration method available. This is because Simple Mode and Advanced Mode only migrate the difference in data between the source and the target, and do not take into account the sequence of the data changes between the two systems.

This document includes information about migrating data using all of the methods listed above. However, we highly recommend that you enable and use the Ivanti Service Manager development project, and therefore, this document focuses on those processes.

About Data Migration

The process of migrating data can be defined as moving information from the production instance of the tenant to the staging instance of the tenant, and then optionally moving information from the staging instance of the tenant to the UAT instance of the tenant, and then finally moving information back to the production instance of the tenant (from either the staging or UAT instance of the tenant).

There are two scenarios:

- Production instance → staging instance → UAT instance → production instance.
- Production instance → staging instance → production instance.

The migration process has three parts:

1. Migrating data from the production instance of the tenant to the staging instance of the tenant. This process is described in **Creating the First Staging or UAT Instance of the Tenant from the Production Instance of the Tenant**.
2. Migrating data from the staging instance of the tenant to the UAT instance of the tenant. Although this step is optional, we recommend it. This step is made up of two parts:

   a. Migrating the metadata from the staging instance of the tenant to the UAT instance of the tenant. This process is described in **Appendix B: Migrating Data from One Tenant Instance to Another Using Simple Mode or Advanced Mode**.

   b. Migrating the transactional data from the production instance of the tenant to the UAT instance of the tenant. This is the same process as **Creating the First Staging or UAT Instance of the Tenant from the Production Instance of the Tenant**.
3. Migrating data from the UAT instance of the tenant to the production instance of the tenant (if you migrated from the staging instance to the UAT) or migrating data from the staging tenant to the production tenant (if you did not migrate data to the UAT tenant). This process is described in **Appendix B: Migrating Data from One Tenant Instance to Another Using Simple Mode or Advanced Mode**.

Conceptual Information: Migration Process Overview

There are two scenarios when you perform the migration process:

- When the customer initially purchases and provisions their system. When you initially purchase and provision your system. See **About Initially Provisioning the Production Instance of the Tenant**.
- When the database changes. See **About Migrating Data After Creating the Initial Production Instance of the Tenant**.

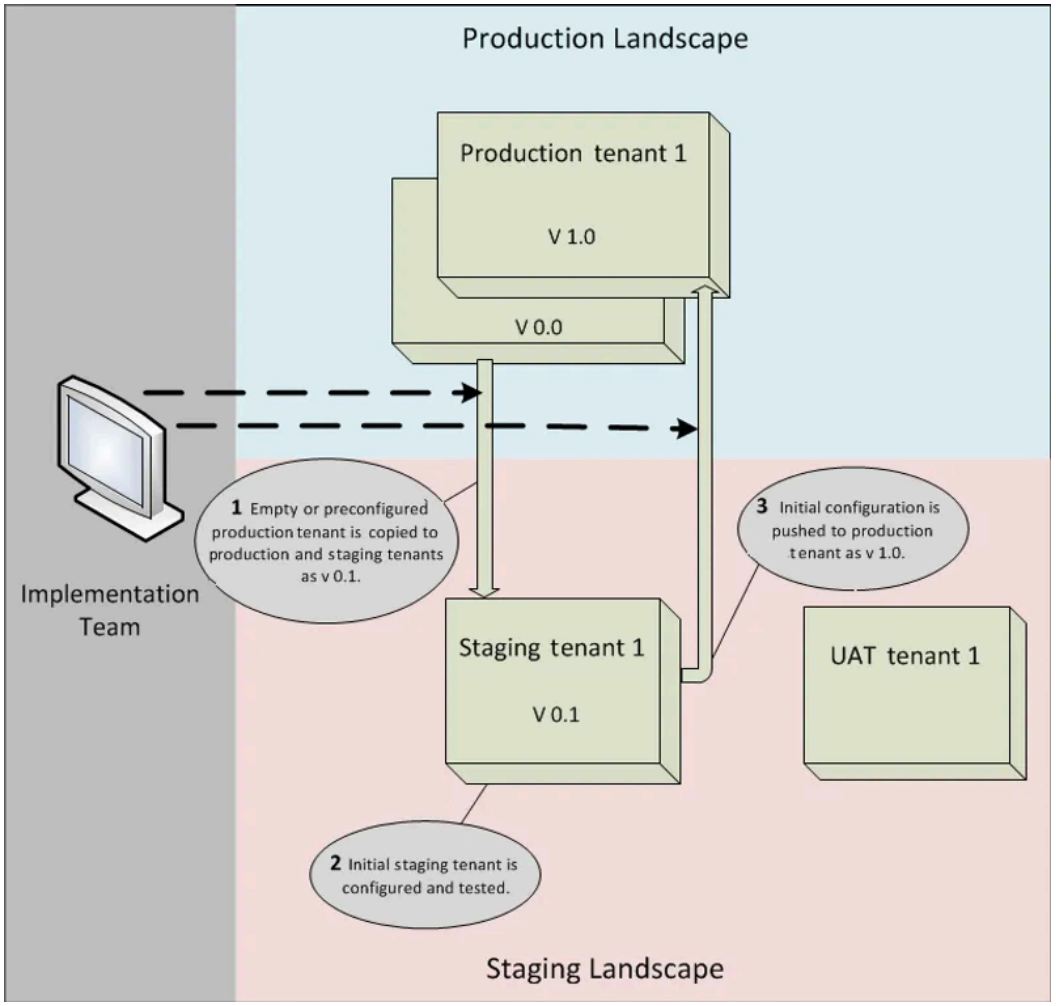
About Initially Provisioning the Production Instance of the Tenant

- **Overview of Initially Provisioning the Production Instance of the Tenant**
- **Overview of the Initial Production Instance of the Tenant Creation Process**

Overview of Initially Provisioning the Production Instance of the Tenant

Adding a Tenant shows the process for migrating data from the initial production instance of the tenant (version 0.0) to the staging instance of the tenant, and then optionally from the staging instance of the tenant to the UAT instance of the tenant, and then from the staging (or UAT) instance of the tenant back to the production instance of the tenant. This process establishes the first production instance of the tenant (version 1.0). See **Overview of the Initial Production Instance of the Tenant Creation Process** for information about the numbered steps shown in **Adding a Tenant**.

Adding a Tenant



Things to note in **Adding a Tenant**:

- The version numbers used in this example are for reference only. The software residing on a tenant does not actually have the version numbers shown here.
- Version 0.0 of the production tenant represents the starting state of the production tenant immediately after it was created. This version (or instance) typically does not contain any transactional (customer) data. You can preconfigure the metadata by either using a customer template, or if no template is available, in a way that provides a default user interface look and feel. Examples of templates are a retail business template and a high tech template.
- When creating the initial version (or instance) of the production tenant (version 1.0), the system usually bypasses the UAT instance of the tenant because there is no baseline configuration to use as a reference for regression testing.

You do not make changes to configuration data and metadata directly in a running database on the production instance of the tenant. Instead, you copy the database to the staging instance of the tenant and makes the changes there. The production instance of the tenant remains unlocked throughout the provisioning and configuration process so that you can update the production instance of the tenant with minimal overhead. The system locks the production instance of the tenant for the first time after you verify the production readiness of the production instance of the tenant.

Overview of the Initial Production Instance of the Tenant Creation Process

The following sections provide a high-level description of the initial production instance of the tenant creation process.

- Before Starting the Migration
- Step 1a: Creating the Initial Production Instance of the Tenant (Version 0.0)
- Step 1b: Creating the Initial Staging Instance of the Tenant (Version 0.1)
- Step 2: Updating the Data on the Staging Instance of the Tenant (Version 0.1)
- Step 3: Pushing the Data from the Staging Instance of the Tenant (Version 0.1) to the Production Instance of the Tenant (Version 1.0)

Before Starting the Migration

- Determine the customer's system requirements to help you design the initial production instance of the tenant (version 0.0).
- Determine your system requirements to help you design the initial production instance of the tenant (version 0.0).

Step 1a: Creating the Initial Production Instance of the Tenant (Version 0.0)

- Create the initial production instance of the tenant, called version 0.0. This initial production instance of the tenant contains basic help desk and other functionality based on industry standards and preliminary requirements. Version 0.0 is used as a starting point for additional configuration.
- The process for creating the initial production instance of the tenant is described in How to Create a New Tenant.
- Send an email to the customer with the credentials for and a link to the initial production instance of the tenant (version 0.0).

Step 1b: Creating the Initial Staging Instance of the Tenant (Version 0.1)

- Move the initial production instance of the tenant (version 0.0) to the staging instance of the tenant. The version on the staging instance of the tenant is renamed version 0.1.
- The process for moving the initial production instance of the tenant to the staging instance of the tenant is described in Appendix B: Migrating Data from One Tenant Instance to Another Using Simple Mode or Advanced Mode.
- Timing: This activity typically takes 5 days.

Step 2: Updating the Data on the Staging Instance of the Tenant (Version 0.1)

- Edit the metadata (and optionally configuration data) in version 0.1 on the staging instance of the tenant to meet your requirements.
- Ask the customer to validate the configuration on the staging instance of the tenant.
- Validate the updated configuration on the staging instance of the tenant.
- Timing: This activity typically takes from two to twelve weeks.

Step 3: Pushing the Data from the Staging Instance of the Tenant (Version 0.1) to the Production Instance of the Tenant (Version 1.0)

- Remove any sample and test data that had been temporarily added to the staging instance of the tenant.
- Push the version 0.1 metadata and configuration data from the staging instance of the tenant to the production instance of the tenant, where it becomes the first fully functional release, called version 1.0.
- The process for moving the staging instance of the tenant to the production instance of the tenant is described in Appendix B: Migrating Data from One Tenant Instance to Another Using Simple Mode or Advanced Mode.
- Ask the customer to verify that the production instance of the tenant configuration (version 1.0) is ready to "go live."

- Verify that the production instance of the tenant configuration (version 1.0) is ready to "go live."
- Lock the production instance of the tenant, either after you receive the "go live" readiness confirmation or 10 days after the "go live" date if you do not receive any confirmation.
- Timing: This activity typically takes from two to eight weeks.

About Migrating Data After Creating the Initial Production Instance of the Tenant

- Requirements Before Migrating Data from the Staging or UAT Instance of the Tenant to the Production Instance of the Tenant
- Overview of the Data Migration Process
- Data Migration Process Overview

Requirements Before Migrating Data from the Staging or UAT Instance of the Tenant to the Production Instance of the Tenant

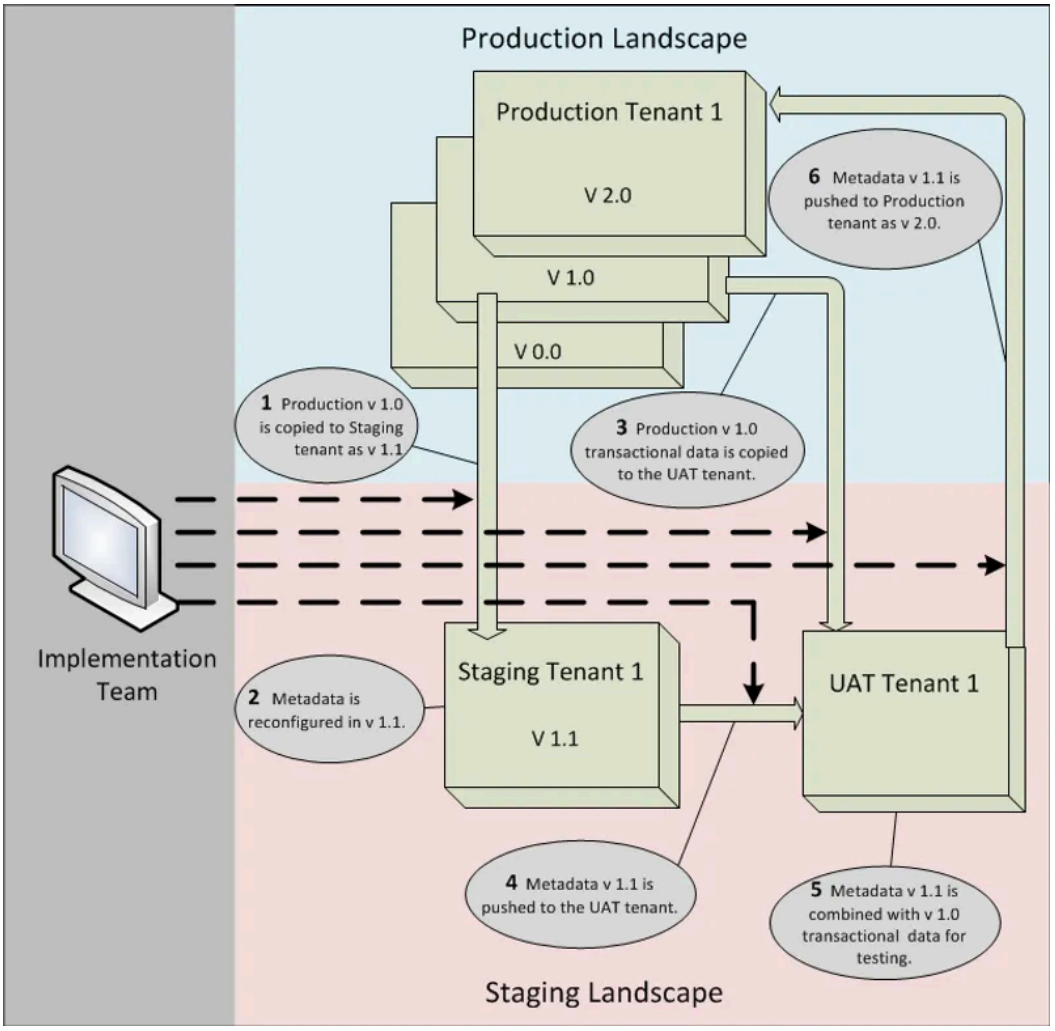
Ensure that you have completed the following requirements before you migrate data from the staging or UAT instance of the tenant to the production instance of the tenant:

- Verify that the logs for the staging or UAT instance of the tenant are free of all configuration-related errors. There cannot be any logs for quick actions, workflow, email, and so on, and there cannot be any “object reference not set to an instance of an object” errors.
- Ensure that all customer-related integration details (such as emails, LDAP, and VPN) have been documented and provided to the implementation team. Complete the information in the tab called **Email and Integration** and attach any associated documentation to the service request.
- Ensure that the implementation team completes the spreadsheet that lists the data to be cleared from the staging instance of the tenant before migrating the data to the production instance of the tenant. This might include details such as “truncate the following tables", "delete ... when...", and so on. Update the standard objects spreadsheet.
- Complete a spreadsheet that lists the data to be cleared from the staging instance of the tenant before migrating the data to the production instance of the tenant. This might include details such as “truncate the following tables", "delete ... when...", and so on. Update the standard objects spreadsheet.
- Ensure that a service request contains the required go-live date. This does not guarantee the go-live date, but gives information about when it is planned. The implementation team plans and communicates this to the Solution/Implementation Manager.
- Provide a complete list of custom or updated reports. Export the RDL for all reports and attach it to the service request.

Overview of the Data Migration Process

Data Migration Process shows the steps that occur during a major update to a functioning production tenant instance (in this example, from version 1.0 to version 2.0).

Data Migration Process



Note the following:

- The version numbers used in **Data Migration Process** are for reference only. The software residing on a tenant does not actually have the version numbers shown here.
- Version 1.0 of the production tenant instance represents a functional, running production tenant instance whose configuration remains unchanged until step 6 below.

Data Migration Process Overview

This section describes the actions that occur when data is migrated to the production instance of a tenant.

- **Before Starting the Migration**
- **Step 1**
- **Step 2**
- **Step 3**
- **Step 4**
- **Step 5**
- **Step 6**
- **Afterward**

- If a Step Fails

Before Starting the Migration

- Work with the customer to create a comprehensive project plan describing the requested changes, change verification testing, project team contact information, schedule constraints, and other relevant information.
- Create a comprehensive project plan describing the requested changes, change verification testing, project team contact information, schedule constraints, and other relevant information.

Step 1

- Copy version 1.0 of the production instance of the tenant in its entirety (that is, master data, metadata, configuration data, and transactional data) to the staging instance of the tenant. The copied version on the staging instance of the tenant is renamed version 1.1.
- Disable all email configurations (@customer.mx.domain) for the production instance of the tenant in the staging instance of the tenant, and enable the staging instance of the tenant email configuration (@tenantname-stg.saasit.com).
- Lock the production instance of the tenant. The production instance of the tenant cannot be reconfigured until approved configuration changes are applied in step 6. While the production instance of the tenant is locked, no one can make changes that will affect the migration process or be overwritten by the new configuration.
- Shrink the data on the production instance of the tenant to 5GB or less.
- The staging instance of the tenant is considered current and active.
- Timing: This step usually takes five days.

Step 2

- Edit the metadata (and optionally the configuration data) in version 1.1 to meet the customer's requirements.
- Edit the metadata (and optionally the configuration data) in version 1.1 to meet your requirements.
- Record the configuration details in a change register.
- Timing: This step typically takes three weeks. It takes two weeks to prepare and migrate the configuration from the staging instance of the tenant to the UAT instance of the tenant, and one week within the UAT instance of the tenant for final testing.

Step 3

- Copy the transactional data (that is, customer data records) from the version 1.0 production instance of the tenant to the UAT instance of the tenant and if necessary shrink the data to less than 5 GB.
- Use the transactional data to test the metadata and configuration data updates in step 5. Version 1.0 metadata and other configuration data from the production instance of the tenant is not copied to the UAT instance of the tenant.
- Timing: This step typically takes three days.

Step 4

- Ask the customer to approve the metadata push from the staging instance of the tenant to the UAT instance of the tenant. If the customer does not approve it, send the approval request again. If the customer does not approve the request a second time, work with the customer to resolve any issues.
- Review and approve the metadata push from the staging instance of the tenant to the UAT instance of the tenant.

- Send the customer a tenant instance configuration migration template to complete, which provides details about exactly what data should be migrated.
- Complete a tenant instance configuration migration template, which provides details about exactly what data should be migrated.
- Review the migration template.
- Push version 1.1 of the metadata and configuration data to the UAT instance of the tenant. Only migrate the items included in the template. This is a full metadata migration from the source to the target; that is, this is not a delta metadata migration.
- Lock the UAT instance of the tenant, making it read-only and under change control.
- Timing: This step typically takes at least one week.

Step 5

- Ask the customer to perform acceptance testing on the UAT instance of the tenant using version 1.1 metadata and configuration data combined with version 1.0 transactional data. The customer performs the tests based on documented use cases to ensure that the configuration is "go-live" ready.
- Perform acceptance testing on the UAT instance of the tenant using version 1.1 metadata and configuration data combined with version 1.0 transactional data. Performs tests based on documented use cases to ensure that the configuration is "go-live" ready.
- The customer verifies that the UAT instance of the tenant configuration and logs are "go-live" ready. Do not update the production instance of the tenant if the UAT instance of the tenant is not "go-live" ready based on predefined requirements.
- Verify that the UAT instance of the tenant configuration and logs are "go-live" ready. Do not update the production instance of the tenant if the UAT instance of the tenant is not "go-live" ready based on predefined requirements.
    - All logs on the UAT instance of the tenant must be clear of any configuration-related failures. For example, there should be no logs reporting failures in workflows, quick actions, email, object references not set to instances of objects, and so on.
    - You must approve all configurations. The customer must provide documentation describing integration details and attach it to the service request.
    - Documentation must include the completed migration template, updated clean-up scripts (if necessary), descriptions of all configuration changes and additions, and test cases that the implementation team can use to verify tenant behavior.
    - We recommend that you create documentation such as a completed migration template, updated clean-up scripts (if necessary), descriptions of all configuration changes and additions, and test cases. You can use this information to verify tenant behavior.
- The customer approves the configuration push from the UAT instance of the tenant to the production instance of the tenant. If the customer does not approve the configuration, resend the approval request. If the customer does not approve the configuration the second time, work with the customer to resolve issues and validate the changes.
- Approve the configuration push from the UAT instance of the tenant to the production instance of the tenant.
- Timing: Testing on the UAT instance of the tenant typically takes two days.

Step 6

- After the customer has approved the migration and submitted the clean-up scripts, remove any sample and test data that had been temporarily added to the staging instance of the tenant and then migrated to the UAT instance of the tenant.
- After you have approved the migration and submitted the clean-up scripts, remove any sample and test data that had been temporarily added to the staging instance of the tenant and then migrated to the UAT instance of the tenant.

- Push the UAT instance of the tenant configuration (that is, the version 1.1 metadata and configuration data) to the production instance of the tenant. This is a full metadata migration from the source to the target; that is, it is not a delta metadata migration.
- Timing: Because this step could result in a brief outage of the production instance of the tenant, we recommend that you schedule it to occur during a maintenance window (usually on a Friday or weekend), unless the customer requests otherwise.

Afterward

- The customer verifies that the production instance of the tenant is go-live ready.
- Verify that the production instance of the tenant is go-live ready.
- Request approval for the production instance of the tenant to go live. If the customer does not approve the request, work with them to resolve any issues and approve the production tenant.
- Place the UAT instance of the tenant in maintenance mode. It is no longer available to the customer.
- Place the UAT instance of the tenant in maintenance mode. It is no longer available to you.
- Lock the production instance of the tenant.
- Copy the configuration for the new production instance of the tenant to the staging instance of the tenant while the staging instance of the tenant is still locked.
- Unlock the staging instance of the tenant. From now on, the customer cannot use the staging instance of the tenant for further configuration without refreshing the production instance of the tenant. Refresh the staging instance of the tenant from the production instance of the tenant periodically to ensure that both tenant instances remain synchronized. The customer can use the staging instance of the tenant for testing and debugging purposes.
- Unlock the staging instance of the tenant. From now on, you cannot use the staging instance of the tenant for further configuration without refreshing the production instance of the tenant. Refresh the staging instance of the tenant from the production instance of the tenant periodically to ensure that both tenant instances remain synchronized. You can use the staging instance of the tenant for testing and debugging purposes.
- Timing: This activity typically takes three business days.

If a Step Fails

Step 4: If the metadata and configuration data migration from the staging instance of the tenant to the UAT instance of the tenant in step 4 fails, work with the customer to do the following:

- Obtain a corrected patch on the staging instance of the tenant.
- Re-initiate the process of migrating the metadata and configuration data from the staging instance of the tenant to the UAT instance of the tenant.

Step 6: If the metadata and configuration data migration from the UAT instance of the tenant to the production instance of the tenant in step 6 fails:

- Roll back the changes (that is, the backup copy is restored).
- Reinitiate the process of migrating the metadata and configuration data from the staging instance of the tenant to the UAT instance of the tenant.

8.2.1. Items Excluded from Migration

- Configuration Data (Settings)
- Master Data

Configuration Data (Settings)

Configuration data, also called settings, is defined as core configuration attributes that define Ivanti Service Manager.

The only configuration data (settings) that the system migrates when you migrate configuration data (settings), as described in **Updating Configuration Data (Copy Settings)**, are the following:

- Allowed attachments
- Authentication providers such as LDAP, SAML, OpenId, and Windows Integrated Security
- Bulk upload profiles
- Dynamic styles
- Global constants
- Integration settings
- LDAP configurations
- Object matching lists
- Password policies
- Voice integrations
- Watch list items
- Email configurations (including approval status keywords, email status update mapping settings, ignore message with subject settings, object mappings for email, task assignment status keywords, truncate body settings, and trusted sender domains)

You cannot migrate other types of configuration data (settings) using the Operations Console. To migrate this data, do so manually outside of the Operations Console by using Microsoft SQL commands.

Master Data

You cannot selectively update master data. If you need to update only certain records of master data (for example, only one record in a table), use Microsoft SQL commands to do so outside of the Operations Console.

8.3. Starting the Data Migration

- Starting the Tenant Instance Migration
- Creating the First Staging or UAT Instance of the Tenant from the Production Instance of the Tenant

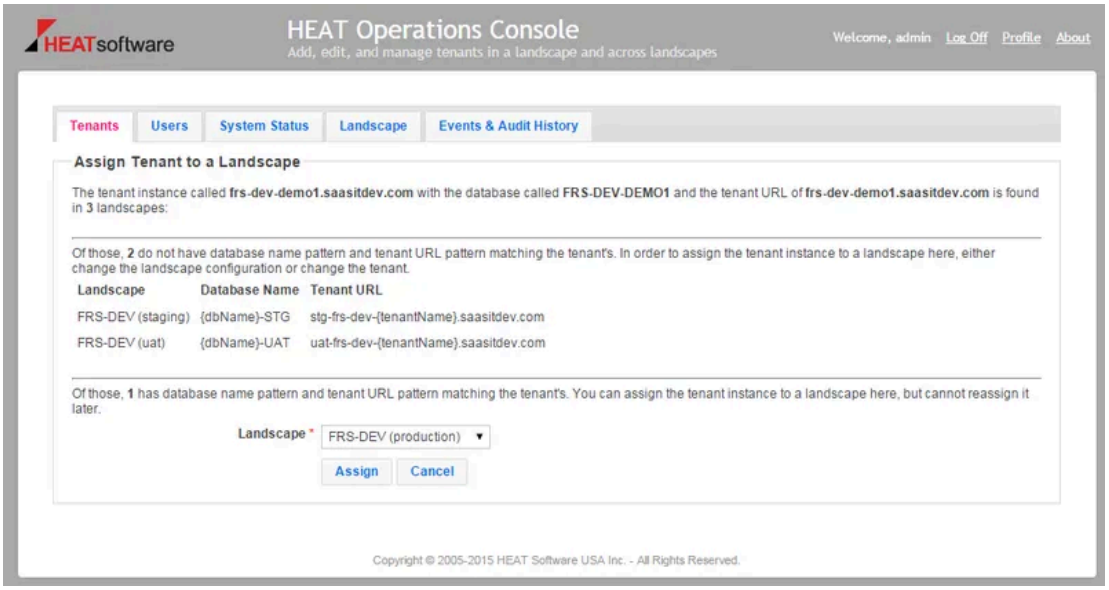
Starting the Tenant Instance Migration

Perform the following steps to start migrating data from the one tenant instance to another. These steps apply for each of the four migration methods. (See The Four Ways to Migrate Data for information about the different migration methods.)

1. Log in to the Operations Console and click the **Tenants** tab.
2. Navigate to the tenant instance to perform the migration for and click **Manage Migration**.

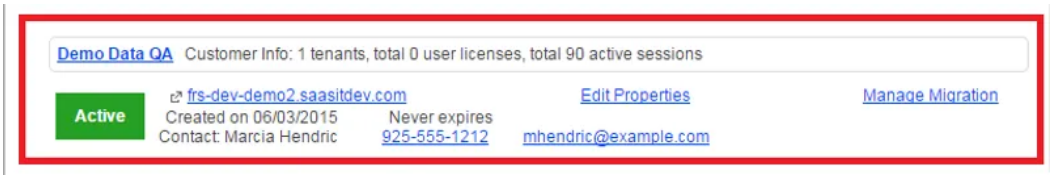
If the tenant instance is not assigned to a landscape, the system prompts you to assign the tenant instance to a landscape to prepare it for migration. See **Assigning a Tenant to a Landscape**. (If you do not see this screen, your tenant instance is ready for migration. Go to step 5.)

Assigning a Tenant to a Landscape

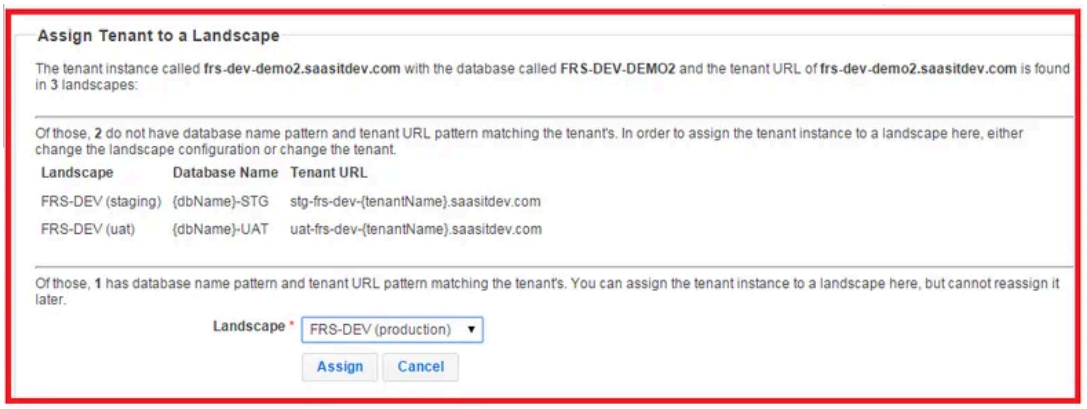


For example, if you click Manage Migration in the screen shown in **Database Pattern Screen 1**, the system displays the screen in **Database Pattern Screen 2**.

Database Pattern Screen 1

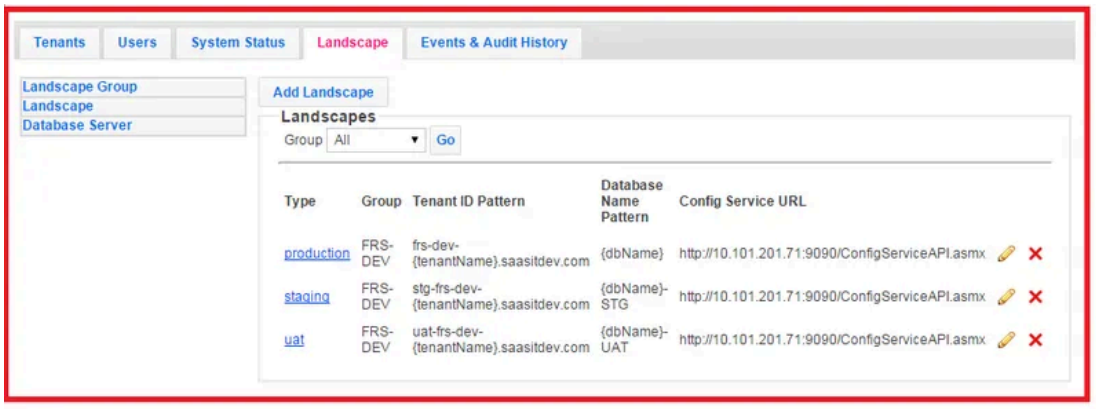


Database Pattern Screen 2



These screens are based on the landscape information, shown in **Database Pattern Screen 3**.

Database Pattern Screen 3



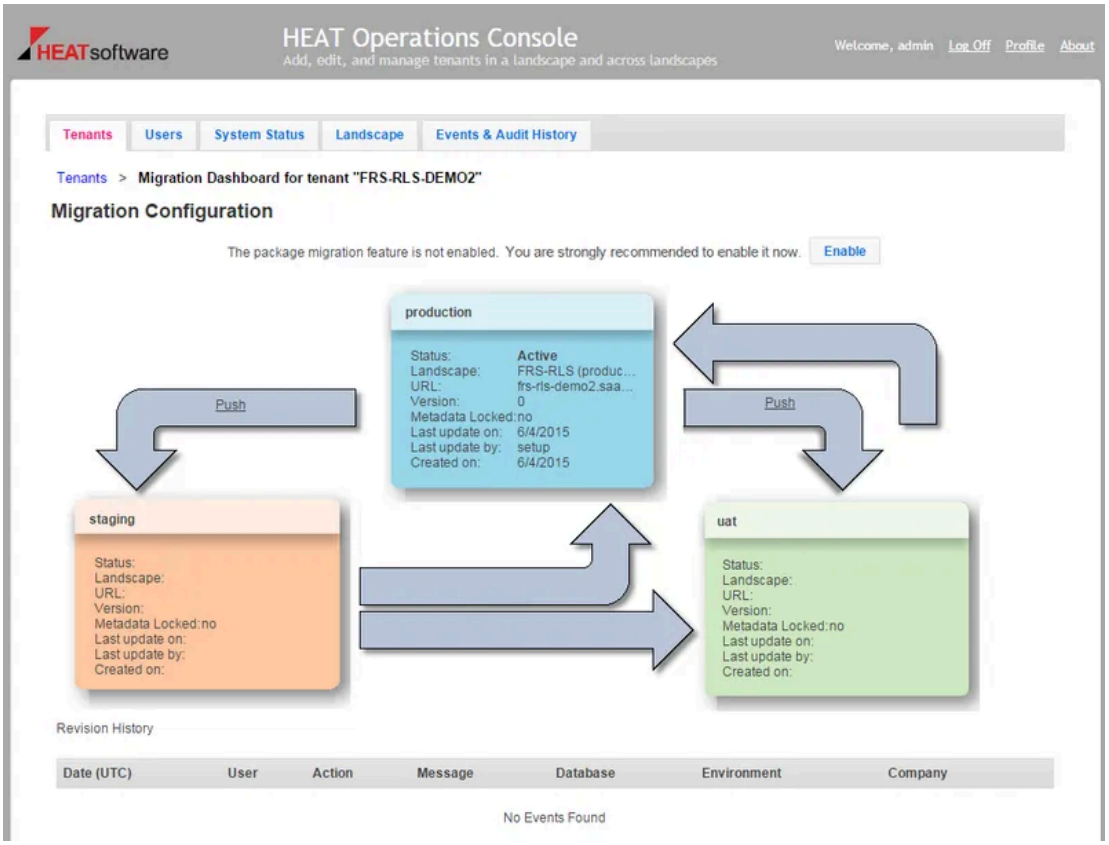
There is only one landscape with the pattern frs-dev-.saasitdev.com and that is the production landscape. Therefore, on the bottom of **Database Pattern Screen 2**, you can assign that tenant instance to that landscape. In the middle part of **Database Pattern Screen 2** are the other available landscapes but these do not have the same pattern.

3. Select a landscape from the **Landscape** drop-down list.

If you do not want to assign it to any of the landscapes in the **Landscape** drop-down list, you can click Cancel and go to the **Edit Landscape** screen to edit the landscape database pattern. See **Editing a Landscape**.

4. Click Assign. The system displays the **Migration Configuration** screen, showing all of the tenants and tenant instances and available migration paths in the landscape. See **Migration Dashboard**.

Migration Dashboard



Things to note in **Migration Dashboard**:

- The display box for each tenant instance (production, staging, and UAT in this example) contains status information that is updated whenever a tenant instance is updated.
- Arrows show all possible migration paths. If an arrow contains a **Push** or **Apply Package** link, the path is available to use based on the current status of all tenant instances in the landscape. Depending on the status of the tenant instance, some migrations paths might not be available.
- A migration history log is displayed at the bottom of the migration dashboard. If no migrations have taken place yet, "No Events Found" displays there as shown in **Migration Dashboard**.
- To enable the Ivanti Service Manager development project to migrate data, click Enable. See **Migrating Data using Heat Development Project**.

5. To start the migration, click either **Push** or Apply Package in the arrow representing the migration path to use. The system displays the **Copy *source_tenant_instance* to *target_tenant_instance*** (for example, **Copy Production to Staging**) dialog box.
6. The remaining steps in the migration procedure differ depending on which migration method you use and whether this is the first migration for this migration path. If this is the first migration, the system *creates* the first configuration on the target tenant instance. If a migration has already been performed and this is a subsequent migration, the result is an *update* of the configuration on the target tenant instance.
   - If this is the first migration, go to **Creating the First Staging or UAT Instance of the Tenant from the Production Instance of the Tenant**. Follow this process for each of the four migration methods. (See **The Four Ways to Migrate Data** for information about the different migration methods.)
   - If this is *not* the first migration and you are enabling the Ivanti Service Manager development project and either using the push link or applying a package, go to **Migrating Data using Heat Development Project**.

- If this is *not* the first migration and you are disabling the Ivanti Service Manager development project and using either Simple Mode or Advanced Mode, go to Starting the Data Migration. *We do not recommend disabling the Ivanti Service Manager development project and using either Simple Mode or Advanced Mode.*

Creating the First Staging or UAT Instance of the Tenant from the Production Instance of the Tenant

This process applies for each of the four migration methods. (See **The Four Ways to Migrate Data** for information about the different migration methods.)

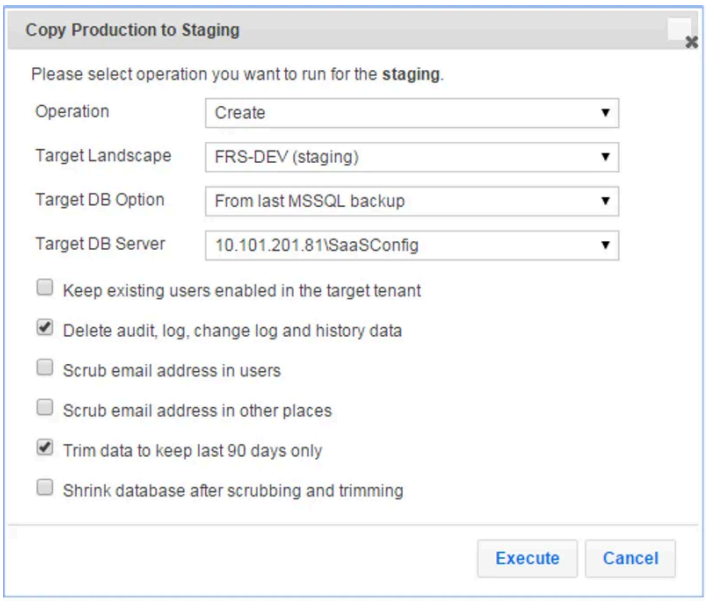
At the beginning of the migration process, you migrate the data from the production instance of the tenant to the staging instance of the tenant. When you migrate the data from the staging instance of the tenant to the UAT instance of the tenant, you must also migrate data from the production instance of the tenant to the UAT instance of the tenant. That way, the UAT instance of the tenant has the latest data from both the staging and production instances of the tenant.

Follow these steps to create the first staging or UAT instance of the tenant from the production instance of the tenant. You also follow these same steps if you want to refresh a staging or UAT instance of the tenant from the production instance of the tenant.

Perform steps 1 to 4 in **Starting the Tenant Instance Migration** before you perform the steps in this section.

When you migrate data from the production instance of the tenant to either the staging or UAT instance of the tenant for the first time, the system creates the first configuration on the staging or UAT instance of the tenant and you see a **Copy Production to Staging** (or **Copy Production to UAT**) dialog box similar to **Copy Dialog Box (Creating a Tenant Instance Configuration from the Production Tenant Instance)**.

Copy Dialog Box (Creating a Tenant Instance Configuration from the Production Tenant Instance)



1. Enter the following information in the **Copy** dialog box:

You cannot enter data for the following:

- In the **Operation** field, **Create** is the only option available. (This is because the system *creates* the first populated iteration of the target tenant instance.)
- For the **Target Landscape**, the system automatically populates this field based on where the **Push** arrow is pointing. For example, in **Migration Dashboard**, if you click Push in the arrow between the production instance of the tenant and the staging instance of the tenant, the system automatically enters the staging tenant instance into this field.

    - Select a backup option in the **Target DB Option** field. Because this is the first time that you are creating the staging or UAT instance of the tenant, there is no backup, so you must select to use a live backup. See **Backup Types** for information about the options.
    - In the **Target DB Server** field, if the target landscape contains more than one database server, select the server that hosts the Configuration Database.
    - Check **Keep existing users enabled in the target tenant** to keep all the users, and not just the administrators, active.
    - Check Delete audit, log, change log and history data to delete the data in these tables:
        - Logs
        - Frs_ops_logon_history
        - Frs_ops_gbu_pipeline_history
        - Frs_ops_metadata_history
        - Frs_ops_upload_cache
        - Journal
        - LDAPImportLog
        - Frs_ops_package_item
        - Frs_ops_package
        - All audit tables, with the name Audit_*table*
    - Check **Scrub email address in users** to ensure that you do not accidentally email users repeatedly during configuration testing on the target tenant instance. If you check this option, the system changes all email fields containing customer email to use the @it.com domain instead of the company domain. This change applies to the user's primary email, contact email, manager email, and team email.
    - Check **Scrub email address in other places** to scrub email accounts saved in transactional data (such as in incidents, service requests, workflow tasks, and so on).
    - Check **Trim data to keep last 90 days only** to only copy transactional data that was created or modified within the last 90 days. If you are updating the database from a copy of your existing customer database, most records are likely to be current within 90 days.

    > Do not check this option if you are creating the database from the sample ("out of the box") database, because the records are likely to be older than 90 days in the sample database.

    - Check **Shrink database after scrubbing and trimming** to truncate unnecessary data when the tenant instance is created from a backup. This is performed as a background task. If the database is very large, this task can take a long time or could abort. The Operations Console does not report any status on this task. If your database is very large, we recommend that you consider shrinking it outside of the Operations Console instead of in this procedure.

2. Click **Execute**. In the confirmation dialog box, click OK.

3. If there are any error or warning messages, the system displays them. Click OK.

The system migrates the data from the production instance of the tenant to the staging or UAT instance of the tenant, and configures the staging or UAT instance of the tenant for the first time.

8.4. Migrating Data using Heat Development Project

This is the recommended method for migrating data. See **The Four Ways to Migrate Data** for a list of the ways to migrate data.

For information about using the Ivanti Service Manager development project with packages that have been downloaded from the App Store, see **Appendix D: Working with Packages from the App Store**.

- About Using the Ivanti Service Manager Development Project to Migrate Data
- Cautions About the Ivanti Service Manager Development Project
- About the Ivanti Service Manager Development Project
- Using the Ivanti Service Manager Development Project to Create a Data Package
- Using the Ivanti Service Manager Development Project to Migrate a Data Package
- About Metadata Read-Only Mode
- Example: Migrating Request Offering Changes By Enabling the Ivanti Service Manager Development Project

About Using the Ivanti Service Manager Development Project to Migrate Data

Use the Ivanti Service Manager development project in the Ivanti Service Manager Configuration Console (not within the Operations Console) to create packages to migrate data from one instance of a tenant (such as staging) to another instance of a tenant (such as UAT). In this way, you can pick and choose which data to migrate instead of migrating the entire system. The Ivanti Service Manager development project allows Ivanti Service Manager to maintain a history of changes in metadata, validation data, configuration settings, and a few specific master data business objects. It also allows you to package a unit of integral changes into a package for migration to another tenant instance.

Cautions About the Ivanti Service Manager Development Project

- The Ivanti Service Manager development project gives you the ability to update the production instance of your tenant with configuration changes made in the staging instance of the tenant. If these changes conflict in some way with your Ivanti Service Manager system, they cannot be undone.
- We recommend that you enable and use the Ivanti Service Manager development project to migrate data, and that you do not switch between enabling and disabling the Ivanti Service Manager development project. See **The Four Ways to Migrate Data** for a list and description of the four different ways that you can migrate data.
- We recommend that you plan and prepare for moving a package as described in this section. We strongly encourage you to make a complete backup of the target tenant instance and to create a procedure for restoring the backup if need be:
  - If there is a problem with the UAT instance of your tenant, you can restore it from the production instance of your tenant using the Operations Console, by refreshing the UAT instance of your tenant.
  - If there is a problem with the production instance of your tenant, you need to restore the production instance of the tenant from the backup as quickly as possible using Microsoft SQL tools, as any transactions that are created in the production instance of the tenant after a backup and before a restore will be lost.
- Before you enable and use the Ivanti Service Manager development project, check with Support by visiting **Ivanti Support Team**.

About the Ivanti Service Manager Development Project

- Limitations
- Terminology
- About Using Your Named Account to Log In

Limitations

- The Ivanti Service Manager development project has not yet provided a way to validate package dependency. You need to ensure that if you make a change that depends on another change, you migrate both changes.
- Before handing out a package for migration, determine whether the package is complete and self-contained. The package should contain all dependent changes. Not having all the required changes in the package may cause the system to fail when you migrate the package and you may have to restore the target system from the back-up system.
- You cannot use the Ivanti Service Manager development project and Simple Mode or Advanced Mode at the same time. (See **The Four Ways to Migrate Data** for a list and description of the four different ways that you can migrate data.) To use Simple Mode or Advanced Mode, you must disable the Ivanti Service Manager development project.
- You cannot use the Ivanti Service Manager development project to migrate certain types of data. For a list of the data that cannot be migrated, see **Items Excluded from Migration**.

Terminology

The Ivanti Service Manager development project uses the following terms:

- **Transaction**: An individual change to the database.
- **Transaction set**: A group of transactions.
- **Project**: A group of transaction sets.
- **Package**: Contains the transaction sets from multiple projects.
- **Aspect**: Information about a feature, a component, a patch, or even a configuration. An aspect presence indicates the presence of the feature.

About Using Your Named Account to Log In

The first time that you log into Ivanti Service Manager as an administrator, the system displays a message stating that you are logging in as an administrator. This is not suitable for making configuration changes. If you make configuration changes while logged in as an administrator, you may not be able to package the changes properly.

We recommend that you no longer use a generic administrator account when you are in the Configuration Console. With the Ivanti Service Manager development project, the Ivanti Service Manager system tracks all the changes made and attributes them to the user who made the changes. If all users log in using the generic administrator account, there is no way to know who made the changes.

The system displays a message stating that the Configuration Console is in production metadata read-only mode. If you see this message, click OK and then log out as an administrator and log back in using your named account. See **About Metadata Read-Only Mode**.

Using the Ivanti Service Manager Development Project to Create a Data Package

See **Appendix C: Using the ISM Development Project to Migrate Data** for complete information about how to use the Ivanti Service Manager Configuration Console to create packages, which can be used when migrating data. You perform those tasks in the Ivanti

Service Manager Configuration Console and not in the Operations Console. This is the same information that is found in the Ivanti Service Manager online help and is only copied here for your convenience.

Using the Ivanti Service Manager Development Project to Migrate a Data Package

- Enabling the Ivanti Service Manager Development Project
- Migrating the Package
- Checking the Data Migration
- Disabling the Ivanti Service Manager Development Project

Enabling the Ivanti Service Manager Development Project

To enable the Ivanti Service Manager development project, follow these steps:

1. Log in to the Operations Console and click the **Tenants** tab.
2. Navigate to the instance of the tenant for which you will perform the migration and click **Manage Migration**.
3. If the tenant is not part of a landscape, the system prompts you to add the tenant to a landscape to prepare it for migration. Click **Proceed**. The migration dashboard opens showing all of the tenants and tenant instances and available migration paths in the landscape.
4. To enable the Ivanti Service Manager development project, click Enable.
5. Click OK at the confirmation prompt.

The migration dashboard looks like this:

> For customers who have been using HEAT Service Management since Release 2014.2 or earlier, you might not see the **Apply Package** link, even though you have enabled the Ivanti Service Manager development project. If you do not see the **Apply Package** link, refresh the staging instance of the tenant by following the process in Creating the First Staging or UAT Instance of the Tenant from the Production Instance of the Tenant.

Migrating the Package

> - To migrate a package from the staging instance to the production instance of a tenant, the package must be closed. However, the package does not need to be closed if you are migrating from the staging instance to the UAT instance of a tenant.
> - The system displays an informational message if you migrate the same package to the same tenant instance more than once.

To migrate a data package, do the following:

1. Log in to the Operations Console and click the **Tenants** tab.
2. Navigate to the instance of the tenant for which you will perform the migration and click **Manage Migration**.
3. To migrate the data package, click Apply Package in the corresponding arrow. The system displays the **Applying Package To** page.
4. Enter selections as follows:

   a. Select the name of the source package from the drop-down menu.

b. We highly recommend that you do not check Skip Target Tenant Backup.

c. Select Apply without validation to apply the package without first validating it.

5. Click Execute to migrate the package or click Preview to create an XML file with changes that you can review before you migrate the data.
6. Click OK at the confirmation prompt.

When the migration is complete, the system displays a message that the package has been applied successfully. When you refresh the staging or UAT instances of the tenant, all transaction sets are cleared.

Checking the Data Migration

To check that the data migrated successfully, log in to the UAT instance of the tenant and search for the changes.

1. Log in to the Ivanti Service Manager system for the UAT instance of the tenant. The system displays a message stating that the UAT instance of the tenant is locked.
2. Search for the changes.

Disabling the Ivanti Service Manager Development Project

1. Log in to the Operations Console and click the **Tenants** tab.
2. Navigate to the instance of the tenant for which you will perform the migration and click **Manage Migration**.
3. If the tenant is not part of a landscape, the system prompts you to add the tenant to a landscape to prepare it for migration. Click **Proceed**. The migration dashboard opens showing all of the tenants and tenant instances and available migration paths in the landscape.
4. To disable the Ivanti Service Manager development project, click Disable.

About Metadata Read-Only Mode

When you use the Operations Console to migrate data, the production and UAT instances of the tenant are always in metadata read-only mode, which is equivalent to being locked. When the system is locked, you cannot make changes to most types of metadata in the Configuration Console. (You can only make changes to the following four items: saved searches, reports, quick actions, and dashboards.) While migrating data, the system temporarily unlocks the metadata on the target system.

You cannot import high-impact packages if your system is in metadata read-only mode.

To put a system into metadata read-only mode, do the following:

1. Log in to the Operations Console.
2. Click Tenant. The system displays the **Tenants** workspace.
3. Find the tenant to edit and click Edit Properties.
4. Check Metadata Locked.
5. Click Save.

8.4.1. Example: Migrating Request Offering Changes By Enabling the Ivanti Service Manager Development Project

This example is an end-to-end example of migrating request offering changes by enabling using the Ivanti Service Manager development project. It describes how to make a change to a request offering in the staging instance of a tenant and then migrate the change to the production instance of the tenant. In this example, the Ivanti Service Manager development project is enabled and data is migrated by both applying a package and by using the push link.

This example is copied directly from the Ivanti Service Manager online help.

- About this Example
- Before You Begin
- Packaging the Changes Using Ivanti Service Manager
- Migrating the Data from the Staging Instance to the UAT Instance
- Making Changes to the Request Offering in the Configuration Console
- Testing the Change
- Migrating the Data from the Staging Instance to the Production Instance
- Migrating the Service Request Data from the UAT Instance to the Production Instance

About this Example

Ivanti Service Manager does not have a system for tracking request offerings, so to migrate a request offering from one environment to another, you must create and apply a package *and* also do a master data push using the Operations Console. Creating and applying a package alone does not migrate the request offering change.

Before You Begin

Before you begin, use the Operations Console to refresh the staging instance of the tenant from the production instance of the tenant. This ensures that you have the same baseline in both environments before you make the change.

Packaging the Changes Using Ivanti Service Manager

When you make changes to request offerings, Ivanti Service Manager tracks the changes made to the request offering and associated workflows and quick actions, but does not track the changes made to some related content. Ivanti Service Manager does not record changes for the organizational unit, configuration item service, location, or request offering links to organizational units, configuration item services, or locations. This can cause confusion when you move changes from one system to another.

The changes that the system records may differ depending on if you are logged in as an administrator or as another role when you make the changes.

- If you make the changes as a service owner, you cannot select the project to apply the changes to. Therefore, all changes are added to the default project.
- If you make the changes as an administrator, you can select the project to apply the changes to. Before you make the changes to the request offering, we recommend that you create a new project and apply the changes to that project. See Creating a Project.

1. (Optional. For administrators only.) Go to the Configuration Console, create a project, and then select it.
2. Make the changes to the request offering. The system tracks the changes to the workflows and quick actions associated with the request offering.
3. Create a package with the transaction sets associated with the changes that you made to the request offering. Ensure that the package contains all of the changes made to the system since you made the baseline, as described in **Before You Begin**, regardless of if they are related to your project.

Migrating the Data from the Staging Instance to the UAT Instance

1. Log into the Operations Console.
2. Refresh the UAT instance of the tenant from the production instance of the tenant.
3. From the **Tenants** tab, find the instance to migrate and click Manage Migration.
4. At the top of the page, click Enable to enable the Ivanti Service Manager development project. At the confirmation message, click OK.
5. In the arrow going from the staging instance of the tenant to the UAT instance of the tenant, click Apply Package.
6. Do the following:
   - In the **Source Package** field, select the package to migrate.
   - Check Skip Target Tenant Backup.
   - Select Apply without validation. This allows you to apply the package even if there are errors. You may need to fix the errors after the change is applied, if you do not want to restore the database and fix the root cause.
   - Click Execute.

   The system migrates the package.
7. In the arrow going from the staging instance of the tenant to the UAT instance of the tenant, click Push.
8. Do the following:
   - In the **Operation** field, select Copy Master Data Only.
   - Check Delete data items in uat that do not exist in staging.
   - Check Service request.
   - Click Execute.

   The system migrates all of the service requests.

Making Changes to the Request Offering in the Configuration Console

If the target tenant instance does not have an equivalent request offering, (for example, if you are importing a package from the App Store or a package from another system), you must manually link the configuration item service, organizational unit, and location.

1. Log into the Service Desk Console on the UAT instance of the tenant.
2. Open the **Request Offering** workspace.
3. Find and open the existing request offering.
4. On the **1. Define request offering** page, for the **Service** field, select the configuration item service associated with this request offering. Note that in many cases, the configuration item service is represented by a GUID and not by its name in the drop-down list. Be sure to select the correct configuration item service.

5. Click Next twice until you reach the **4. Publish Action Access** page.
6. Click Add to add a new organizational unit and location. The system displays a new line.
7. Click Any under the **Team** header and select the organizational unit associated with this request offering. Click OK. Note that in many cases, the organizational unit is represented by a GUID and not by its name in the drop-down list. Be sure to select the correct organizational unit.
8. Click Any under the **Location** header and select the location associated with this request offering. Click OK. Note that in many cases, the location is represented by a GUID and not by its name in the drop-down list. Be sure to select the correct location.
9. Click Save & Exit.

Testing the Change

1. Log into the UAT instance of the tenant and test the changes.
2. If anything is wrong, perform the procedures in both **Packaging the Changes Using Ivanti Service Manager** and **Migrating the Data from the Staging Instance to the UAT Instance**.
3. When you are finished testing the change and are satisfied with the results, go back to the staging instance of the tenant and close the package. This ensures that no further changes in the system are added to the package.

Migrating the Data from the Staging Instance to the Production Instance

1. Log into the Operations Console.
2. From the **Tenants** tab, find the instance to migrate and click Manage Migration.
3. At the top of the page, click Enable to enable the Ivanti Service Manager development project. At the confirmation message, click OK.
4. In the arrow going from the staging instance of the tenant to the production instance of the tenant, click Apply Package.
5. Do the following:
    - In the **Source Package** field, select the package to migrate.
    - Ensure that Skip Target Tenant Backup is not checked.
    - Select Apply without validation. This allows you to apply the package even if there are errors. You may need to fix the errors after the change is applied, if you do not want to restore the database and fix the root cause.
    - Click Execute.

    The system migrates the package.

Migrating the Service Request Data from the UAT Instance to the Production Instance

You must publish the service request master data from the UAT instance of the tenant to the production instance of the tenant because the UAT instance has the tested service request data. (If you are sure that no changes were made to the staging instance of the tenant, you can push the service request data from there to the production instance of the tenant instead.)

1. Log into the Operations Console.
2. From the **Tenants** tab, find the instance to migrate and click Manage Migration.
3. In the arrow going from the UAT instance of the tenant to the production instance of the tenant, click Push.

If there is no **Push** link in the arrow, you must push the data from the staging instance to the UAT instance. In the **Operation** field, select No Op (Increase version only) and click Execute.

4. Do the following:
    - In the **Operation** field, select Copy Master Data Only.
    - In the **Additional Data** section, select any or all of Service (CI.Service), Organizational Unit (OrganizationalUnit), and Location (Location).
    - Check Skip target tenant backup before operation.
    - Check Delete data items in production that do not exist in uat.
    - Click Execute.

The system migrates all of the service requests.

8.5. Troubleshooting and Known Issues

- Debugging the Migration
- Known Issues in the Operations Console

Debugging the Migration

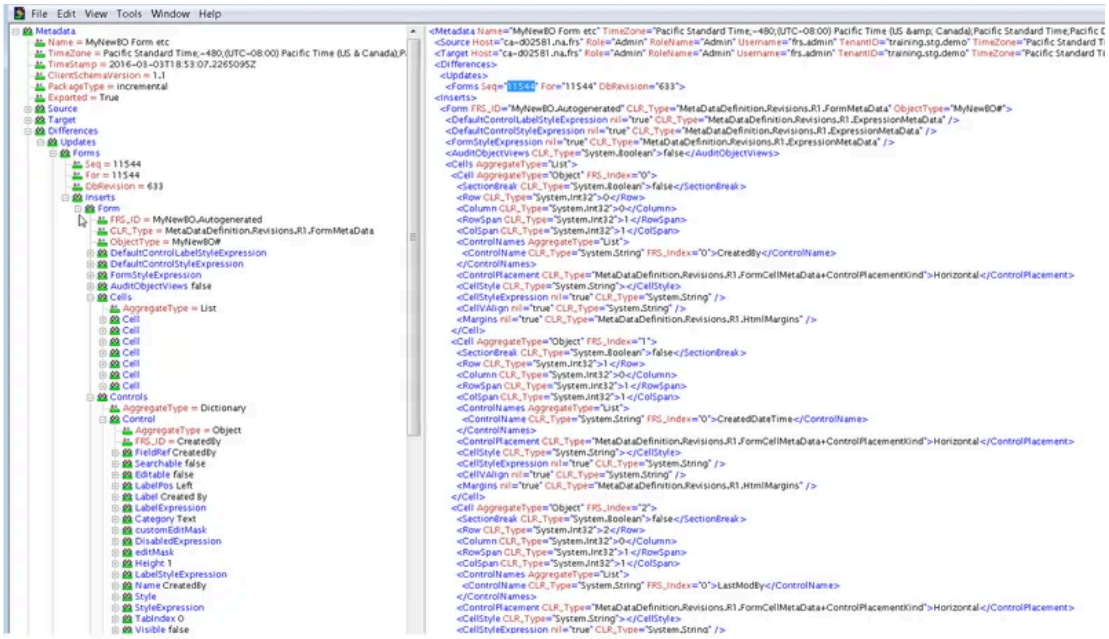
There are several ways to debug the data migration:

- Previewing the Package Before Migrating It
- Viewing the Warning and Error Messages
- Reviewing the Package Content
- Viewing the Operations Console Log Files
- Running Packages in the Correct Order
- Checking the API Keys

Previewing the Package Before Migrating It

If you used the Ivanti Service Manager development project, as described in Appendix C: Using the ISM Development Project to Migrate Data, you can preview the changes to ensure that they are correct. On the **Applying Package To** screen, click Preview. The system downloads an XML file with the proposed changes. Open the file in an XML editor and ensure that the changes are correct. The system associates all warnings to a change record by using a unique sequence number.

Example of XML File with Proposed Changes

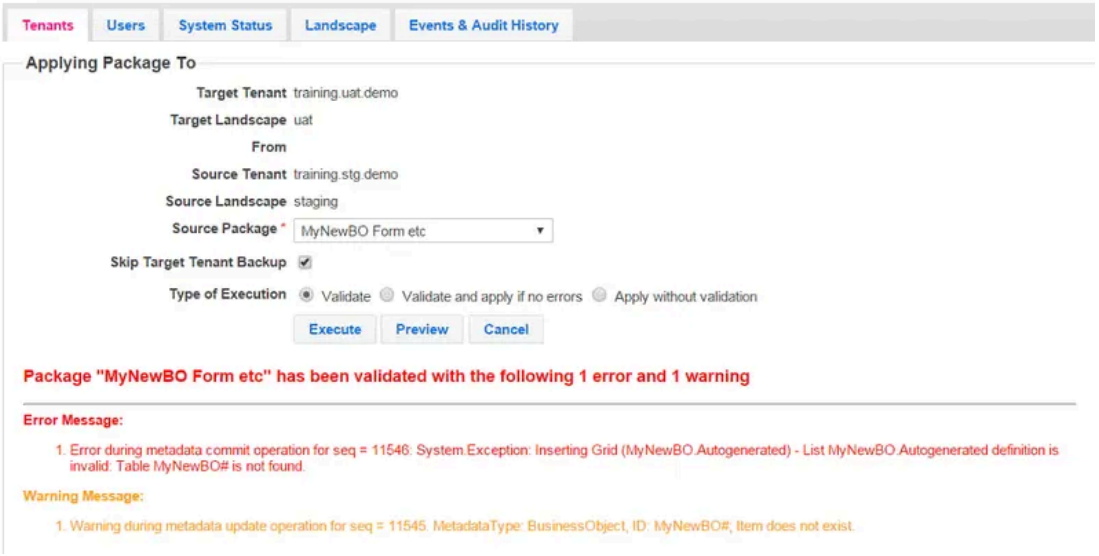


Viewing the Warning and Error Messages

After you perform the first migration from the production instance of the tenant to the staging instance of the tenant, review the error messages. You can review them in any of the following places:

- The **Event and Audit History** tab in the Operations Console. See Viewing Events and Audit History.
- The patch log in the Ivanti Service Manager configuration database (ConfigDB).
- The patch log in the Ivanti Service Manager target tenant instance. See the Ivanti Service Manager online help for information.
- If you used the Ivanti Service Manager development project, as described in Appendix C: Using the ISM Development Project to Migrate Data, the system displays errors and warnings on the bottom of the **Applying Package To** screen.

Sample Errors for a Package



Reviewing the Package Content

Review the package content:

- This is located in a file called *source-tenant-timestamp*.MetadataPatch.zip which resides in the c:\Temp\OpsConsoleDiff directory on the server on which the Operations Console is installed. This package file lists the contents of the migration.
- Check the **Transaction Details** workspace in Ivanti Service Manager. This workspace lists the details of the package migrations.

Viewing the Operations Console Log Files

Find and open the log files from the source tenant instance and the target tenant instance and compare the metadata.

Running Packages in the Correct Order

If you used the Ivanti Service Manager development project, as described in Appendix C: Using the ISM Development Project to Migrate Data, and your migration contains multiple packages, ensure that you migrate the packages in the correct order. If items must be migrated in a specific order and you do not follow that order, the system may display an error. To avoid this, we recommend creating a master package that contains the incremental packages in the correct order.

Checking the API Keys

Check the API keys. Each API key must be unique and cannot be shared between tenant instances. See About System Information.

Known Issues in the Operations Console

This section contains information about issues and limitations in the Operations Console.

- Copying Child Business Objects
- Migrating Logos
- Five Minute Timeout Causes Problems When Migrating Data in a Load-Balancing Deployment
- Operations Console Does Not Work with Microsoft Internet Explorer Releases 8 and 9
- Data Not Showing in Microsoft Internet Explorer Release 11
- There is No Push Link
- Cannot Migrate Data if a Business Object Contains More than 100,000 Records
- Wrong Session Key Format Error
- Cannot Validate or Duplicate the API Key
- Report Template and Schedule Issues
- Some Dashboard and Saved Search Properties are Not Migrated
- Duplicate Data or Database Constraint Errors
- Problems Creating Packages
- No Tenants in the Operations Console After Upgrading

Copying Child Business Objects

Symptom

When copying data from the staging or UAT instance of a tenant, selecting a business object does not copy the data of its child business objects.

Affected Release

All releases.

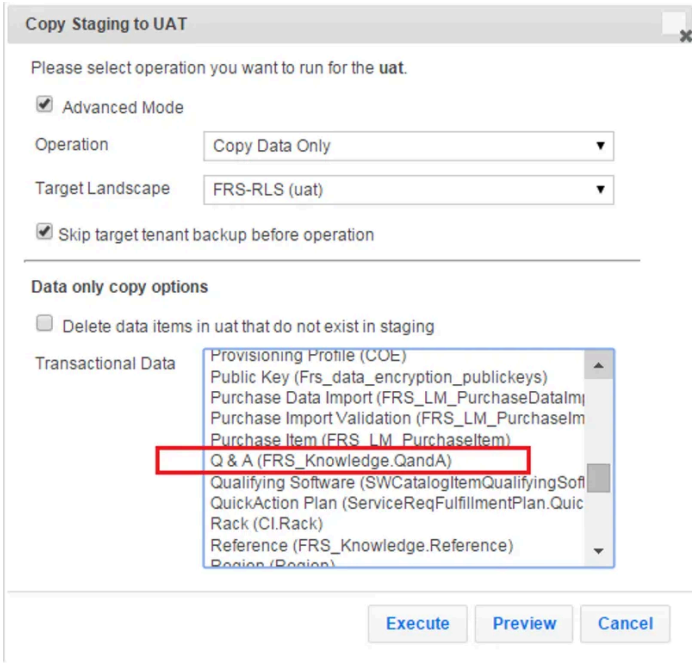
Resolution

To copy the data for a child business object, you cannot simply select its parent business object. You must select the child business object.

For example, to copy all of the knowledge articles in the **QandA** category, you cannot select only the **FRS_Knowledge** business object. You must select the child business object, which in this case is **FRS_Knowledge.QandA**.

The configuration items are listed in the drop-down list by their display names. See below for an example.

Configuration Items



Migrating Logos

Symptom

When migrating data that includes a logo, the updated logo does not migrate.

Affected Release

All releases.

Resolution

The Operations Console does not touch the logo during data migration. It does not remove the logo, but it also does not update the logo. Therefore, if you change the logo and then migrate data, after the data migration you must manually upload the new logo.

Also, changing the logo is not something that gets tracked in Ivanti Service Manager. Therefore, you cannot use the package process for this item.

Five Minute Timeout Causes Problems When Migrating Data in a Load-Balancing Deployment

Symptom

When migrating data on a system that uses a load-balancing deployment, the system is usually set to time out after 5 minutes for security reasons, but often it takes more than 5 minutes to migrate data. This causes the data migration to fail and the system displays an error message stating "Unable to retrieve metadata from *tenant_name* due to the underlying connection was closed."

Affected Release

All releases.

Resolution

You can configure the Operations Console to go directly to the Ivanti Service Manager application server without going through the load balancer. To do this, go to the file called C:\Windows\System32\drivers\etc\hosts and add the host name for the multi-instance service URL. You can also use another port number for the multi-instance service URL in the Ivanti Service Manager application server. See **Editing a Landscape** for more information about the multi-instance service URL.

You can also change the timeout setting in your network, but we do not recommend doing this.

Operations Console Does Not Work with Microsoft Internet Explorer Releases 8 and 9

Symptom

The Operations Console does not work with Microsoft Internet Explorer Release 8 or Release 9 because they do not support the latest third-party Javascript code.

Affected Release

All releases.

Resolution

Upgrade to a newer release of Microsoft Internet Explorer or to another browser.

Data Not Showing in Microsoft Internet Explorer Release 11

Symptom

The Operations Console does not work in Microsoft Internet Explorer Release 11 due to a new security feature in Microsoft Internet Explorer Release 11.

Affected Release

All releases.

Resolution

Add the URL of the Operations Console to the trusted sites list in Microsoft Internet Explorer Release 11. Follow these steps:

1. Go to Tools > Internet Options > Security.
2. Highlight Trusted sites and click Sites.
3. Click Add.

There is also a known issue with Microsoft Internet Explorer Release 11 and the French operating system. If you run into this problem when using the French operating system, clear the Microsoft Internet Explorer browser cache.

There is No Push Link

Symptom

There are three scenarios where this happens:

1. There is no **Push** link between the staging instance of the tenant and the production instance of the tenant if the production instance of the tenant is not at version 0.
2. There is no **Push** link between the UAT and production instances of the tenant after migrating data from UAT to production.
3. There is no **Push** link between the UAT and production instances of the tenant after you apply a closed package from the staging instance of the tenant to the production instance of the tenant.

You can only migrate data from one instance to another if there is a different version number. The system automatically increases the version number after a successful migration.

In a three-tier landscape group, you can only migrate data from staging to production when the production instance of the tenant is at version 0.

With a two-tier landscape group, the migration path is only between two tenant instances, usually the production instance of the tenant and the staging instance of the tenant. With a three-tier landscape group, there are migration paths between three tenant instances, usually the production instance of the tenant, the staging instance of the tenant, and the UAT instance of the tenant.

We recommend always using a three-tier landscape group unless you have a demo environment.

Affected Release

All releases.

Resolution

For the first scenario, we recommend temporarily switching from a three-tier landscape group to a two-tier landscape group to immediately migrate data between the staging instance of the tenant and the production instance of the tenant. See **Editing a Landscape Group** for information about doing this. After the migration, switch back to a three-tier landscape group.

For the second and third scenarios, you must increase the version of the UAT instance of the tenant. To do this, click Push and in the **Operation** field, select No Op (Increase version only). This increases the version of the UAT instance of the tenant so that the **Push** link appears. See **Increasing the Version Number (No Op [Increase Version Only])**.

Cannot Migrate Data if a Business Object Contains More than 100,000 Records

Symptom

The Operations Console cannot migrate database tables with more than 100,000 records. In some deployments, the **ServiceReqParam** table can have more than 100,000 records.

Affected Release

All releases.

Resolution

We have removed the **ServiceReqParam** tables so they are no longer part of the service request copy in the Operations Console.

We recommend as a best practice that your database tables be smaller than 100,000 records. We also recommend that if you do have a table with more than 100,000 records, you do not migrate it using the Operations Console.

If you do have a table with more than 100,000 records that needs to be migrated, use Microsoft SQL Server commands to manually copy the data and its relationships, or break the table into smaller tables, migrate the smaller tables, and then reform the data into a larger table after the migration.

Wrong Session Key Format Error

Symptom

When migrating data with the Operations Console, the system gives a "Wrong Session Key Format" error.

Affected Release

All releases.

Resolution

The system gives this error when you copy the database from another server and restore it for a tenant without using the push action in the Operations Console to set up the tenant. To solve this, migrate the data using the Operations Console using the copy to replace with database already restored option.

You can also use Microsoft SQL Server to truncate the table using this command: truncate table Frs_ops_session_secondary_key;.

Cannot Validate or Duplicate the API Key

Symptom

When migrating data or adding a new tenant with the Operations Console, the system gives an error message about being unable to validate the API key.

Affected Release

All releases.

Resolution

Check that all active tenants in the Ivanti Service Manager configuration database have a valid database and proper login credentials.

Make sure that none of the tenants has duplicate API keys by logging into the Operations Console, clicking the **System Status** tab, and looking for duplicate API key entries. If you see a duplicate entry, click Fix to fix the issue. See **Fixing Problems** for more information.

Report Template and Schedule Issues

Symptom

The Operations Console does not migrate out-of-the-box report templates (.rdl files) if the source and target tenants are on different servers.

The Operations Console does not migrate custom-created report templates (.rdl files) even if the source and target tenants are on the same server. (These report templates are stored in the Report Server, which is specific to the tenant.)

The Operations Console does not migrate the report schedules, report subscriptions, or report distributions for migrated reports.

The system does not provision a report when its report template is not out-of-the-box (that is, if it is a custom report template), or if the report template has not been migrated.

Affected Release

All releases.

Resolution

The Operations Console does migrate new report definitions, regardless of if there is a template or not.

The Operations Console does migrate updated report definitions. You can overwrite the target if the source definition is older than the target definition.

You can delete report definitions in the target tenant if it does not exist in the source tenant.

The system provisions the new report only if the report template is out-of-the-box, or if the report template exists for the tenant on the Report Server.

To perform all other report migration and provisioning tasks, use the Ivanti Service Manager reporting service.

Some Dashboard and Saved Search Properties are Not Migrated

Symptom

Some properties (such as "Set as default for myself", "Default for ...", and "My default") for dashboards and saved searches are not migrated.

Affected Release

All releases.

Resolution

These "default" properties are not metadata and are therefore not copied.

Duplicate Data or Database Constraint Errors

Symptom

When you migrate data, the system gives this error:

> Error during metadata commit operation: DataLayer.ConstraintViolationException: DB constraint violation exception: Violation of PRIMARY KEY constraint .....
>
> Error during metadata commit operation: DataLayer.ConstraintViolationException: DB constraint violation exception: Violation of UNIQUE KEY constraint .....

Affected Release

All releases.

Resolution

This happens when you add or edit data in both the source and target instance of the tenant. When you use the Operations Console to migrate data, you cannot add or edit data on both the source instance of the tenant and the target instance of the tenant.

If you try to migrate data after adding or editing data on both the source instance of the tenant and the target instance of the tenant, the system duplicates the records. You must ignore the error message and fix the errors manually.

Problems Creating Packages

Symptom

When creating packages, it can be difficult to create the correct package for the migration. It can be challenging to find the right pieces of metadata to package and to find all of the dependencies.

Affected Release

All releases.

Resolution

- Before making any changes, always add a Ivanti Service Manager development project.
- Before making any changes, select a project. It is easier for you to include all changes in a project to a package.
- Review the transaction details to find and review transaction sets. You can search based on the object type, business object, entity name, and the name of the person who made the change. You can then add the transaction sets to the package.
- You may need to do a few iterations before you get the right package.

No Tenants in the Operations Console After Upgrading

Symptom

After upgrading Ivanti Service Manager, the system updated the connection strings for the landscape with the incorrect name for the data source.

Affected Release

All releases.

Resolution

In the Operations Console, edit each landscape to update the data source to the correct database server name. See **Editing a Landscape** for more information on how to do this.

8.5.1. Configuring Landscapes

For a definition of landscape and common landscape types, see **Landscapes**.

- **About the Landscape Workspace**
- **Working with Landscape Groups**
- **Working with Landscapes**
- **Working with Database Servers**

About the Landscape Workspace

Use the **Landscape** workspace to perform a variety of landscape management activities, including creating and configuring landscape groups, landscapes, and database servers. You must create a landscape group before you can create a landscape, and you must create a landscape before you can create any tenants.

Working with Landscape Groups

- **About Landscape Groups**
- **Adding a Landscape Group**
- **Editing a Landscape Group**
- **Deleting a Landscape Group**

About Landscape Groups

A landscape group is a group comprised of different landscapes. The following are examples of landscape groups:

- Independent landscapes group: The production, staging, and UAT landscapes all have different Ivanti Service Manager configuration databases. Each landscape is on a separate server which makes it easier to independently shut down or restart the servers, if needed. However, this option requires more hardware, which means you need additional licenses.
- Shared landscapes group: The production landscape has its own Ivanti Service Manager configuration database and the staging and UAT landscapes share the same Ivanti Service Manager configuration database. This makes it easier to perform upgrades because you have to upgrade the staging and UAT landscapes at the same time. (All landscapes must use the same release of Ivanti Service Manager in order to migrate data.) This is typically how Ivanti Service Manager Cloud systems are deployed.
- All-in-one landscape group: The production, staging, and UAT landscapes all share the same Ivanti Service Manager configuration database and servers. This deployment is used mainly by testing and development teams or systems with limited resources.

We recommend using either an independent or shared landscape group. However, there are advantages and disadvantages to each landscape group. See the *Installation and Deployment Guide for Ivanti Service Manager* for more information about the different deployment types.

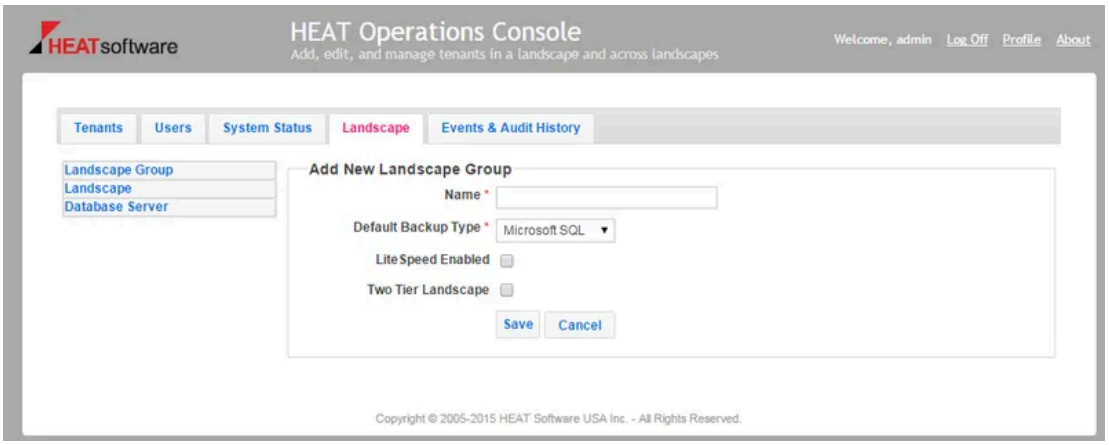
> We highly recommend that you only install the Operations Console in the staging landscape, and not in any other landscape. For migrating data, you only need to install the Operations Console in one landscape; you should not install it in more than one landscape.

Adding a Landscape Group

You must add a landscape group before you can add any landscapes. To add a landscape group, do the following:

1. Log in to the Operations Console.
2. Click the **Landscape** tab. The system displays the **Landscape** workspace.
3. Click Landscape Group on the left. The system displays all of the landscape groups in the system.
4. Click Add Landscape Group. The system displays the **Add New Landscape Group** page. See Add New Landscape Group Page.

Add New Landscape Group Page



5. Enter a name for the landscape group.
6. Choose the default backup type. You can either select Microsoft SQL or LiteSpeed. Note that LiteSpeed is not supported for Ivanti Service Manager on-premise customers.
7. If you selected LiteSpeed as the default backup type, you must check LiteSpeed Enabled. Otherwise, leave it empty.

> Ivanti Service Manager supports using LiteSpeed for Microsoft SQL Server for backing up your Ivanti Service Manager databases. Before you can use LiteSpeed for Microsoft SQL Server, you must purchase it, install it on your system, and configure the Operations Console to support it.

8. Ensure that Two Tier Landscape is not checked. With a two-tier landscape group, the migration path is only between two tenant instances, usually the production instance of the tenant and the staging instance of the tenant. If you do not check th is option, you have a three-tier landscape group. With a three-tier landscape group, there are migration paths between three tenant instances, usually the production instance of the tenant, the staging instance of the tenant, and the UAT instance of the tenant.

> We recommend always using a three-tier landscape group unless you have a demo environment.

9. Click Save.

Editing a Landscape Group

1. Log in to the Operations Console.
2. Click the **Landscape** tab. The system displays the **Landscape** workspace.
3. Click Landscape Group on the left. The system displays all of the landscape groups in the system.

4. Do one of the following:
    - Click the underlined name of the landscape group. In the resulting dialog box, click Edit.
    - Click the pencil icon to the right of the landscape group name.
5. Edit the values.
6. Click Save.

Deleting a Landscape Group

1. Log in to the Operations Console.
2. Click the **Landscape** tab. The system displays the **Landscape** workspace.
3. Click Landscape Group on the left. The system displays all of the landscape groups in the system.
4. Click the red X to the right of the landscape group name to delete it.
5. In the resulting dialog box, click OK.

Working with Landscapes

- About Landscapes
- Adding a Landscape
- Editing a Landscape
- Deleting a Landscape

About Landscapes

A landscape consists of a Ivanti Service Manager configuration database and two Ivanti Service Manager web servers for connecting to the Ivanti Service Manager application server.

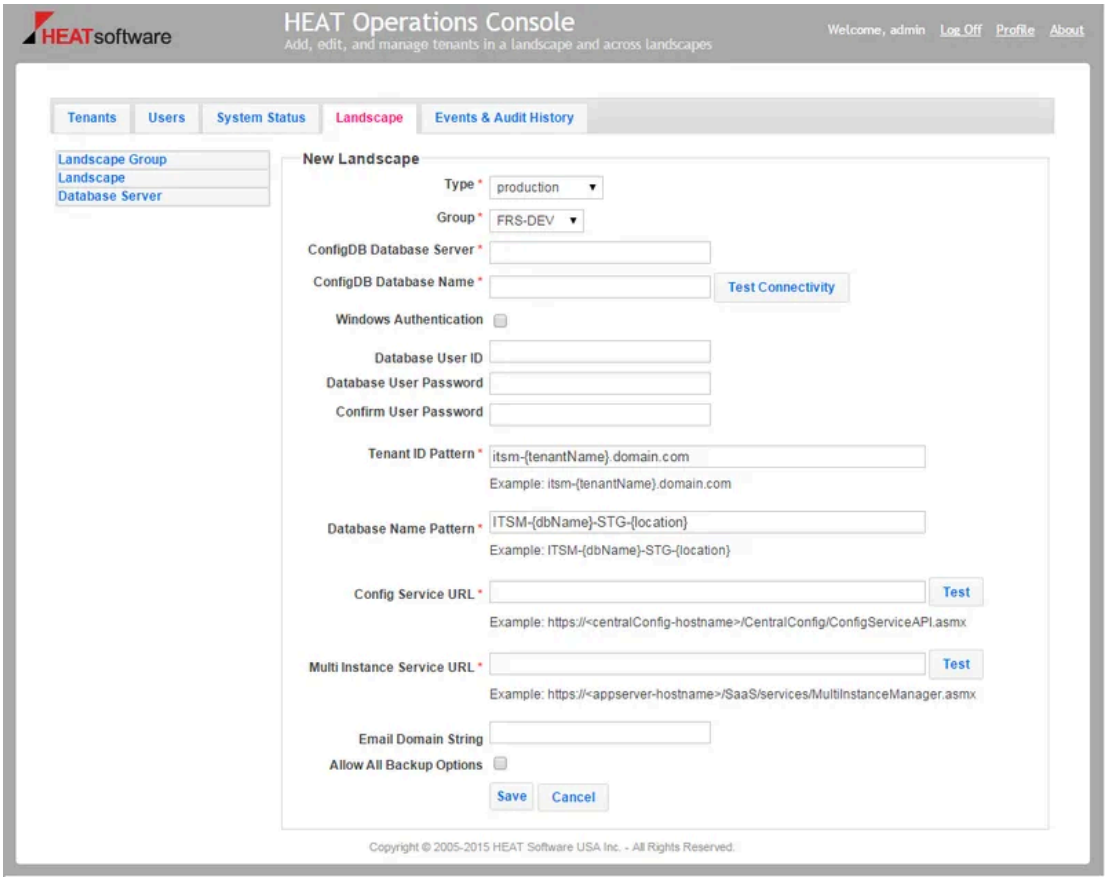
After you create a landscape group, you can start creating landscapes to add to it. You can have a maximum of one of the following landscape types in a landscape group:

- Production
- Staging
- UAT
- Development
- Pilot

Adding a Landscape

1. Log in to the Operations Console.
2. Click the **Landscape** tab. The system displays the **Landscape** workspace.
3. Click Landscape on the left. The system displays all of the landscapes in the landscape group.
4. (Optional) To view the landscapes for a different landscape group, select a different landscape group from the drop-down list and click Go. The system displays all of the landscapes in the landscape group.
5. Click Add Landscape. The system displays the **New Landscape** page. See New Landscape Page.

New Landscape Page



5. Select a landscape type from the drop-down list.
6. Select the landscape group to which this landscape will be attached.
7. Enter the name of the Ivanti Service Manager configuration database server.
8. Enter the name of the Ivanti Service Manager configuration database.
9. Click Test Connectivity to test the connection from the Operations Console to the Ivanti Service Manager configuration database.
10. If you are using Windows Integrated Security, check Windows Authentication.

> If you check Window Authentication, the system hides the **Database User ID**, **Database User Password**, and **Confirm User Password** fields. These fields are only displayed if you do not check Window Authentication, which means that you are using Microsoft SQL authentication.

11. If you did not check Window Authentication, enter values into the **Database User ID**, **Database User Password**, and **Confirm User Password** fields.
12. Enter a tenant ID pattern. This is the format in which all tenants in this landscape will be named. You can optionally use the variable as a placeholder for the tenant name. The following are examples:
    - for the production instance of the tenant.
    - .stg for the staging instance of the tenant.
    - .uat for the UAT instance of the tenant.

For example, if the value of the parameter is demo1, then for the production landscape, the tenant ID is demo1 and the URL is http://demo1/HEAT; for the staging landscape, the tenant ID is demo1-STG and the URL is http://demo1-STG/HEAT; and for the UAT landscape, the tenant ID is demo1-UAT and the URL is http://demo1-UAT/HEAT.

13. Enter a database name pattern. This is the format in which all databases in this landscape will be named. You can optionally use the variable as a placeholder for the database name. The following are examples:
    - for the production instance of the tenant.
    - -STG for the staging instance of the tenant.
    - -UAT for the UAT instance of the tenant.

For example, if the value of the parameter is TRAINING, then the database name for the production landscape is TRAINING; the database name for the staging landscape is TRAINING-STG; and the database name for the UAT landscape is TRAINING-UAT.

14. Enter the URL for the Ivanti Service Manager configuration service, which is the location of the Ivanti Service Manager configuration database. The value must be in the format http[s]://*Central_ConfigDB_hostname*/CentralConfig/ConfigServiceAPI.asmx. Click Test to ensure that Ivanti Service Manager can contact the URL.
15. Enter the URL for the multi-instance service. This is the web service URL for connecting to a tenant. The value must be in the format http[s]://*App_Server_hostname*/HEAT/services/MultiInstanceManager.asmx. Click Test to ensure that Ivanti Service Manager can contact the URL.
16. Enter the email domain string for your organization.
17. Check Allow All Backup Options to allow all backup options for tenant and tenant instances that are part of this landscape. See **Backup Types** for more information about database backup types.
18. Click Save.

Editing a Landscape

1. Log in to the Operations Console.
2. Click the **Landscape** tab. The system displays the **Landscape** workspace.
3. Click Landscape on the left. The system displays all of the landscapes in the landscape group.
4. (Optional) To view the landscapes for a different landscape group, select a different landscape group from the drop-down list and click Go. The system displays all of the landscapes in the landscape group.
5. Do one of the following:
    - Click the underlined name of the landscape. In the resulting dialog box, click Edit.
    - Click the pencil icon to the right of the landscape name.
6. Edit the values.
7. Click OK.

Deleting a Landscape

1. Log in to the Operations Console.
2. Click the **Landscape** tab. The system displays the **Landscape** workspace.
3. Click Landscape on the left. The system displays all of the landscapes in the landscape group.

4. (Optional) To view the landscapes for a different landscape group, select a different landscape group from the drop-down list and click Go. The system displays all of the landscapes in the landscape group.
5. Click the red X to the right of the landscape name.
6. In the resulting dialog box, click OK.

Working with Database Servers

- About Database Servers
- Adding a Database Server
- Editing a Database Server
- Deleting a Database Server

About Database Servers

Each landscape must have one or more Ivanti Service Manager database servers associated with it to host the tenant instance databases. The Ivanti Service Manager database server is controlled by the Ivanti Service Manager configuration database. You must specify the data and backup locations for the database server before you can add a tenant using the Operations Console.

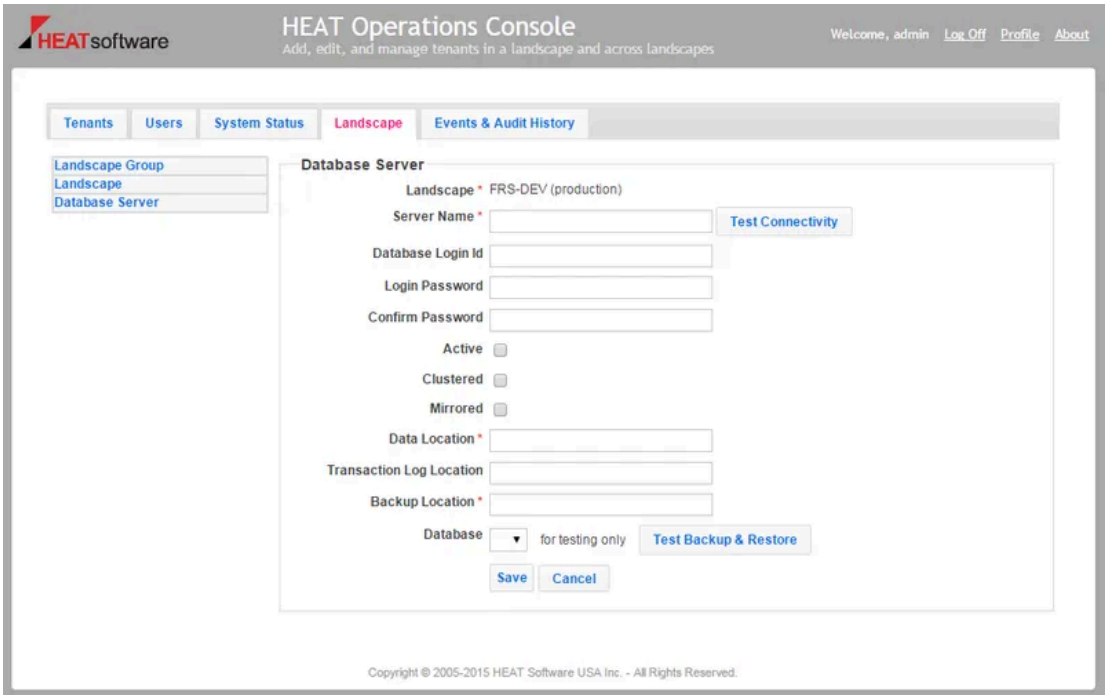
Ivanti Service Manager database servers either use Windows Integrated Security or by Microsoft SQL Server authentication, in which case you must enter a user name and password. You define which authentication model to use when you initially install Ivanti Service Manager using the System Configuration Wizard.

Adding a Database Server

To add a database server, do the following:

1. Log in to the Operations Console.
2. Click the **Landscape** tab. The system displays the **Landscape** workspace.
3. Click Database Server on the left. The system displays all of the database servers in the landscape.
4. (Optional) To view the database servers for a different landscape, select a different landscape from the drop-down list and click Go. The system displays all of the database servers in the landscape.
5. Click Add Database Server. The system displays the **Database Server** screen as shown in Add a Database Server Screen.

Add a Database Server Screen



5. Enter a name for the database server. This is usually the host name, IP address, or machine name.
6. Enter a database login ID and password. The database login ID must have permission to back up any tenant or tenant instance database; must have permission to restore any tenant or tenant instance database; and must become the database owner of any new databases that are created on the tenant or tenant instance. The database login ID must have read and write access to the data location and backup locations.
7. Click Test Connectivity to test the connectivity between the Operations Console and the Ivanti Service Manager databases.

If you are using Windows Integrated Security (as opposed to Microsoft SQL authentication), the **Database Login Id**, **Login Password**, and **Confirm Password** fields are not shown. These fields are only displayed if you are using Microsoft SQL authentication.

8. Do not check Active, Clustered, or Mirrored. These are internal settings that should not be used.
9. Enter a location for the database server data. Do not enter a UNC path. Change the application pool identity so that it has read access to the data location.
10. Enter the location of the transaction log. Do not enter a UNC path. Change the application pool identity so that it has read access to the transaction log location.
11. Enter a location for the backup data for the database server. This is used for the automatic transfer of the database backup files from the source Microsoft SQL database server to the target Microsoft SQL database server.
    - If you have more than one database server, each database server must have its own backup data location, but it must be a shared location that is dedicated for Ivanti Service Manager backups. Use a UNC path if you are using a shared location. Do not use the Microsoft SQL Server default backup location if you have more than one database server.
        - Change the application pool identity so that it has read and write permissions on both the **Sharing** and **Security** tabs in the Windows Explorer **Properties** dialog box for the shared backup location.
        - The identity (user) that the database server runs must have read and write permissions on both the **Sharing** and **Security** tabs in the Windows Explorer **Properties** dialog box for the shared folder used for the database backup location of the given database server.

        - Change the application pool identity to an account that has read and write permission to the backup data location.
    - If all of the landscapes share the same database instance, you can use one backup data location.
12. To test the backup and restore function, select a database from the drop-down list. Click Test Backup & Restore. This may take some time.

> - The drop-down list is only populated if you successfully tested the connection as shown in step 8 above.
> - Select a small database, such as the Ivanti Service Manager configuration database, to quickly test the user's permission to back up and restore, to ensure that the backup location is set with the correct permissions, and to ensure that the data location for creating the database is correct.

13. Click Save.

Editing a Database Server

1. Log in to the Operations Console.
2. Click the **Landscape** tab. The system displays the **Landscape** workspace.
3. Click Database Server on the left. The system displays all of the database servers in the landscape.
4. (Optional) To view the database servers for a different landscape, select a different landscape from the drop-down list and click Go. The system displays all of the database servers in the landscape.
5. Do one of the following:
    - Click the underlined name of the database server. In the resulting dialog box, click Edit.
    - Click the pencil icon to the right of the database server name.
6. Edit the values.
7. Click Save.

Deleting a Database Server

> You cannot delete a database server that is in use.

1. Log in to the Operations Console.
2. Click the **Landscape** tab. The system displays the **Landscape** workspace.
3. Click Database Server on the left. The system displays all of the database servers in the landscape.
4. (Optional) To view the database servers for a different landscape, select a different landscape from the drop-down list and click Go. The system displays all of the database servers in the landscape.
5. Click the red X to the right of the database server name.
6. In the resulting dialog box, click OK.

8.5.2. Working with Tenants and Tenant Instances

- About the Tenants Workspace
- How to Create a New Tenant
- Editing a Tenant

About the Tenants Workspace

Ivanti Service Manager uses the concept of tenants so that it can update a system database while not touching the official version of the database. The official version of a database is always located on the production instance of the tenant.

If you need to make changes to the official system database, you do not want to make the changes realtime to a live version because you might make a mistake and accidentally delete everything. Therefore, the Operations Console allows you to make a copy of the production tenant, called an instance, and make changes there. When the database changes have been reviewed and approved, only then are they moved back to the production instance of the tenant.

Use the **Tenants** workspace to perform a variety of tenant and tenant instance management activities, including creating, locking, migrating, and deleting tenants and tenant instances.

- Workspace Filter Controls
- Tenant Display Order Controls
- Expanded View and List View

Workspace Filter Controls

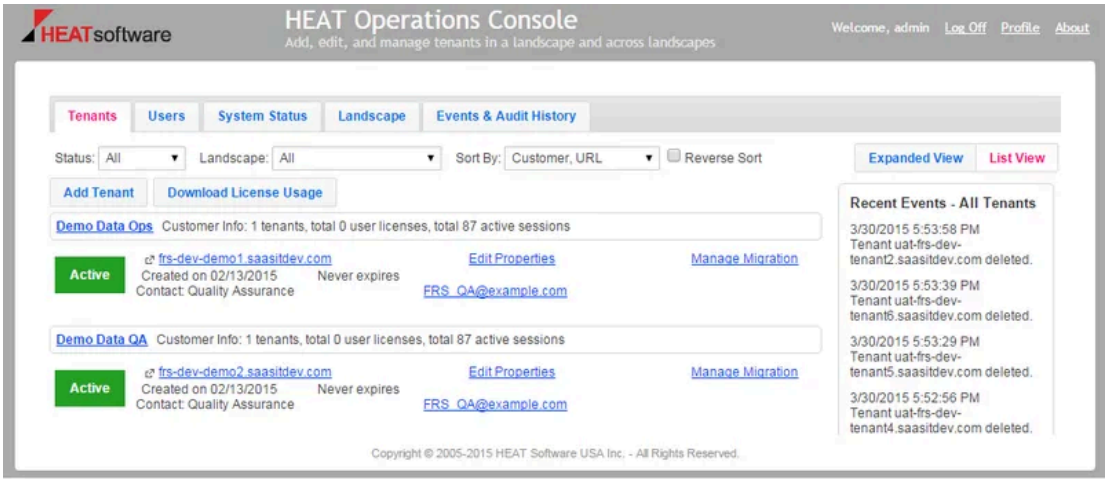
Use the **Status** and **Landscape** drop-down boxes to filter which tenants and tenant instances are displayed in the workspace:

- Status: Displays tenants and tenant instances based on their status. You can select **All**, **Pending**, **Active**, **OnHold**, **Closed**, or **Demo**.
- Landscape: Displays tenants and tenant instances from a landscape type, such as production, or you can choose **All** to display tenants and tenant instances from all landscapes.

> If you do not see as many tenants or tenant instances as you expect when you first open the **Tenants** workspace, select **All** from both the **Status** and **Landscape** drop-down boxes to display all of the tenants and tenant instances on your system.

See Tenants Expanded View.

Tenants Expanded View



Tenant Display Order Controls

Use the **Sort By** field and the **Reverse Sort** button to configure the order in which tenants and tenant instances are displayed:

- Sort By: Select whether to sort by customer and then URL (Customer, URL option); customer and then status (Customer, Status option); or by customer and then expiration date (Customer, Exp Date option) as the tenant and tenant instance display order.
- Reverse Sort: Check here to display tenants and tenant instances in reverse alphabetical or numerical order.

See Tenants Expanded View.

Expanded View and List View

Use the **Expanded View** and **List View** buttons to switch between expanded and list views. The name of the currently selected view is shown in a pink font.

- Expanded View
- List View

Expanded View

Use the **Expanded View** to display the maximum amount of information for each tenant and tenant instance. This information is displayed:

- Tenant or tenant instance name (for example, ConfigDB, Server_01, etc.).
- Ivanti Service Manager database name. This is sometimes the same as the tenant or tenant instance name (for example, ConfigDB), depending on how the tenant or tenant instance was originally configured.
- Tenant or tenant instance creation and expiration dates.
- The name and contact information of the primary contact for the tenant or tenant instance.
- The number of licenses, sessions, and timeout minutes for each tenant or tenant instance.
- A log of recent tenant or tenant instance changes.

You may also see some or all of these links, depending on how the tenant or tenant instance is configured:

- **Edit Properties**: Opens the **Edit** dialog box for the tenant or tenant instance where you can set these parameters:
  - Company name.
  - Status of the tenant or tenant instance (for example, active, pending, on hold, closed, and demo).

    - Put the tenant or tenant instance into maintenance mode.
    - Lock metadata.
    - Tenant or tenant instance expiration date.
    - The number of named and concurrent licenses.
    - The session timeout value.
    - The person who is the point of contact for matters involving the tenant or tenant instance.
- **Manage Migration**: Opens the **Tenants** workspace, from which you can migrate data between tenants or tenant instances.
- **Extend Expiration Date**: Opens the **Expiration Date** dialog box, from which you can select a new expiration date for the tenant or tenant instance.
- **Deactivate**: Opens the **Deactivate** dialog box, from which you can deactivate the tenant or tenant instance.
- Clear Cache: Clears the cache of the Ivanti Service Manager application server without having to shut down the application.

List View

Use the **List View** to display a minimal amount of information for each tenant and tenant instance. This information is displayed:

- Tenant or tenant instance name (for example, ConfigDB, Server_01, and so on).
- Tenant or tenant instance creation and expiration dates.
- The name and contact information of the primary contact for the tenant or tenant instance.
- One or both of these links, depending on how the tenant or tenant instance is configured:
    - **Edit Properties**: Opens the **Edit** dialog box for the tenant or tenant instance.
    - **Manage Migration**: Opens the **Tenants** workspace, from which you can migrate data between tenants or tenant instances.

How to Create a New Tenant

When you install Ivanti Service Manager, you automatically create a tenant with a production instance. However, in some situations, you may need to create a new tenant. For example, a customer may have two applications running on one system and each application requires its own tenant.

The following sections describe two ways to create a new tenant:

- Creating a New Tenant by Using the Operations Console
- Creating a Tenant by Copying an Existing Tenant

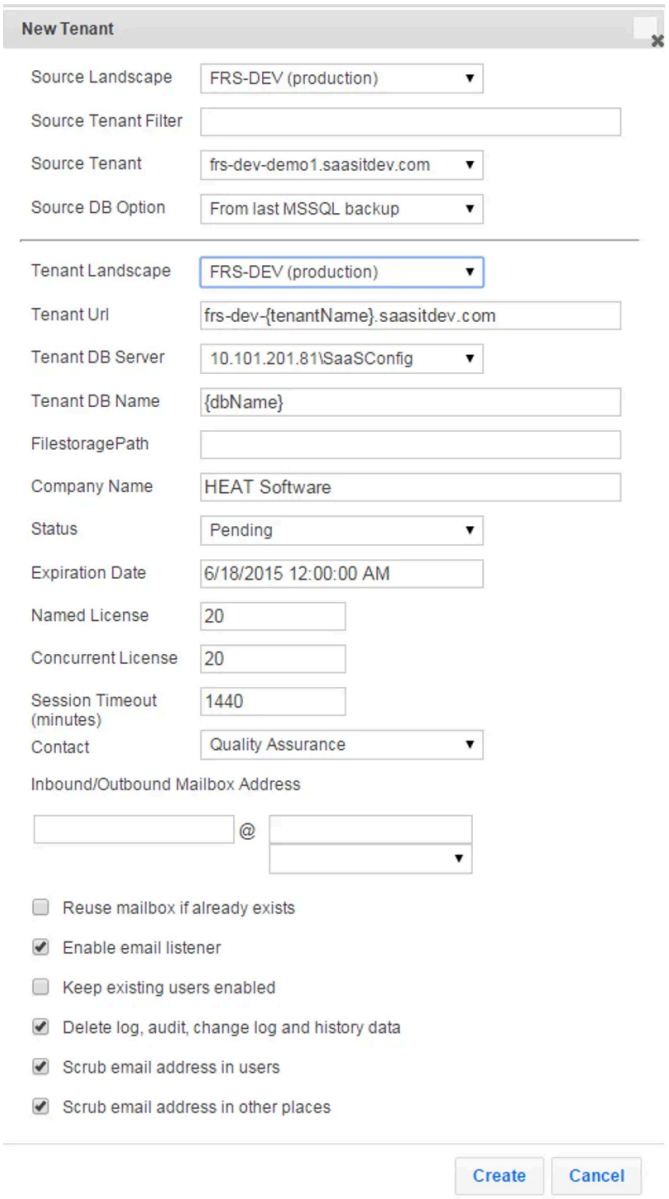
Creating a New Tenant by Using the Operations Console

The easiest way to create a new tenant is to use the Operations Console.

> This process creates a new tenant. It does not create instances, such as staging or UAT, of existing tenants.

1. Log in to the Operations Console.
2. Click Tenant. The system displays the **Tenants** workspace.
3. Click Add Tenant. See Adding a Tenant.

Adding a Tenant



4. Enter information in the fields about the source that will act as a starting point for your new tenant:
    - Choose a source landscape. This is the landscape where the source tenant resides. You can only select production or other.
        - If you select production, you can create a tenant in either the **production** or **other** landscapes.
        - If you select other, you can only create a tenant in the **other** landscape.
    - (Optional) Choose a source tenant filter. This allows you to narrow down the source tenants that the system displays in the **Source Tenant** field below.
    - Choose a source tenant. These are all the tenants in the source landscape that you selected above.
    - Choose a source database option. This is where the database for the source tenant comes from.
        - See **Backup Types** for more information about database backup types.
        - The options you see here depend on if you checked Allow All Backup Options when you created the landscape. If you checked Allow All Backup Options, then you can choose to use live backups when creating a tenant. If you did not

check it, you can only restore your database from a previously backed up version. See **Adding a Landscape** for more information about the **Allow All Backup Options** parameter.

- The options you see here also depend on if you checked LiteSpeed Enabled when you created the landscape group. See **Adding a Landscape Group** for information about the **LiteSpeed Enabled** parameter.

5. Enter information in the fields about the new tenant:
   - Choose a tenant landscape. This is the landscape where the new tenant will reside. You can either choose the production landscape or the other landscape, depending on what you entered in the first bullet of step 4 above.
   - Enter a tenant URL pattern. This is the format in which all tenant URLs will be named. You can optionally use the variable as a placeholder for the tenant name.
   - Choose the tenant database server.
   - Enter a tenant database name pattern. This is the format in which all databases will be named. You can optionally use the variable as a placeholder for the database name.
   - Enter the file storage path.
   - Enter the company name.
   - Enter the tenant status. Can be pending, active, on hold, closed, or demo.
   - Enter the expiration date for the tenant. You can choose a specific date or select never expires. Choose the expiration date carefully. When a tenant has expired, you cannot log in to it anymore. You must contact Ivanti Software to extend the expiration date.
   - Enter the number of named licenses for the tenant.
   - Enter the number of concurrent licenses for the tenant.
   - Enter the session timeout in minutes. If the session is dormant for longer than this time, you must log in to the system again.
   - Choose the primary point of contact for this tenant.
   - Enter the inbound and outbound email address for this tenant. Any domain names that you already entered on the **Adding a Landscape** page are populated in the drop-down menu. You can either choose a domain name from the drop-down menu or you can enter a domain name.
   - Check Reuse mailbox if already exists to reuse the email address if it was already used.
   - Check Enable email listener to enable the email listener.
   - Check Keep existing users enabled to keep the existing users enabled.
   - Check Delete audit, log, change log and history data to delete the data in these tables:
     - Logs
     - Frs_ops_logon_history
     - Frs_ops_gbu_pipeline_history
     - Frs_ops_metadata_history
     - Frs_ops_upload_cache
     - Journal
     - LDAPImportLog
     - Frs_ops_package_item
     - Frs_ops_package

    - All audit tables, with the name Audit_*table*
  - Check **Scrub email address in users** to ensure that you do not accidentally email users repeatedly during configuration testing on the target tenant instance. If you check this option, the system changes all email fields containing customer email to use the @it.com domain instead of the company domain. This change applies to the user's primary email, contact email, manager email, and team email.
  - Check **Scrub email address in other places** to scrub email accounts saved in transactional data (such as in incidents, service requests, workflow tasks, and so on).
  - Check **Trim data to keep last 90 days only** to only copy transactional data that was created or modified within the last 90 days. If you are updating the database from a copy of your existing customer database, most records are likely to be current within 90 days.
  - Check **Scrub email address in users** to ensure that you do not accidentally email users repeatedly during configuration testing on the target tenant. If you check this option, the system changes all email fields containing customer email to use the @it.com domain instead of the company domain. This change applies to the user's primary email, contact email, manager email, and team email.
  - Check **Scrub email address in other places** to scrub email accounts saved in transactional data (such as in incidents, service requests, workflow tasks, and so on).
  - Enter the host name of the report server.
  - Enter the user name and password for the person who will sign in and use the report server.
6. Click Create. The system creates the tenant. This may take two or more minutes. After you create the tenant, you can log in to it.

Creating a Tenant by Copying an Existing Tenant

- About Copying an Existing Tenant
- Option 1: Tenants are in Separate Environments and do not Share Resources
- Option 2: Tenants are in Separate Environments and Share Resources

About Copying an Existing Tenant

Under normal circumstances, you can use the Operations Console to create a new tenant and manage migrations in an automated fashion in a Ivanti Service Manager deployment. In that scenario, all environments are managed as separate tenants on the same installation. That process is described in Creating a New Tenant by Using the Operations Console.

However, there are times when you might need to move or copy a tenant from one landscape (for example, a separate development environment) to another installation (for example, a final production environment). The end result is the creation of a new tenant in the target environment.

> We recommend only using this process for creating templates or baselines of existing tenants. We do not recommend using this process to create a new tenant. Use the process described in Creating a New Tenant by Using the Operations Console to create new tenants.

The following sections describe two similar ways to create a new tenant by copying an existing tenant. In the first scenario (described in Option 1: Tenants are in Separate Environments and do not Share Resources), the source environment and target environment do not share resources. In the second scenario (described in Option 2: Tenants are in Separate Environments and Share Resources), databases and other resources are shared between the source and target environments.

For both options, both the source and destination tenants must be running the same Ivanti Service Manager software release.

> We recommend that you only use this method to *create* a tenant. We do not recommend using this method to update an existing tenant.
> If the tenant already exists in the destination system and you use this method to update that tenant, this method completely overwrites the existing tenant, including transactional data. Therefore, this method is not suitable for moving data or tenants between staging and production servers on an ongoing basis. The person performing the procedure is responsible for ensuring that data is moved correctly and does not accidentally overwrite critical data.

Option 1: Tenants are in Separate Environments and do not Share Resources

Perform the following procedure to create a new tenant.

1. On the source system, use Microsoft SQL Management Server or a similar tool to make a full Microsoft SQL backup of the Ivanti Service Manager tenant database.
2. Paste the Microsoft SQL backup onto the destination system.
3. On the destination system, log in to the Operations Console.
4. If you are going to overwrite a tenant, delete the existing tenant by doing the following:
   a. Click Tenants. The system displays the **Tenants** workspace.
   b. Click **Expanded View**.
   c. Select the tenant to delete.
   d. Select **Deactivate** to deactivate the tenant.
   e. Select **Delete** to delete the tenant. Deleting it also deletes the database.
5. Click Tenants. The system displays the **Tenants** workspace.
6. Click **Add Tenant**.
7. In the **Source DB Option** field, select From copied MSSQL backup.
8. In the **Backup Filename** field, enter the fully-qualified domain name of the source Microsoft SQL backup.
9. Enter the remaining tenant details (for example, tenant URL, DB Server, and so on), and provide a valid expiration date. See Creating a New Tenant by Using the Operations Console for information on the remaining fields.
10. Click **Create**.

The system creates the new tenant with the data from the Microsoft SQL backup. This creates or overwrites an existing tenant with new data.

Option 2: Tenants are in Separate Environments and Share Resources

You can manually reconfigure the Operations Console on the development and staging instances of a tenant to support the actual production instance and the staging/UAT instances of the tenant on separate database servers. These environments may be accessed by locally-installed Ivanti Service Manager systems as well.

For the purposes of this document, we assume that both the staging and UAT instances of the tenant are managed on a single Ivanti Service Manager system and the actual production environment is deployed on a separate system. The actual production

environment must be accessible through database queries and UNC paths from the server on which the staging and UAT instances reside.

Perform the following procedure to create a new tenant:

1. Deploy Ivanti Service Manager in both the staging/UAT and production environments.
2. In the staging/UAT environment, use the Operations Console to push the initial production instance of the tenant to the staging/UAT instances of the tenant. (See **Creating the First Staging or UAT Instance of the Tenant from the Production Instance of the Tenant** for details about this procedure.) These will become the permanent staging/UAT instances of the tenant.
3. Use the Operations Console on the staging/UAT instance of the tenant to move the metadata from each tenant instance to the next as required to manage the development lifecycle.

> Be sure to only use the Operations Console in the staging/UAT environment.
> We recommend that you do not install the Operations Console in the production environment in this scenario.

Editing a Tenant

- **Editing the Properties of a Tenant**
- **Placing a Tenant into Maintenance Mode (Deactivating a Tenant)**
- **Deleting a Tenant**
- **Clearing the Cache**

Editing the Properties of a Tenant

Follow these steps to edit an existing tenant:

> If you only want to extend the expiration date of the tenant, you can click Extend Expiration Date from the **Tenants** workspace

1. Log in to the Operations Console.
2. Click Tenants. The system displays the **Tenants** workspace.
3. Find the tenant that you want to edit and click Edit Properties.
4. Edit any or all of the following fields:
    - Company name
    - Status
    - In maintenance
    - Metadata locked
    - Expiration date
    - Named license
    - Concurrent license
    - Session timeout (in minutes)
    - Contact name
    - Enable email listener

5. Click Save.

Placing a Tenant into Maintenance Mode (Deactivating a Tenant)

To place a tenant into maintenance mode, which is also known as deactivating it, do the following:

1. Log in to the Operations Console.
2. Click Tenants. The system displays the **Tenants** workspace.
3. Find the tenant that you want to place into maintenance mode (deactivate) and do one of the following:
    - Click Deactivate.
    - Click Edit Properties, check In Maintenance, and click Save.

Deleting a Tenant

Before you can delete a tenant, you need to move it into maintenance mode. See **Placing a Tenant into Maintenance Mode (Deactivating a Tenant)**.

1. Log in to the Operations Console.
2. Click Tenants. The system displays the **Tenants** workspace.
3. Find the deactivated tenant that you want to delete and click Delete.
4. Click OK in the confirmation dialog box.

Clearing the Cache

If the Ivanti Service Manager application server starts returning incorrect data, you may need to clear the cache. You can click Clear Cache to clear the metadata cache on the Ivanti Service Manager application server without having to shut down the Ivanti Service Manager application server. This returns the Ivanti Service Manager application server to the right state and synchronizes it with Ivanti Service Manager.

1. Log in to the Operations Console.
2. Click Tenants. The system displays the **Tenants** workspace.
3. Click Clear Cache.

8.6. Appendix A: Using the Operations Console to Perform Administrative Tasks

- Working With Operations Console Users
- Viewing System Information
- Viewing Events and Audit History
- Reviewing License Usage
- Configuring Landscapes
- Working with Tenants and Tenant Instances

Working With Operations Console Users

- About Operations Console Users
- About the Users Workspace
- Adding a New Operations Console User
- Editing a Operations Console User
- Deleting a Operations Console User
- Changing Your User Display Name
- Changing Your Password

About Operations Console Users

Use the **Users** workspace to add and edit Operations Console users. These are users who can log in to and use the Operations Console; these are not tenant users. Your environment may have just one admin login ID, which is used by multiple users, or it may have multiple login IDs with one for each user.

The administrator users here in the Operations Console are not the same as the administrator users in Ivanti Service Manager.

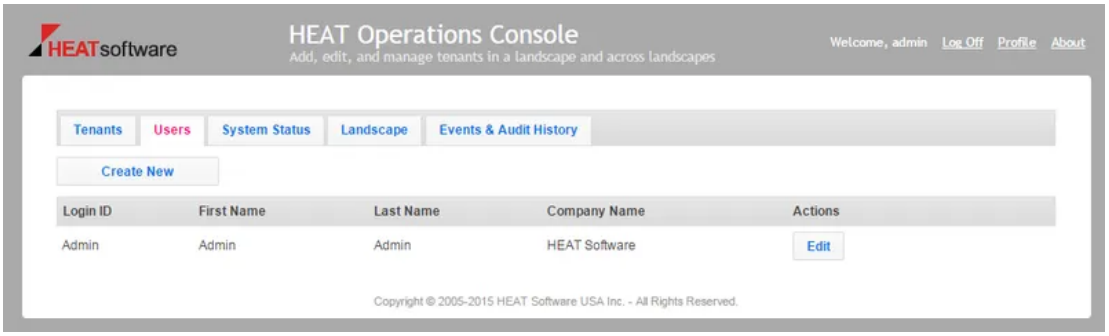
Users who are administrators can delete non-administrator users, and have access to all areas of the Operations Console. Users who are not administrators have a limited set of abilities.

Function	Administrators	Non-Administrators
View the **Users** tab	Y	Y
Create a user	Y	N
Edit their own user account	Y (Can delete their own administrator status)	Y (Cannot make themselves an administrator)
Edit other users	Y (But cannot delete administrator status from the user call Admin)	N
Delete their own user account	N	N
Delete other users who are not administrators	Y (But cannot delete the user called Admin)	N
Delete other users who are administrators	N	N
View the **Tenants** tab	Y	Y
View all tenants and tenant instances in the system	Y	N (Can only view tenants and tenant instances associated with their company, unless they leave the company name blank in which case they can see all tenants and tenant instances)
View the **Landscape** tab	Y	N
View the **System Status** tab	Y	N
View the **Events & Audit History** tab	Y	Y

About the Users Workspace

Use the **Users** workspace to view and edit information about the users who can access the Operations Console. See **Users Tab When Logged In as an Administrator**.

Users Tab When Logged In as an Administrator



From this workspace you can:

- View the login ID, first name, last name, and company name of each user.
- Create a new user (if you have permission).
- Edit the information for a user (if you have permission).
- Delete a user (if you have permission).

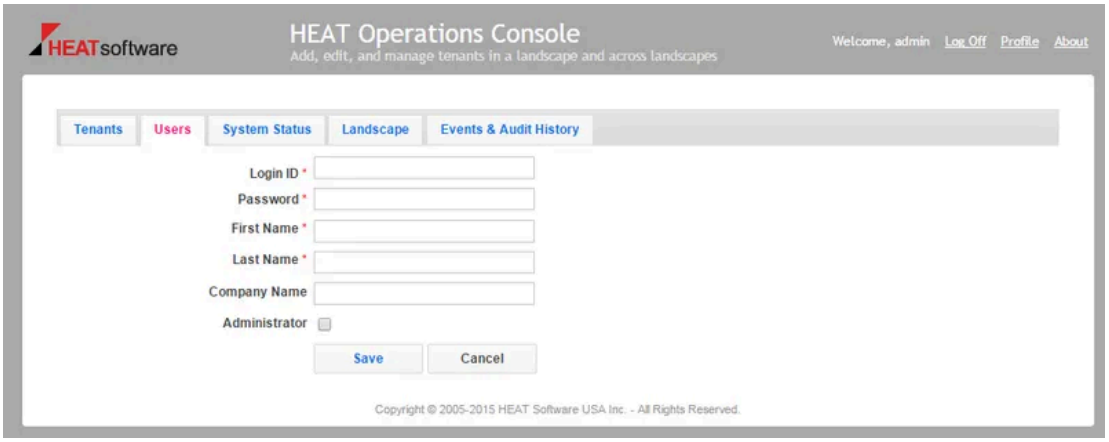
Adding a New Operations Console User

Only administrators can add new users.

To add new users who can access the Operations Console, do the following:

1. Log in to the Operations Console.
2. Click the **Users** tab.
3. Click Create New.
4. Enter data in the fields. See **Adding a New User Page**.

Adding a New User Page



5. Click Save.

Editing a Operations Console User

If you are not an administrator, you can only edit yourself, but you cannot make yourself an administrator.
If you are an administrator, you can edit yourself or other users. However, you cannot edit other administrator users to make them non-administrators. You can, however, edit yourself to make yourself a non-administrator, but once you do so, you cannot change yourself back to an administrator.

1. Log in to the Operations Console.
2. Click the **Users** tab.
3. Click Edit for the user to edit.
4. Update the fields as needed.
5. Click Save.

Deleting a Operations Console User

Only administrators can delete users, but they cannot delete themselves.

1. Log in to the Operations Console.
2. Click the **Users** tab.
3. Click Delete for the user to delete.
4. Click Yes at the verification prompt.

Changing Your User Display Name

You can change the name that is displayed for first and last name.

1. Log in to the Operations Console.
2. Click Profile in the upper right corner.
3. Update your first or last name, or both.
4. Click Save.

If you need to change the user display names of other users, follow the process described in **Editing a Operations Console User**.

Changing Your Password

To change your password, do the following:

1. Log in to the Operations Console.
2. Click Profile in the upper right corner.
3. Click Change password.
4. Enter the old password, the new password, and then reenter the new password.
5. Click Save.

If you need to change the passwords of other users, for example if a user forgets his password, follow the process described in **Editing a Operations Console User**.

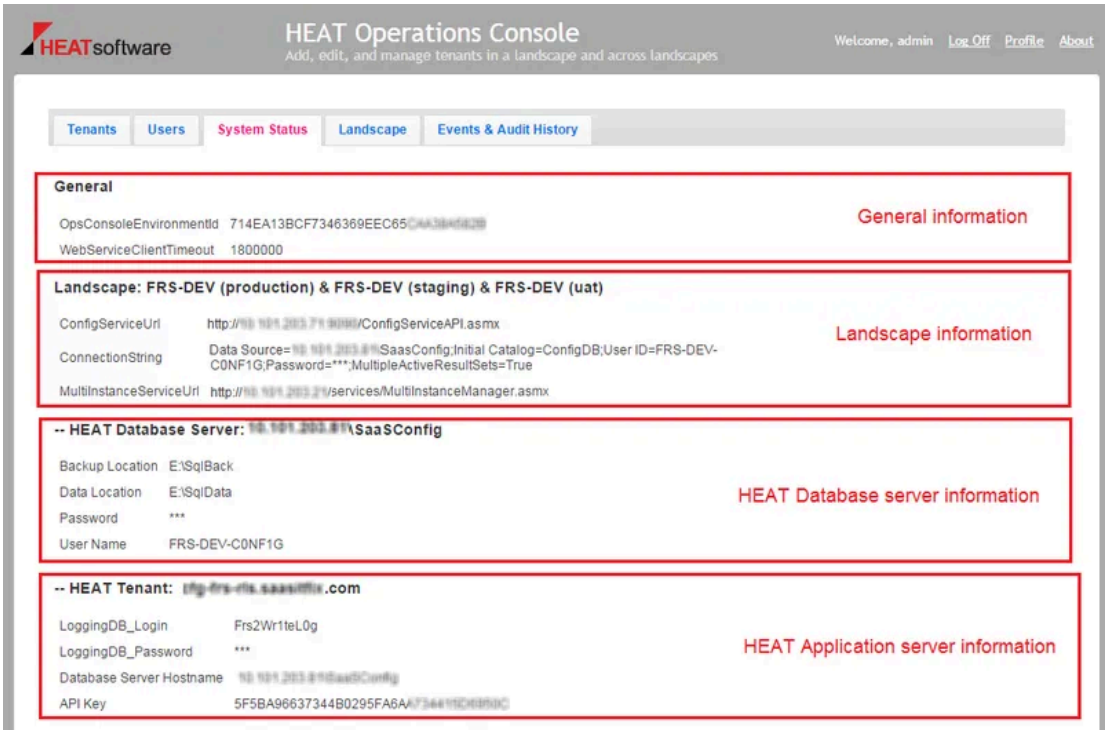
Viewing System Information

- **About System Information**
- **Viewing Additional Information**
- **Fixing Problems**

About System Information

Use the **System Status** workspace to view status information about the Ivanti Service Manager environments. Information that you can view includes general information such as the Operations Console environment ID and the timeout value used throughout the environment. See **System Status Tab**.

System Status Tab



For each environment, you can also see the following landscape information:

- The URL for the Ivanti Service Manager configuration server
- The URL for the multi-instance service
- The connection string settings, which describes where the Operations Console resides. Usually, you use the same configuration database as the Operations Console database; however, it can be an independent database anywhere outside of Ivanti Service Manager. You configure this connection string information when you initially install Ivanti Service Manager using the System Configuration Wizard.

For each Ivanti Service Manager database server, you can view:

- IP address
- Backup location
- Data location
- Password
- Administrator user name

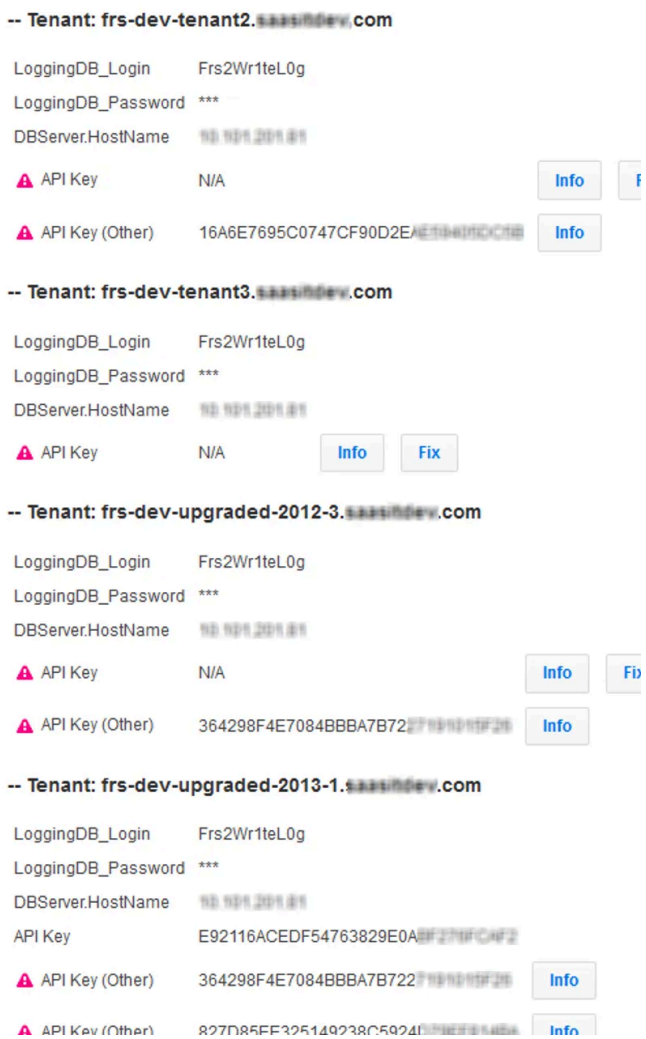
For each Ivanti Service Manager application server, you can view:

- The login ID and password for the database that logs events.
- The Ivanti Service Manager database server host name.
- The API key. There can be multiple API keys. The one without any modifiers, such as **API Key (other)**, is the Operations Console API key and is most important. The API key must be unique and cannot be shared between tenant instances. The system automatically generates the API key if you add a tenant instance using the Operations Console, but the field is blank if you add the tenant instance manually.

Viewing Additional Information

When you click System Status and view the system information, there may be errors. See **System Status Problems**. To see information about the error, click Info.

System Status Problems



Fixing Problems

When you view the system information, there may be errors. See **System Status Problems**. If the system can fix the error, it displays the word Fix next to the item. To fix the error, click Fix.

One problem that must be fixed is having duplicate API keys. If a tenant instance has a duplicate API key, the system cannot communicate with the tenant instance. To fix this problem, click Fix. If the API key is blank, you can generate an API key by clicking Fix.

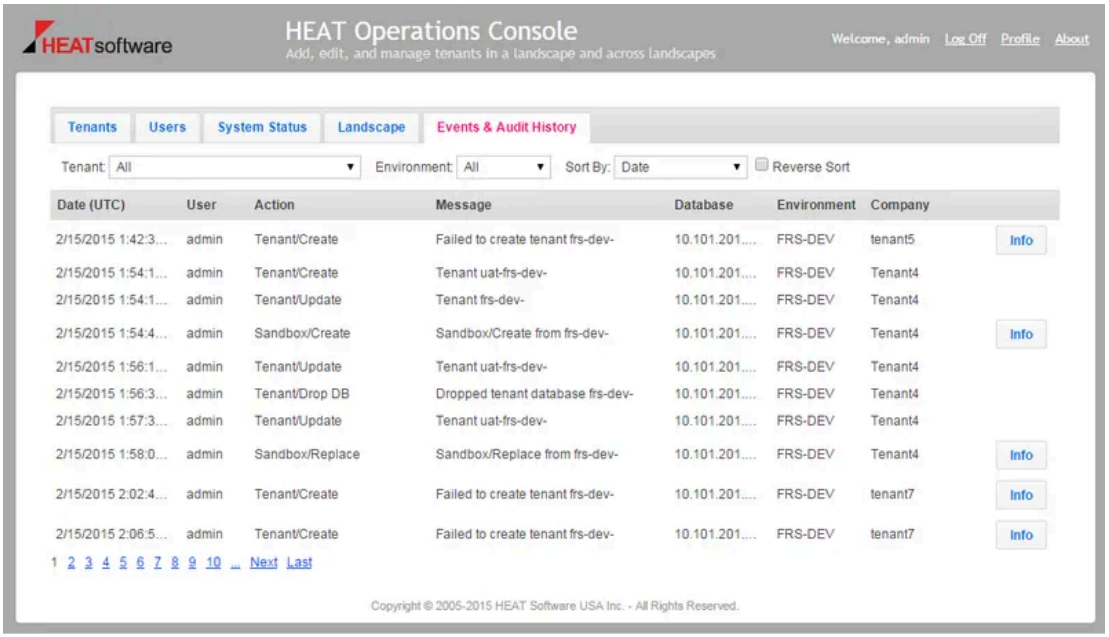
Viewing Events and Audit History

- **About Events and Audit History**
- **Manipulating Events and Audit History Data**
- **Viewing Additional Information about Events and Audits**

About Events and Audit History

Use the **Events & Audit History** workspace to view information about each action that has occurred on a tenant or a tenant instance. You can use this information to help with troubleshooting problems. It lists each action that has occurred in chronological order and displays additional information for errors. See **Events and Audit History**.

Events and Audit History



The page lists the following information:

- The action date, in UTC time.
- Who performed the action.
- What the action was.
- The message that displayed when the action took place.
- The database on which the action was performed.
- The environment in which the action was performed.
- The company to which the tenant or tenant instance belongs.

Manipulating Events and Audit History Data

Use the controls to do the following:

- Filter the results by tenant, tenant instance, or environment.
- Sort the results by date, tenant, tenant instance, or company name.
- Sort the results in reverse order.
- Move from page to page of data.

Viewing Additional Information about Events and Audits

Some items may have additional information. To see this information, if available, click Info next to the message.

Reviewing License Usage

You can use the Operations Console to review the license usage for all tenants and tenant instances by doing the following:

1. Log in to the Operations Console.
2. Click Tenants.
3. Click Download License Usage.

The system downloads a Microsoft Excel file with the license information for all tenants and tenant instances. The file lists the date the license was used, the number of licenses used, and the types of licenses used.

Appendix B: Migrating Data from One Tenant 8.7. Instance to Another Using Simple Mode or Advanced Mode

> We do not recommend using these methods for migrating data. See **The Four Ways to Migrate Data** for a list of the ways to migrate data.
> We recommend enabling and using the Ivanti Service Manager development project for migrating data, as described in **Migrating Data using Heat Development Project**.

- About Using Simple Mode and Advanced Mode
- How to Use Simple Mode to Migrate Data
- Using Advanced Mode to Migrate Data
- Migration Scenarios Using Simple Mode and Advanced Mode

About Using Simple Mode and Advanced Mode

The relationship data between two business objects is copied from the source to the target under the following operations and conditions:

Operations:

- Simple mode copy
- Copy configuration and master data
- Copy master data only

Condition:

- Only both business objects on both sides of the relationship are included for copying at the same time.

For example, if you copy user-related data, but do not copy the users, then data such as user to team and user to group assignment is not copied. As another example, if you want the service to category assignment to be copied, you must copy both the service (CI#Service) and the category (Category#) in the same push operation.

Choose between the following:

- Simple Mode: See **How to Use Simple Mode to Migrate Data**.
- Advanced Mode: See **Using Advanced Mode to Migrate Data**.

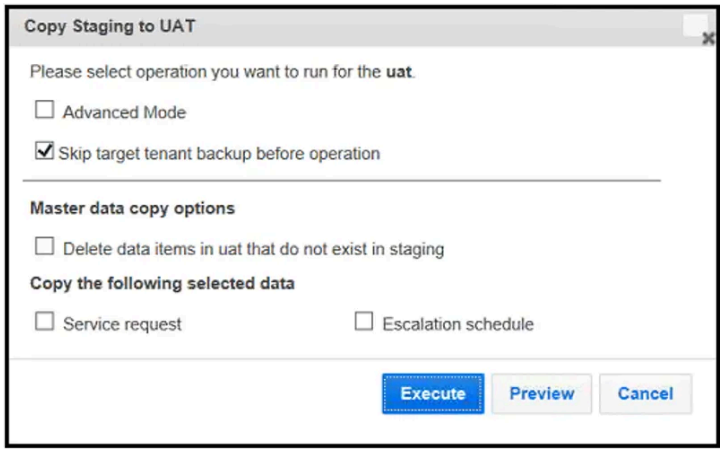
How to Use Simple Mode to Migrate Data

- Using Simple Mode to Migrate Data
- Simple Mode Migrated Data

Using Simple Mode to Migrate Data

When you start the migration, you see the **Copy** dialog box in **Operations Console with Simple Mode and Advanced Mode**.

Operations Console with Simple Mode and Advanced Mode



1. Do not check Advanced Mode, which means you are choosing to use Simple Mode, which migrates a predefined set of data.
2. Check Skip target tenant backup before operation to not perform a database backup. Leave the box empty if you want to back up the database before proceeding with the migration.
3. Check Delete data items in *target_tenant_instance* that do not exist in *source_tenant_instance* to delete the data items in the target tenant instance that do not exist in the source tenant instance.
4. Under **Copy the following selected data**, check Service request if you want to migrate service request information, and check Escalation schedule if you want to migrate escalation schedule information.
5. Do one of the following:
    - Click Preview to preview the changes that will be migrated. The system generates an XML file for you to review. You must save the XML file and open it with an XML editor to see the changes. If you need to make additional changes, go back into the system and make them. If you are satisfied with the changes, go back to the Operations Console and click Execute.
    - Click Execute to start the data migration.

Simple Mode Migrated Data

- Metadata
- Custom Validation Data
- Master Data
- Settings

Metadata

> See Types of Data for information about metadata.

- Deletes the metadata items in the target tenant instance that do not exist in the source tenant instance for the following items (equivalent to checking the box next to the metadata item name in the **Delete metadata items in *target_tenant_instance* that do not exist in *source_tenant_instance*** section):
    - Role
    - Layout
    - Form
    - Grid

    - Trigger
    - Rule
    - Rule override
    - Validation data
    - Validation list definition
- Does not delete the metadata items in the target tenant instance that do not exist in the source tenant instance for the following items (equivalent to not checking the box next to the metadata item name in the **Delete metadata items in *target_tenant_instance* that do not exist in *source_tenant_instance*** section):
    - Business object
    - Branding
    - Workflow
    - Counter
    - Grid-part metadata
    - Chart-part metadata
    - Special-part metadata
    - Search
    - Report
    - Dashboard
    - Quick action
    - View part

Does not replace the metadata items in the target tenant instance that are newer than those in the source tenant instance for the following items (equivalent to not checking the box next to the metadata item name in the **Replace metadata items in *target_tenant_instance* that are newer than those in *source_tenant_instance*** section):

- Search
- Report
- Dashboard
- Quick action

See Simple Mode Options -- Top Half.

Custom Validation Data

> See Types of Data for information about validation data.

Deletes all custom validation data objects in the target tenant instance that do not exist in the source tenant instance. This is equivalent to checking **Delete data items in *target_tenant_instance* that do not exist in *source_tenant_instance*** in the **Custom defined validation list data objects** section.

See Simple Mode Options -- Top Half.

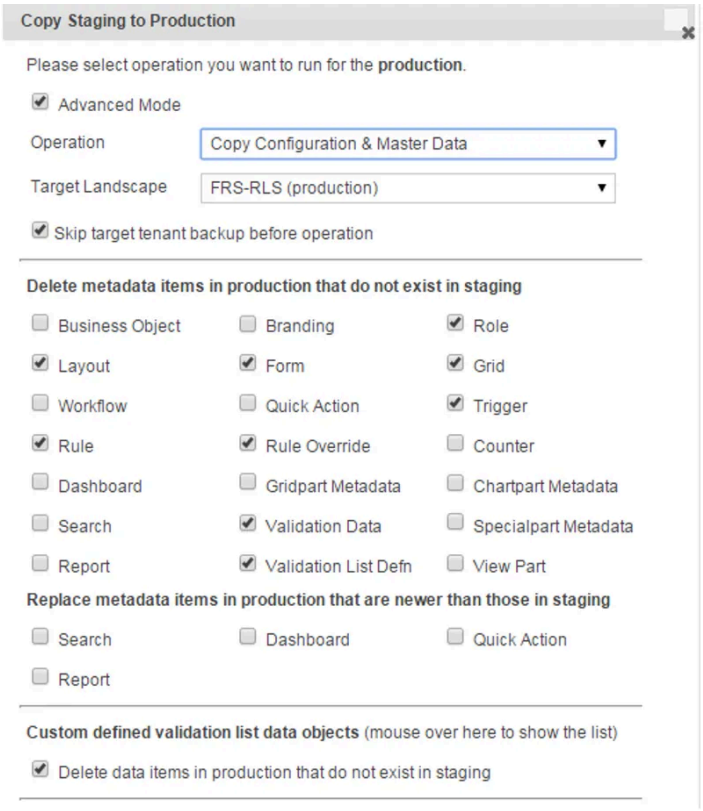
Master Data

> See **Types of Data** in the **Terminology** section for information about master data.

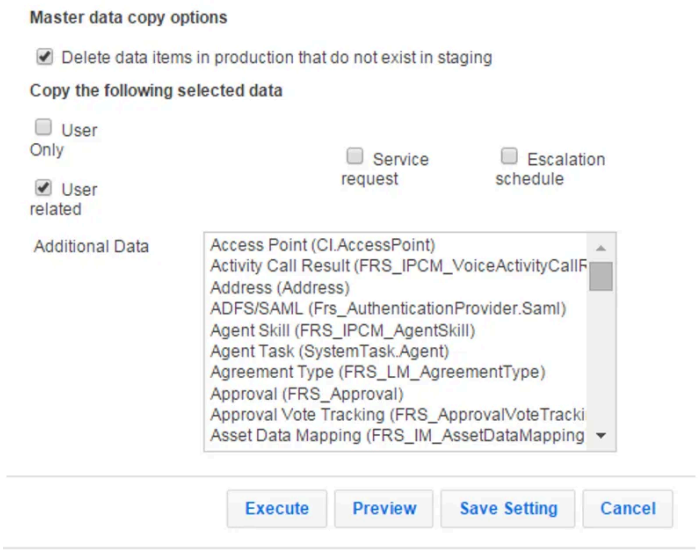
- Migrates user-related groups of master data. This is equivalent to checking Delete data items in *target_tenant_instance* that do not exist in *source_tenant_instance* in the **Master data copy options** section.
- Migrates user-related data (including employee, external contact, profile base, and profile employee information), which is the equivalent of checking User related in the **Copy the following selected data** section.

See **Simple Mode Options -- Top Half** and **Simple Mode Options -- Bottom Half**.

Simple Mode Options -- Top Half



Simple Mode Options -- Bottom Half



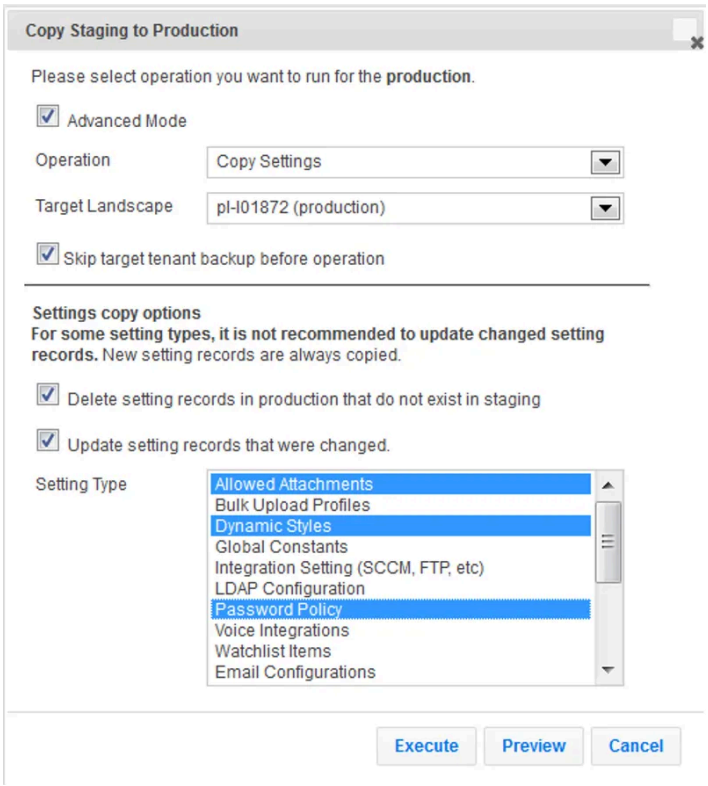
Settings

> See **Types of Data** for information about settings, also referred to as configuration data.

- Deletes settings in the target tenant instance that do not exist in the source tenant instance. This is equivalent to checking Delete setting records in *target_tenant_instance* that do not exist in *source_tenant_instance* in the **Settings copy options** section.
- Updates settings that were changed. This is equivalent to checking Update setting records that were changed in the **Settings copy options** section.
- Applies the two options above to the following settings:
  - Allowed attachments
  - Dynamic styles
  - Password policy
  - Approval status keywords
  - Email status update mapping settings
  - Ignore message with subject settings
  - Object mapping for email
  - Task assignment status keywords
  - Truncate body settings
  - Trusted sender domains

See **Simple Mode -- Settings Options**.

Simple Mode -- Settings Options



Using Advanced Mode to Migrate Data

- Migrating Metadata and Validation Data (Copy Configuration)
- Updating Metadata, Validation Data, and Master Data (Copy Configuration & Master Data)
- Updating Master Data (Copy Master Data Only)
- Overwriting all Metadata and Validation Data (Copy to Replace)
- Updating Only Transactional Data (Copy Data Only)
- Updating Configuration Data (Copy Settings)
- Updating Configuration Data (Copy Translation)
- Increasing the Version Number (No Op [Increase Version Only])

Migrating Metadata and Validation Data (Copy Configuration)

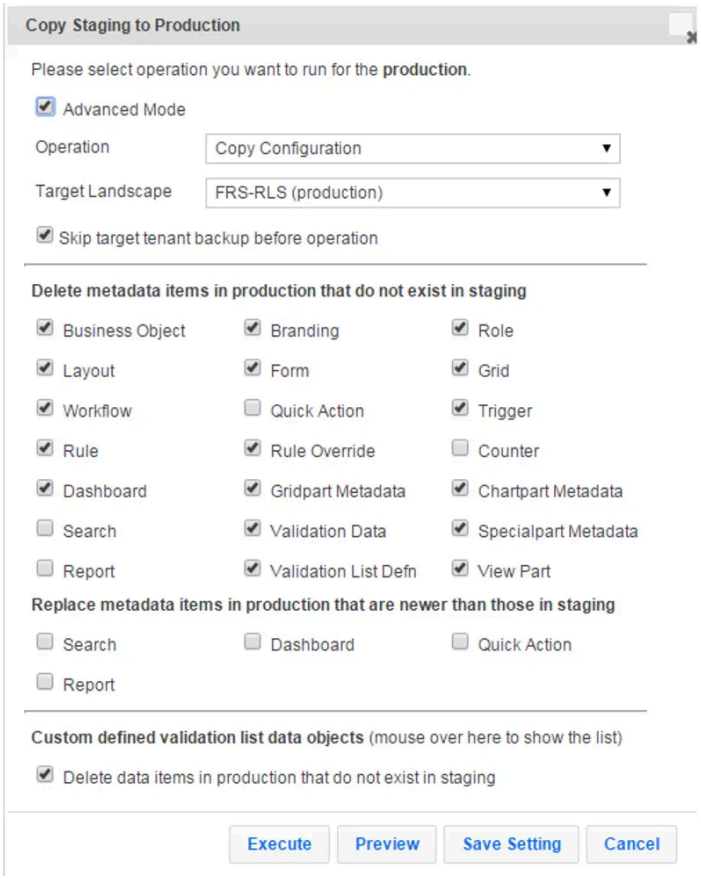
Perform the following steps to update the configuration of a tenant instance by migrating metadata and validation data. (See Types of Data for information about metadata and validation data.) When you perform this migration, metadata and validation data that exist on the target tenant instance are overwritten by the data on the source tenant instance.

> Perform steps 1 to 4 in Starting the Tenant Instance Migration before you perform the steps in this section.

1. Check Advanced Mode.
2. In the **Copy** dialog box, for the **Operation** field, choose **Copy Configuration** to configure the Operations Console to copy metadata and validation data. The dialog box expands and appears similar to the dialog box in Copy Dialog Box (Migrate Metadata and Validation Data).

> You cannot change the value for the **Target Landscape** field, as it is read-only and depends on the target you chose before you started the migration

Copy Dialog Box (Migrate Metadata and Validation Data)



3. Check **Skip target tenant backup before operation** to select whether to back up the target tenant instance prior to the data migration.
    - To back up the target tenant instance, leave this checkbox empty. If you have not yet backed up the target tenant outside of the Operations Console, we highly recommend that you back it up here.
    - To not back up the target tenant instance, select this checkbox. If you have already backed up the target tenant outside of the Operations Console, it is generally not necessary to back it up here.
4. Use the **Delete metadata items in *target_tenant_instance* that do not exist in *source_tenant_instance*** section to select how to process items that were deleted in the source tenant instance prior to the migration.

For example, assume that staging is the source tenant instance and UAT is the target tenant instance. If you deleted some business objects on the staging tenant instance as part of a reconfiguration procedure, you can select whether to delete or retain those business objects on the UAT tenant instance when you perform the data migration.

- To delete business objects on the UAT tenant instance, if those business objects were also deleted on the staging tenant instance, check **Business Objects**.
- To retain business objects on the UAT tenant instance even though they were deleted on the staging tenant instance (that is, to *merge* the staging and UAT business object lists), do not check **Business Objects**.
- Use the checkboxes for other items (layouts, workflows, rules, and so on) in the same way.

If you do not select any options in this section, all data is merged, and no data on the UAT instance of the tenant is deleted.

> Even if the target tenant instance is locked, you can still create, edit, and delete dashboards, quick actions, reports, and searches. If you select any of these items, be aware that they could be deleted or updated when you execute a migration even if the target tenant instance is locked

5. Use the **Replace metadata items in *target_tenant_instance* that are newer than those in *source_tenant_instance*** section to select how to process items that exist on both the source tenant instance and the target tenant instance when the version on the target tenant instance is newer than the version on the source tenant instance. This situation is not typically encountered or recommended, but the ability to respond to it is provided because it does occur occasionally.

For example, assume that staging is the source tenant instance and UAT is the target tenant instance. During testing on the UAT tenant instance, it was determined that a saved search needed to be modified, and the modification was performed on the UAT instance of the tenant. At the same time, another item (for example, change management processes and approvals) was updated on the staging instance of the tenant as part of a scheduled reconfiguration.

   - To *merge* the data on the staging instance of the tenant with the data on the UAT instance of the tenant, check **Search**. This results in the most recent change updates (from staging) and the most recent saved search updates (from UAT) coexisting on the UAT instance of the tenant after the migration. This is the recommended procedure because it ensures that the most recent version of all data is present on the UAT instance of the tenant after the migration.
   - To *overwrite* all of the data on the UAT instance of the tenant with data from the staging instance of the tenant (including overwriting the saved searches that were updated on the UAT instance of the tenant), do not check **Search**. This results in the most recent change updates (from staging) migrating to the UAT instance of the tenant, but the saved search changes that were made on the UAT instance of the tenant prior to the migration are overwritten and reverted to the version on the staging instance of the tenant.
   - Use the checkboxes for other items (dashboards, quick actions, and reports) in the same way.

If you do not select any checkboxes in this section, all data on the UAT instance of the tenant is *overwritten* by data from the staging instance of the tenant. To ensure that all data on the staging instance of the tenant is *merged* with the data on the UAT instance of the tenant (instead of being overwritten), select every checkbox in this section.

> Even if the target tenant instance is locked, you can still create, edit, and delete dashboards, quick actions, reports, and searches. If you select any of these items, be aware that they could be deleted or updated when you execute a migration even if the target tenant instance is locked.

6. Use the **Custom defined validation list data objects** section to select how to process validation lists that have been customized. *Customized* validation lists are not included in the **Validation Data** and **Validation List Definition** checkboxes shown in Copy Dialog Box (Migrate Metadata and Validation Data) and described in step 3. To migrate customized validation lists, you must specify them in this step instead. (See Types of Data in the Terminology section for information about validation data.)

For example, assume that staging is the source tenant instance and UAT is the target tenant instance. If a customized validation list already exists on the UAT tenant instance, and the validation list is then further customized on the staging tenant instance, you can select whether to *overwrite* the UAT version with the staging version or *merge* the two validation lists.

   - To see the customized validation lists, place the mouse pointer over the checkbox.
   - To delete customized validation list items on the UAT instance of the tenant that were also deleted on the staging instance of the tenant (that is, to *overwrite* the UAT validation list with the staging validation list), check **Delete data items in UAT**

that do not exist in Staging.

- To retain customized validation list items on the UAT instance of the tenant even though they were deleted on the staging instance of the tenant (that is, to *merge* the staging and UAT validation lists), leave the **Delete data items in UAT that do not exist in Staging** checkbox empty.

7. Select one of these options to start or postpone the data migration:
   - **Execute**: Starts the migration immediately.
   - **Preview**: Creates an XML-based patch file called **MetadataDiff.xmlMetadataDiff.MetadataPatch**. You can review this file to see details of the data that will be migrated. A preview takes the same amount of time to execute as an actual migration.
   - **Save Setting**: Saves your current selections without executing a migration. The next time you open this dialog box it is populated with your saved settings.
   - **Save UAT Settings**: Saves your current selections for use when you migrate from the UAT instance of the tenant to the production instance of the tenant.
   - **Cancel**: Exits the dialog box without saving any settings.

When the data migration finishes, the system creates a log entry for it and updates the status of the affected tenants and tenant instances in the Operations Console migration dashboard. The system also displays an informational message with the results of the migration. Check the **Events & Audit History** workspace to review the warnings and errors before continuing. (See **About the Events & Audit History Workspace**.)

Updating Metadata, Validation Data, and Master Data (Copy Configuration & Master Data)

> Do not use this feature if you have more than 100,000 records as the migration will fail. If you have more data, we recommend that you either exclude some of the records from the data migration or use Microsoft SQL Server scripts or some other tool to migrate the data.

From the Operations Console you can migrate master data together with metadata and validation data in the same migration. (See **Types of Data** in the **Terminology** section for information about master data, metadata, and validation data.)

Master data consists of data sets that are used across an organization and are typically unique to that organization. User and role definitions, role assignments, and categories are examples of master data.

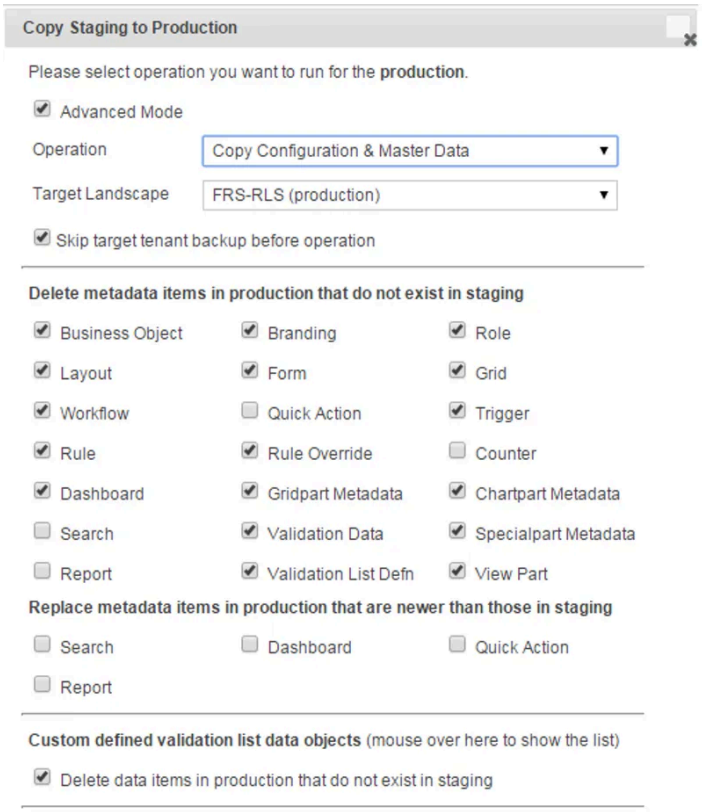
Perform the following steps to update the configuration for a tenant instance by migrating master data together with metadata and validation data. When you perform this migration, master data, metadata, and validation data that exist on the target tenant instance (UAT in this example) are overwritten by the data on the source tenant instance (staging in this example).

> Perform steps 1 to 4 in **Starting the Tenant Instance Migration** before you perform the steps in this section.

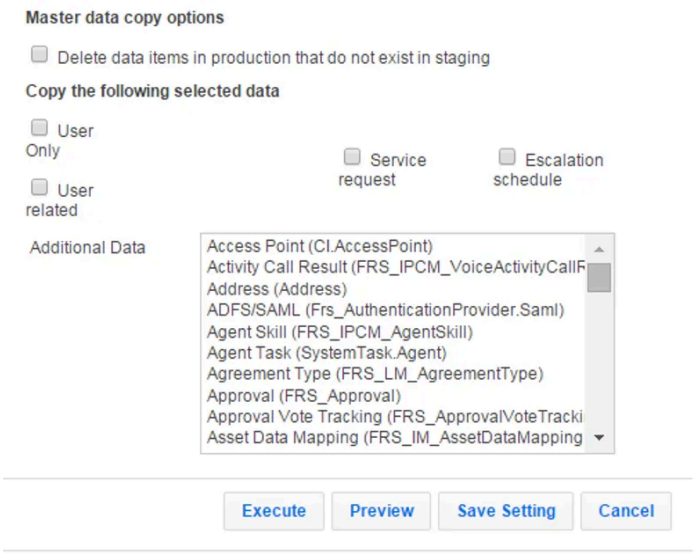
1. Check Advanced Mode.
2. In the **Operation** field in the **Copy** dialog box, choose **Copy Configuration & Master Data** to configure the Operations Console to copy master data together with metadata and validation data. The dialog box expands and appears similar to the dialog box shown in **Copy Dialog Box (Migrate Metadata, Validation Data, and Master Data) -- Top Half** and **Copy Dialog Box (Migrate Metadata, Validation Data, and Master Data) -- Bottom Half** .

Note that the top portion of the dialog box is identical to the dialog box shown in **Copy Dialog Box (Migrate Metadata and Validation Data)**. The bottom portion of the dialog box shown in **Copy Dialog Box (Migrate Metadata, Validation Data, and Master Data) -- Bottom Half** (beginning with **Master data copy options**) is new.

Copy Dialog Box (Migrate Metadata, Validation Data, and Master Data) -- Top Half



Copy Dialog Box (Migrate Metadata, Validation Data, and Master Data) -- Bottom Half



3. Fill in the top portion of the dialog box (the settings for metadata and validation data) as described in **Migrating Metadata and Validation Data (Copy Configuration)**.
4. Use the **Master data copy options** section to select how to process master data that was deleted in the source tenant instance prior to the migration.

For example, assume that staging is the source tenant instance and UAT is the target tenant instance. If you deleted master data on the staging tenant instance as part of a reconfiguration procedure, you can select whether to delete or retain that master data on the UAT tenant instance when you perform the migration.

- To delete master data on the UAT tenant instance that was also deleted on the staging tenant instance (that is, to *overwrite* the UAT master data with the staging master data), check **Delete data items in UAT that do not exist in Staging**. Keep in mind that if deleted master data record definitions are used elsewhere in Ivanti Service Manager, the deletion could break existing relationships or otherwise adversely affect the content of data records.
- To retain master data on the UAT tenant instance even though it was deleted on the staging tenant instance (that is, to *merge* the staging and UAT master data), do not check **Delete data items in UAT that do not exist in Staging**.

  Because of the implications of possibly breaking relationships and data records by deleting master data records in this step, it is sometimes advisable to fill in the **Copy the following selected data** section as described in steps 4 and 5 before making your choice in this step.

5. Use the **Copy the following selected data** section to select which master data to migrate. If you do not select any items in this section, no master data is migrated.

   Master data selection choices are as follows:

   - User Only: Migrates user-related data (including employee, external contact, profile base, and profile employee information).
   - **User related**: Place the mouse pointer over this item to display an expanded list of all business objects contained in this item. Includes contact group, department, email address, organizational unit, and standard user team data). When you select this item, the Operations Console performs a full copy of all master data records in the tenant instance from each of the listed business objects.

> If a tenant instance has more than 20,000 users in the system, we do not recommend selecting **User Related** due to performance impact.

  - The profile base object is included in this selection, meaning that Profile.Employee and Profile.ExternalContact are migrated. If the configuration includes a newly created child of the group-object Profile# (this is the profile base object), you must include the master data of the associated object as part of the additional data to be copied (described in step 5).
  - The organizational unit definition is also included in this item. If any changes were made to organizational units on other tenants through the Ivanti Service Manager Configuration Console or by other means, they are overwritten when this item is migrated.

> The **Address.Email** object is also included in this item. If the email addresses for the staging instance of the tenant are "scrubbed" to use the @saasitdemo.com domain instead of the company domain, the scrubbed email addresses for all Profile.object records are migrated to the target domain. If this configuration is eventually migrated to the production instance of the tenant, the @saasitdemo.com domain setting is still in place, and various login, LDAP import, notification, and email delivery issues will occur. A workaround for this scenario is to not scrub email addresses on the staging instance of the tenant, and instead block all outbound email or redirect outbound email to a single email address. The options to block email and to re-direct all mail to a single account can be set in the email configuration business object

  - Because user and team email addresses on the staging and UAT instances of the tenant might be scrubbed, the migration does not update existing users or teams. Instead, it migrates only new users and user teams.

    - User role assignments and user team assignments are updated as part of this selection.
  - **Service request**: Place the mouse pointer over this item to display an expanded list of all business objects contained in this item. When you select this item, the Operations Console performs a full copy (not a partial or selective copy) of all master data records in the tenant instance from each of the listed business objects.

    To migrate a subset of the master data records shown in the list (for example, to migrate only the service request category (which has the object name "category")), do not check **Service Request**. Instead, include category in the **Additional Data** section as described in step 5. When you migrate data from the **Additional Data** section, it is a full migration (not a partial or selective migration).

    Request offerings and their accompanying workflows are migrated when you select this item. Only those workflows that are part of a request offering design are migrated. Workflows that are associated with the service request business object are not migrated.

  - **Escalation schedule**: Place the mouse pointer over this item to display an expanded list of all business objects contained in this item, such as escalation schedule, escalation setting, service level package, and service level target. When you select this item, the Operations Console performs a full copy of all master data records in the tenant instance from each of the listed business objects.

    As a general rule, all of the escalation parameters displayed in the Ivanti Service Manager Configuration Console **Escalation** tab are migrated.

    Note that when you migrate escalation schedules, existing schedules on the target tenant instance are overwritten.

6. Use the **Additional Data** section to list additional master data or transactional data to migrate. Select one or more items to migrate. The **Additional Data** section contains business objects that:
   - Are not a part of metadata.
   - Are not a part of custom defined validation data.
   - Are not a part of the **User Only**, **User related**, **Service request**, or **Escalation schedule** categories described in step 5.
   - Are not a part of typical transaction data such as incident, task, and change.
   - Might be transactional data. The Operations Console cannot autonomously differentiate between master data and transactional data business objects.

Guidelines for selecting items are as follows:

- You can select zero or more items from the **Additional Data** list.
- Do not select a business object that contains more than 10,000 records due to performance concerns.
- If an object that is usually configured as metadata was for some reason configured as master data (or results in the creation of a related master data object), select the item here so that it is migrated.

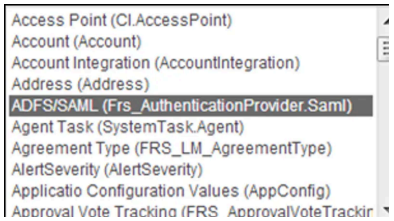
  For example, validation list data is migrated only if the validation list (that is, the pick list) is based on a business object that is part of the business object list for the validation business object. If the validation list (that is, the pick list) is based on a master business object instead of a validation business object, it will not be migrated by checking Validation Data. Instead, you must select the ValidationListData item in the **Additional Data** list (as described here in step 5) in order for the validation list to be migrated.

  Click or <ctrl>-click items in the **Additional Data** list to migrate custom data configured as business objects. Keep in mind that migrating business objects with thousands of records can impact performance; these business objects should usually

be migrated outside of the Operations Console. Smaller objects (such as Category) are better candidates to select here for migrating within the Operations Console.

Note that the Operations Console cannot perform selective copies during a migration. That is, it can only copy a table to a table. For example, if you select the **Additional Data** item **Table ADFS/SAML (frs_AuthenticationProvider.SAML)** (Example Item Selected in the Additional Data List), *all* of the records from the source database are copied to the target database where frs_AuthenticationProvider.AuthType = SAML. There is no option to copy only "active" or some selective option from the table frs_AuthenticationProvider.SAML.

Example Item Selected in the Additional Data List



7. Select one of these options to start or postpone the data migration:
   - **Execute**: Starts the migration immediately.
   - **Preview**: Creates an XML-based patch file for review.
   - **Save Setting**: Saves your current selections without executing a migration. You can continue the migration process later using your saved settings.
   - **Save *Target* Settings**: Saves your current selections for use with the next migration in the migration sequence. For example, if the current migration is from staging to UAT, choose this option to save the settings for the UAT to production migration that you will perform next.
   - **Cancel**: Exits the dialog box without saving any settings.

When the data migration finishes, the system creates a log entry for it and updates the status of the affected tenants and tenant instances in the Operations Console migration dashboard as shown in Migration Dashboard in Starting the Data Migration. The system also displays an informational message with the results of the migration. Check the **Events & Audit History** workspace to review the warnings and errors before continuing.

Updating Master Data (Copy Master Data Only)

From the Operations Console you can migrate master data only. Only do this after you have migrated the configuration data at least once.

> Do not use this feature if you have more than 100,000 records as the migration will fail. If you have more data, we recommend that you either exclude some of the records from the data migration or use Microsoft SQL Server scripts or some other tool to migrate the data.

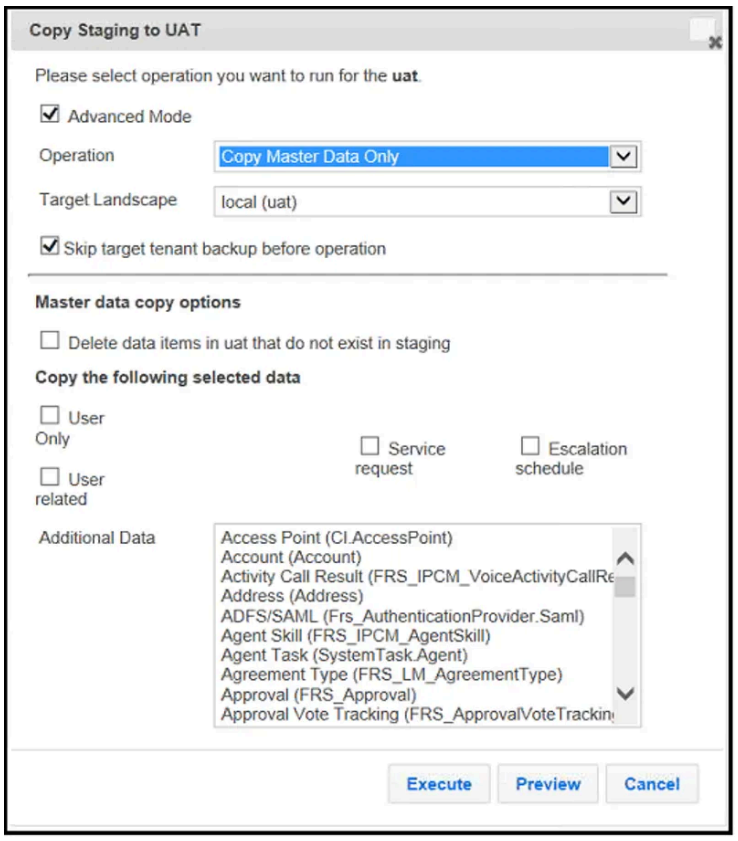
Master data consists of data sets that are used across an organization and are typically unique to that organization. User and role definitions, role assignments, and categories are examples of master data. (See Types of Data for information about master data.)

Perform the following steps to update the configuration of a tenant instance by migrating master data only. When you perform this migration, master data that exists on the target tenant instance (UAT in this example) is overwritten by the data on the source tenant instance (staging in this example).

Perform steps 1 to 4 in **Starting the Tenant Instance Migration** before you perform the steps in this section.

1. Check Advanced Mode.
2. In the **Operation** field in the **Copy** dialog box, choose **Copy Master Data Only** to configure the Operations Console to copy master data only. The dialog box expands and appears similar to the dialog box shown **Copy Dialog Box (Migrate Master Data Only)**.

Copy Dialog Box (Migrate Master Data Only)



3. Fill in the dialog box (master data copy options) as described in steps 4, 5, 6, and 7 of **Updating Metadata, Validation Data, and Master Data (Copy Configuration & Master Data)**.

When the data migration finishes, the system creates a log entry for it and updates the status of the affected tenants and tenant instances in the Operations Console migration dashboard as shown in **Migration Dashboard** in **Starting the Data Migration**. The system also displays an informational message with the results of the migration. Check the **Events & Audit History** workspace to review the warnings and errors before continuing.

Overwriting all Metadata and Validation Data (Copy to Replace)

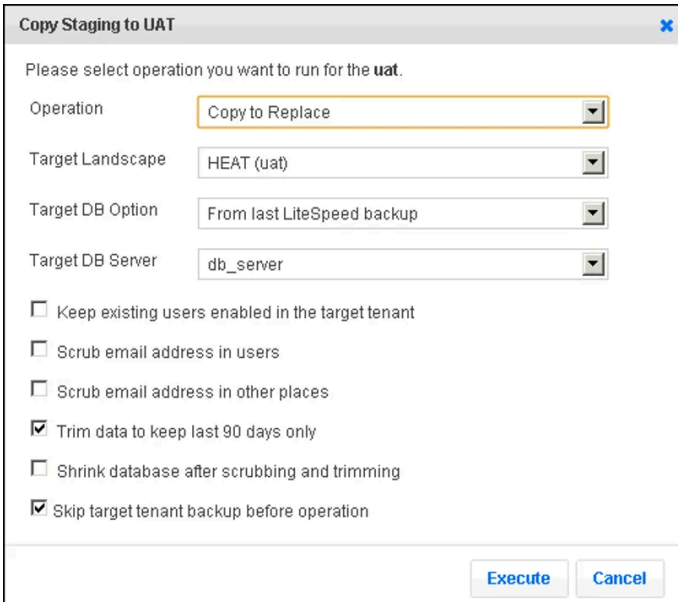
From the Operations Console you can perform a migration that overwrites all metadata and validation data on the target tenant instance with data from the source tenant instance. (See **Types of Data** for information about metadata and validation data.) This migration is equivalent to creating the first configuration on the target tenant instance as described in **Creating the First Staging or UAT Instance of the Tenant from the Production Instance of the Tenant**.

Perform the following steps to update a the configuration for a tenant instance by overwriting all metadata and validation data.

> Perform steps 1 to 4 in **Starting the Tenant Instance Migration** before you perform the steps in this section.

1. Check Advanced Mode.
2. In the **Operation** field in the **Copy** dialog box, select **Copy to Replace**. This configures the Operations Console to overwrite all metadata and validation data on the target tenant instance. The dialog box appears similar to the dialog box shown in **Copy Dialog Box (To Overwrite Metadata and Validation Data)**.
3. Perform the procedure described in **Updating Metadata, Validation Data, and Master Data (Copy Configuration & Master Data)**.

Copy Dialog Box (To Overwrite Metadata and Validation Data)



4. Click **Execute** to initiate the migration.

When the data migration finishes, the system creates a log entry for it and updates the status of the affected tenants and tenant instances in the Operations Console migration dashboard as shown in **Migration Dashboard** in **Starting the Data Migration**. The system also displays an informational message with the results of the migration. Check the **Events & Audit History** workspace to review the warnings and errors before continuing.

Updating Only Transactional Data (Copy Data Only)

> The relationship data between two business objects is not copied from the source to the target when you select the **Updating Only Transactional Data (Copy Data Only)** option.

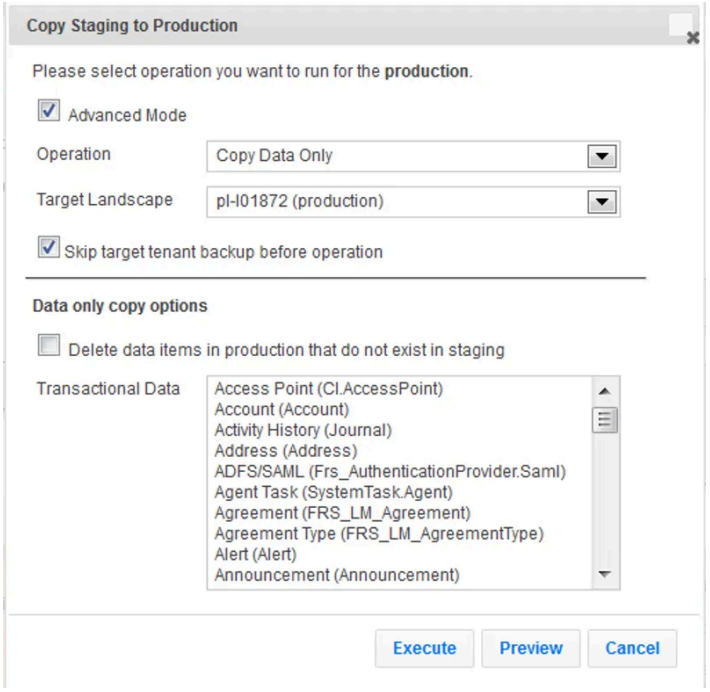
From the Operations Console you can migrate transactional data (data records, including incidents, changes, service requests, and so on) without migrating any other type of data.

Perform the following steps to update the configuration for a tenant instance by migrating transactional data. When you perform this migration, transactional data that exists on the target tenant instance is overwritten by the data on the source tenant instance.

> Perform steps 1 to 4 in **Starting the Tenant Instance Migration** before you perform the steps in this section.

1. Check Advanced Mode.
2. In the **Operation** field in the **Copy** dialog box, select **Copy Data Only** to configure the Operations Console to copy transactional data. The dialog box expands and appears similar to the dialog box shown in Copy Dialog Box (Migrate Only Transactional Data).

Copy Dialog Box (Migrate Only Transactional Data)



3. Check **Skip target tenant backup before operation** to select whether to back up the target tenant instance prior to the data migration.
    - To back up the target tenant instance, leave this checkbox empty. If you have not yet backed up the target tenant instance outside of the Operations Console, we highly recommend that you back it up here.
    - To not back up the target tenant instance, select this checkbox. If you have already backed up the target tenant instance outside of the Operations Console, it is generally not necessary to back it up here.
4. Use the **Data only copy options** section to select how to process transactional data that was deleted in the source tenant instance prior to the migration.

For example, assume that staging is the source tenant instance and production is the target tenant instance. If you deleted transactional data on the staging tenant instance as part of a reconfiguration procedure, you can select whether to delete or retain that data on the production tenant instance when you perform the data migration.

- To delete transactional data on the production tenant instance if that data was also deleted on the staging tenant instance, check **Delete data items in production that do not exist in staging**.
- To retain transactional data on the production tenant instance even though it was deleted on the staging tenant instance (that is, to *merge* the staging and production transactional data), leave this checkbox empty.

5. In the **Transactional Data** list, select the items to migrate.
6. Click **Execute** to initiate the migration or click **Preview** to create an XML-based patch file for review.

When the data migration finishes, the system creates a log entry for it and updates the status of the affected tenants and tenant instances in the Operations Console migration dashboard as shown in Migration Dashboard in Starting the Data Migration. The

system also displays an informational message with the results of the migration. Check the **Events & Audit History** workspace to review the warnings and errors before continuing.

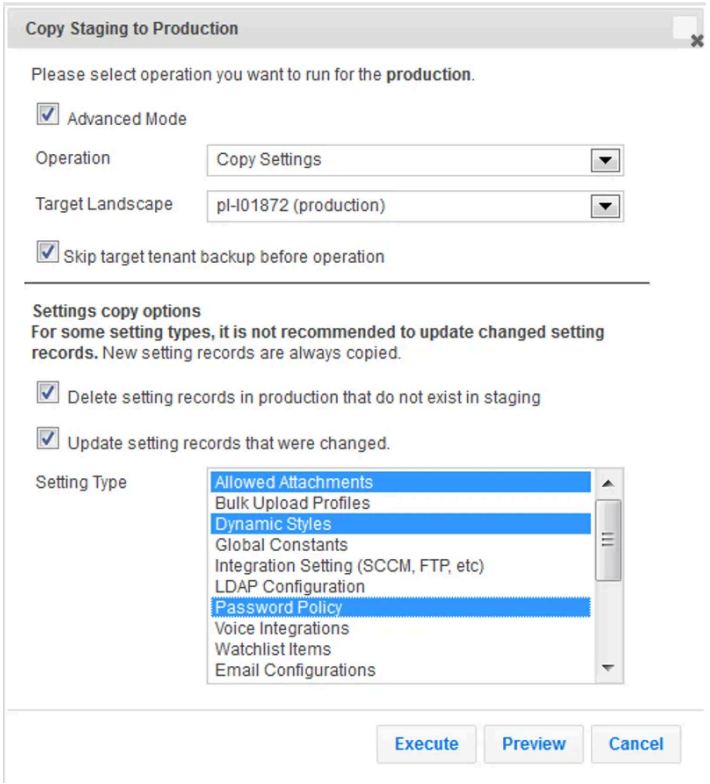
Updating Configuration Data (Copy Settings)

From the Operations Console you can migrate configuration data, also known as settings.

> Perform steps 1 to 4 in **Starting the Tenant Instance Migration** before you perform the steps in this section.

1. Check Advanced Mode.
2. In the **Operation** field in the **Copy** dialog box, choose **Copy Settings** to configure the Operations Console to copy configuration data. See **Copy Dialog Box (Migrate Settings)**.

Copy Dialog Box (Migrate Settings)



3. Check **Skip target tenant backup before operation** to select whether to back up the target tenant instance prior to the data migration.
    - To back up the target tenant instance, leave this checkbox empty. If you have not yet backed up the target tenant instance outside of the Operations Console, we highly recommend that you back it up here.
    - To not back up the target tenant instance, select this checkbox. If you have already backed up the target tenant instance outside of the Operations Console, it is generally not necessary to back it up here.
4. Do one of the following:
    - To delete configuration data (settings) on the target tenant instance if that data was also deleted on the source tenant instance, check **Delete setting records in *target_tenant_instance* that do not exist in *source_tenant_instance***.
    - To retain configuration data (settings) on the target tenant instance even though it was deleted on the source tenant instance (that is, to *merge* the source and target configuration data), leave this check box empty.

5. To update configuration data (settings) that has changed in both the source and the target tenant instances, check **Update setting records that changed**.
6. In the **Setting Type** list, select the configuration data types (settings) to migrate.
7. Click **Execute** to initiate the migration or click **Preview** to create an XML-based patch file for review.

When the data migration finishes, the system creates a log entry for it and updates the status of the affected tenants and tenant instances in the Operations Console migration dashboard as shown in Migration Dashboard in Starting the Data Migration. The system also displays an informational message with the results of the migration. Check the **Events & Audit History** workspace to review the warnings and errors before continuing.

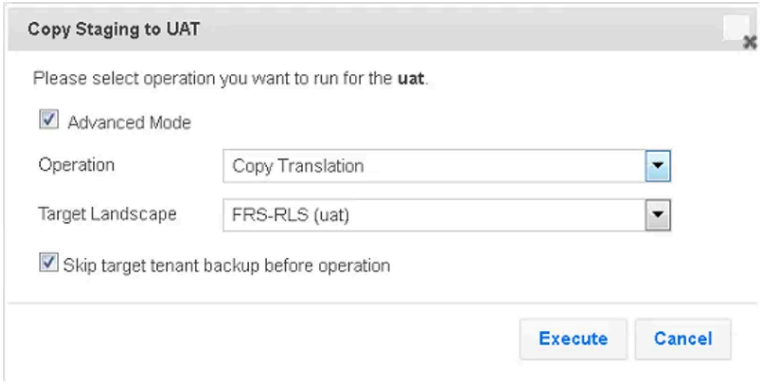
Updating Configuration Data (Copy Translation)

From the Operations Console you can migrate translation. The system connects to the source database and copies over any translations that are different or more recent in the target database. If there is no record in the source database but there is a record in the target, the system does not delete the record in the target database.

> Perform steps 1 to 4 in Starting the Tenant Instance Migration before you perform the steps in this section.

1. Check Advanced Mode.
2. In the **Operation** field in the **Copy** dialog box, choose **Copy Translation** to configure the Operations Console to copy translated data. See Copy Dialog Box (Migrate Settings).

Copy Dialog Box (Copy Translation)



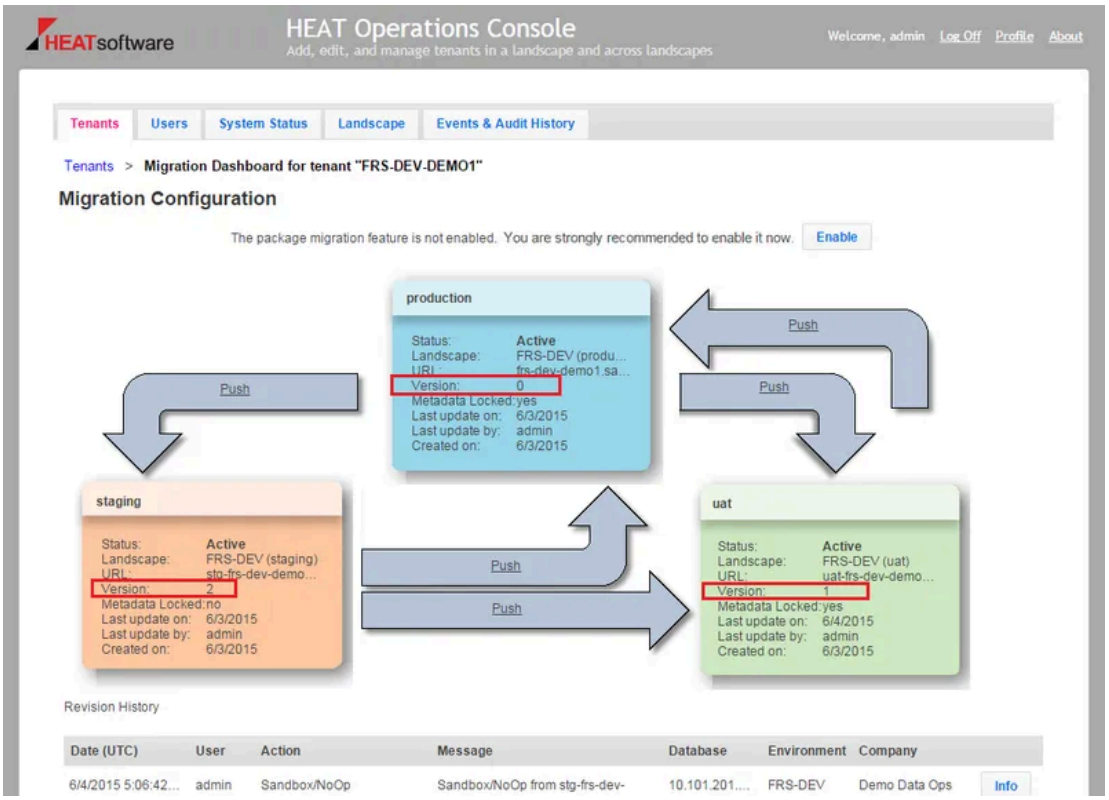
3. Check **Skip target tenant backup before operation** to select whether to back up the target tenant instance prior to the data migration.
   - To back up the target tenant instance, leave this checkbox empty. If you have not yet backed up the target tenant instance outside of the Operations Console, we highly recommend that you back it up here.
   - To not back up the target tenant instance, select this checkbox. If you have already backed up the target tenant instance outside of the Operations Console, it is generally not necessary to back it up here.
4. Click **Execute** to initiate the migration.

When the data migration finishes, the system creates a log entry for it and updates the status of the affected tenants and tenant instances in the Operations Console migration dashboard as shown in Migration Dashboard in Starting the Data Migration. The system also displays an informational message with the results of the migration. Check the **Events & Audit History** workspace to review the warnings and errors before continuing.

Increasing the Version Number (No Op [Increase Version Only])

You can only migrate data from one instance to another if there is a **Push** link and the system only displays a **Push** link if the tenant instances have different version numbers. (The system automatically increases the version number of a tenant instance after a successful migration.) Also, you can only migrate data from staging to production when the production instance of the tenant is at version 0. See **Version Numbers of Tenant Instances** for an example with **Push** links and different version numbers for each tenant instance.

Version Numbers of Tenant Instances

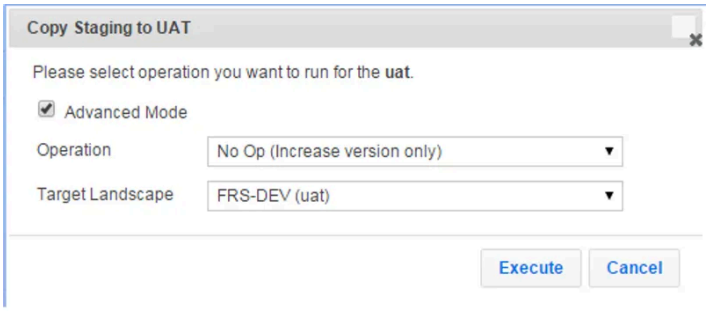


From the Operations Console you can increase the version number without changing any data by selecting **No Op (Increase version only)**.

> Perform steps 1 to 4 in **Starting the Tenant Instance Migration** before you perform the steps in this section.

1. Check Advanced Mode.
2. In the **Operation** field in the **Copy** dialog box, choose **No Op (Increase version only)** to increase the version number. See **Copy Dialog Box (No Op [Increase Version Only])**.

Copy Dialog Box (No Op [Increase Version Only])



3. Click **Execute** to initiate the migration.

When the data migration finishes, the system creates a log entry for it and updates the status of the affected tenants and tenant instances in the Operations Console migration dashboard as shown in **Migration Dashboard** in **Starting the Data Migration**. The system also displays an informational message with the results of the migration. Check the **Events & Audit History** workspace to review the warnings and errors before continuing.

8.7.1. Migration Scenarios Using Simple Mode and Advanced Mode

This section shows the steps to take when migrating data in different scenarios and when disabling the Ivanti Service Manager development project and using Simple Mode and Advanced Mode. It skips the conceptual information and only shows the procedure steps. See Appendix B: Migrating Data from One Tenant Instance to Another Using Simple Mode or Advanced Mode for conceptual information and details about these steps.

These examples migrate data by disabling the Ivanti Service Manager development project and using Simple Mode and Advanced Mode.

- Migrating Data from the Production Instance of the Tenant to the Staging or UAT Instance of the Tenant
- Refreshing the Staging or UAT Instance of the Tenant
- Performing a Quick Migration from the Staging or UAT Instance to the Production Instance
- Migrating Certain Types of Metadata and Validation Data from the Staging or UAT Instance to the Production Instance, or from the Staging Instance to the UAT Instance
- Migrating Certain Types of Metadata, Validation Data, and Master Data from the Staging or UAT Instance to the Production Instance, or from the Staging Instance to the UAT Instance
- Migrating Only Transactional Data from the Staging or UAT Instance to the Production Instance, or from the Staging Instance to the UAT Instance
- Overwriting All Metadata and Validation Data from the Staging or UAT Instance to the Production Instance, or from the Staging Instance to the UAT Instance
- Migrating Configuration Data (Settings) from the Staging or UAT Instance to the Production Instance, or from the Staging Instance to the UAT Instance
- Example: Migrating Service Level Data

Migrating Data from the Production Instance of the Tenant to the Staging or UAT Instance of the Tenant

In this scenario, you are just starting out and have only created an initial production instance of your tenant.

1. Log in to the Operations Console and click the **Tenants** tab.
2. Navigate to the landscape you will perform the migration in and click **Manage Migration**.
3. Click **Push** in the arrow representing the migration path to use. (See Migration Dashboard in Starting the Data Migration.) The system displays the **Copy *source_tenant_instance* to *target_tenant_instance*** (for example, **Copy Production to Staging**) dialog box.
4. Enter the following information for the first data migration to a tenant instance:
    - Select an option in the **Target DB Option** field.
    - In the **Target DB Server** field, if the target landscape contains more than one database server, select the server that hosts the Ivanti Service Manager configuration database.
    - (Optional) Check **Keep existing users enabled in the target tenant** to keep all the users, and not just the administrators, active.
    - (Optional) Check **Scrub email address in users** to scrub user emails.
    - (Optional) Check **Scrub email address in other places** to scrub email accounts saved in transactional data.

- (Optional) Check **Trim data to keep last 90 days only** to only copy transactional data that was created or modified within the last 90 days.

> Do not check this option if you are creating the database from the sample ("out of the box") database, because the records are likely to be older than 90 days in the sample database.

- (Optional) Check **Shrink database after scrubbing and trimming** to truncate unnecessary data when the tenant instance is created from a backup.

5. Click **Execute**.

Refreshing the Staging or UAT Instance of the Tenant

In this scenario, you have a database on the staging or UAT instance of the tenant and you want to update it with the latest version from the production instance of the tenant.

1. Log in to the Operations Console and click the **Tenants** tab.
2. Navigate to the landscape you will perform the migration in and click **Manage Migration**.
3. Click **Push** in the arrow representing the migration path to use. (See Migration Dashboard in Starting the Data Migration.) The **Copy *source_tenant_instance* to *target_tenant_instance*** (for example, **Copy Production to Staging**) dialog box appears.
4. Enter the following information for the first data migration to a tenant instance:
   - Select an option in the **Target DB Option** field.
   - In the **Target DB Server** field, if the target landscape contains more than one database server, select the server that hosts the Ivanti Service Manager configuration database.
   - (Optional) Check **Keep existing users enabled in the target tenant** to keep all the users, and not just the administrators, active.
   - (Optional) Check **Scrub email address in users** to scrub user emails.
   - (Optional) Check **Scrub email address in other places** to scrub email accounts saved in transactional data.
   - (Optional) Check **Trim data to keep last 90 days only** to only copy transactional data that was created or modified within the last 90 days.
   - (Optional) Check **Shrink database after scrubbing and trimming** to truncate unnecessary data when the tenant instance is created from a backup.
5. Click **Execute**.

Performing a Quick Migration from the Staging or UAT Instance to the Production Instance

1. Log in to the Operations Console and click the **Tenants** tab.
2. Navigate to the landscape you will perform the migration in and click **Manage Migration**.
3. Click **Push** in the arrow representing the migration path to use. (See Migration Dashboard in Starting the Data Migration.) The **Copy *source_tenant_instance* to *target_tenant_instance*** (for example, **Copy Staging to Production**) dialog box appears.
4. In the **Copy** dialog box, make these selections:
   - Do not check Advanced Mode.
   - (Optional) Check Skip target tenant backup before operation to not perform a database backup.

- (Optional) Under **Copy the following selected data**, check Service request if you want to migrate service request information, and check Escalation schedule if you want to migrate escalation schedule information.

5. Click Execute.

Migrating Certain Types of Metadata and Validation Data from the Staging or UAT Instance to the Production Instance, or from the Staging Instance to the UAT Instance

1. Log in to the Operations Console and click the **Tenants** tab.
2. Navigate to the landscape you will perform the migration in and click **Manage Migration**.
3. Click **Push** in the arrow representing the migration path to use. (See Migration Dashboard in Starting the Data Migration.) The **Copy *source_tenant_instance* to *target_tenant_instance*** (for example, **Copy Staging to Production**) dialog box appears.
4. In the **Copy** dialog box, check Advanced Mode.
5. In the **Copy** dialog box, for the **Operation** field, choose **Copy Configuration**. (See Copy Dialog Box (Migrate Metadata and Validation Data) in Appendix B: Migrating Data from One Tenant Instance to Another Using Simple Mode or Advanced Mode.)
6. (Optional) Check **Skip target tenant backup before operation** to not back up the target tenant instance prior to the data migration.
7. (Optional) Check the items to delete if those objects were deleted on the target tenant instance in the **Delete metadata items in *target_tenant_instance* that do not exist in *source_tenant_instance*** section. If you do not select any checkboxes in this section, all data is merged, and no data on the source tenant instance is deleted.
8. (Optional) Check the items under the **Replace metadata items in *target_tenant_instance* that are newer than those in *source_tenant_instance*** section to delete items that exist on both the source tenant instance and the target tenant instance when the version on the target tenant instance is newer than the version on the source tenant instance. If you do not select any checkboxes in this section, all data on the target tenant instance is *overwritten* by data from the source tenant instance.
9. (Optional) Check the items under the **Custom defined validation list data objects** section to delete customized validation list items on the target tenant instance that were also deleted on the source tenant instance.
10. Click **Execute**.

Migrating Certain Types of Metadata, Validation Data, and Master Data from the Staging or UAT Instance to the Production Instance, or from the Staging Instance to the UAT Instance

1. Log in to the Operations Console and click the **Tenants** tab.
2. Navigate to the landscape you will perform the migration in and click **Manage Migration**.
3. Click **Push** in the arrow representing the migration path to use. (See Migration Dashboard in Starting the Data Migration.) The system displays the **Copy *source_tenant_instance* to *target_tenant_instance*** (for example, **Copy Staging to Production**) dialog box.
4. In the **Copy** dialog box, check Advanced Mode. The system displays more options in the dialog box.
5. In the **Copy** dialog box, for the **Operation** field, choose **Copy Configuration & Master Data**. (See Copy Dialog Box (Migrate Metadata, Validation Data, and Master Data) -- Top Half in Appendix B: Migrating Data from One Tenant Instance to Another Using Simple Mode or Advanced Mode.)
6. Fill in the top portion of the **Copy** dialog box (the settings for metadata and validation data) as described in Migrating Certain Types of Metadata and Validation Data from the Staging or UAT Instance to the Production Instance, or from the Staging Instance to the UAT Instance.

7. (Optional) Check **Delete data items in *target_tenant_instance* that do not exist in *source_tenant_instance*** to delete master data on the target tenant instance that was also deleted on the source tenant instance.
8. (Optional) Check any items under the **Copy the following selected data** section to include them when the data is migrated.
9. (Optional) Select one or more items to migrate from the **Additional Data** list.
10. Click Execute.

Migrating Only Transactional Data from the Staging or UAT Instance to the Production Instance, or from the Staging Instance to the UAT Instance

1. Log in to the Operations Console and click the **Tenants** tab.
2. Navigate to the landscape you will perform the migration in and click **Manage Migration**.
3. Click **Push** in the arrow representing the migration path to use. (See Migration Dashboard in Starting the Data Migration.) The system displays the **Copy *source_tenant_instance* to *target_tenant_instance*** (for example, **Copy Staging to Production**) dialog box.
4. In the **Copy** dialog box, check Advanced Mode.
5. In the **Copy** dialog box, for the **Operation** field, choose **Copy Data Only**. (See Copy Dialog Box (Migrate Only Transactional Data) in Appendix B: Migrating Data from One Tenant Instance to Another Using Simple Mode or Advanced Mode.)
6. (Optional) Check **Skip target tenant backup before operation** to not back up the target tenant instance prior to the data migration.
7. (Optional) Check **Delete data items in *target_tenant_instance* that do not exist in *source_tenant_instance*** under the **Data only copy options** section to delete transactional data on the target tenant instance if that data was also deleted on the source tenant instance.
8. (Optional) In the **Transactional Data** list, select the items to migrate.
9. Click **Execute**.

Overwriting All Metadata and Validation Data from the Staging or UAT Instance to the Production Instance, or from the Staging Instance to the UAT Instance

1. Log in to the Operations Console and click the **Tenants** tab.
2. Navigate to the landscape you will perform the migration in and click **Manage Migration**.
3. Click **Push** in the arrow representing the migration path to use. (See Migration Dashboard in Starting the Data Migration.) The system displays the **Copy *source_tenant_instance* to *target_tenant_instance*** (for example, **Copy Staging to Production**) dialog box.
4. In the **Copy** dialog box, check Advanced Mode.
5. In the **Copy** dialog box, for the **Operation** field, choose **Copy to Replace**. (See Copy Dialog Box (To Overwrite Metadata and Validation Data) in Appendix B: Migrating Data from One Tenant Instance to Another Using Simple Mode or Advanced Mode.)
6. Enter the following information:
   - Select an option in the **Target DB Option** field.
   - In the **Target DB Server** field, if the target landscape contains more than one database server, select the server that hosts the Ivanti Service Manager configuration database.
   - (Optional) Check **Keep existing users enabled in the target tenant** to keep all the users, and not just the administrators, active.

- (Optional) Check **Scrub email address in users** to scrub user emails.
- (Optional) Check **Scrub email address in other places** to scrub email accounts saved in transactional data.
- (Optional) Check **Trim data to keep last 90 days only** to only copy transactional data that was created or modified within the last 90 days.
- (Optional) Check **Shrink database after scrubbing and trimming** to truncate unnecessary data when the tenant instance is created from a backup.
- (Optional) Check Skip target tenant backup before operation to not back up the target tenant instance prior to the data migration.

7. Click **Execute**.

Migrating Configuration Data (Settings) from the Staging or UAT Instance to the Production Instance, or from the Staging Instance to the UAT Instance

1. Log in to the Operations Console and click the **Tenants** tab.
2. Navigate to the landscape you will perform the migration in and click **Manage Migration**.
3. Click **Push** in the arrow representing the migration path to use. (See Migration Dashboard in Starting the Data Migration.) The system displays the **Copy *source_tenant_instance* to *target_tenant_instance*** (for example, **Copy Staging to Production**) dialog box.
4. In the **Copy** dialog box, check Advanced Mode.
5. In the **Copy** dialog box, for the **Operation** field, choose Copy Settings. (See Copy Dialog Box (Migrate Settings) in Appendix B: Migrating Data from One Tenant Instance to Another Using Simple Mode or Advanced Mode.)
6. (Optional) Check **Skip target tenant backup before operation** to not back up the target tenant instance prior to the data migration.
7. (Optional) Check **Delete setting records in *target_tenant* that do not exist in *source_tenant*** to delete configuration data (settings) on the target tenant instance if that data was also deleted on the source tenant instance.
8. (Optional) To update configuration data (settings) that has changed in both the source and the target tenant instances, check **Update setting records that changed**.
9. In the **Setting Type** list, select the configuration data types (settings) to migrate.
10. Click **Execute**.

Example: Migrating Service Level Data

This example shows how to migrate service level data such as service level agreements, operational level agreements, underpinning contracts, service level targets, and service level packages. All of these are made up of master data and transactional data.

1. Log in to the Operations Console and click the **Tenants** tab.
2. Navigate to the landscape you will perform the migration in and click **Manage Migration**.
3. Click **Push** in the arrow representing the migration path to use. (See Migration Dashboard in Starting the Data Migration.) The system displays the **Copy *source_tenant_instance* to *target_tenant_instance*** (for example, **Copy Staging to UAT**) dialog box.
4. In the **Copy** dialog box, for the **Operation** field, select **Copy Master Data Only**.
5. (Optional) Check **Delete data items in *target_tenant_instance* that do not exist in *source_tenant_instance*** to delete master data on the target tenant instance that was also deleted on the source tenant instance.

6. Under the **Copy the following selected data** section, check Service request and Escalation schedule.
7. From the **Additional Data** list, hold down the control key and select all of the following:
    - Service Agreement (ServiceAgreement)
    - Service Level Agreement (ServiceLevelAgreement.SLA)
    - Service Level Objective (ServiceLevelObjective)
    - Service Level Package (ServiceLevelPackage)
    - Service Level Target (ServiceLevelTarget)
    - Service Level Target – General (ServiceLevelTargetGeneral)
    - Vendor (Vendor)
8. Click Execute.

8.8. Appendix C: Using the ISM Development Project to Migrate Data

> The information in this appendix is copied directly from the Ivanti Service Manager online help. It describes how to use the Ivanti Service Manager Configuration Console to create packages, which can be used when migrating data. The online help contains additional information about the Ivanti Service Manager development project, such as cautions and limitations about the using the Ivanti Service Manager development project, terminology, information about logs and validation, and examples.
>
> You perform these tasks in the Ivanti Service Manager Configuration Console *and not in the Operations Console*.

This appendix contains the following sections:

- Create a project. See **Creating a Project**.
- Optional. View the project. See **Viewing a Project**.
- Make changes to Ivanti Service Manager. See **Creating a Transaction Set**.
- Optional. View the transactions sets. See **Viewing Transaction Sets**.
- Create a package. See **Creating a Package**.
- Optional. View the package. See **Viewing a Package**.
- Assign transaction sets to a package. See **Assigning Transaction Sets to a Package**.
- Add an aspect to a package. See **Adding an Aspect to a Package**.
- Map a package. See **Adding Packages to a Master Package (Mapping)**.
- Close the package. See **Closing a Package**.
- Optional. Export a package to another system. See **About Exporting a Package**.
- Optional. Import a package from another system. See **Importing a Package**. If you get errors, see **Troubleshooting Package Import Errors**.
- Optional. Save the package as a file. See **Saving a Package as a File**.
- Optional. Delete a package. See **Deleting a Package**.

After performing all of these steps, you can migrate the data to the target system.

For information about the packages that have been migrated, see the "Viewing the Patch Log" topic in the Ivanti Service Manager online help.

Creating a Project

You must create a project before you make any configuration changes. If you do not make any projects, all changes are logged to the default project. Use projects to organize the changes made to Ivanti Service Manager. After you create a project and *before* you make any configuration changes, always select your project from the **Project** drop-down list on the top right.

Follow these steps, on the system (such as staging) on which to make changes, to create a project:

1. Log in to Ivanti Service Manager using your named account and not as an administrator. See the Ivanti Service Manager topic called "About Using Your Named Account to Log In" for information about why you must use your named account to log in.
2. Click Configure Application to go to the Configuration Console.
3. Click Build > HEAT Development Project > Projects. The system displays the **Projects** workspace.

4. Click New Record. The system displays a blank project page.
5. Enter a name and description for the project and click Save.
6. Click Refresh. You can now see the new project name under the **Project** drop-down menu in the top right section of the toolbar (next to the **Help** link).

Viewing a Project

1. From the Configuration Console, click Build > HEAT Development Project > Projects to display the **Projects** workspace. The system displays a list of all the projects.
2. Double click a project name to see details of the project.
3. Click the **Transaction Set** tab to see a list of the transaction sets in this project. See **Viewing Transaction Sets**.
4. Click the **Transaction Detail** tab to see the details of each transaction set. See **Viewing Transaction Sets**.
5. View the activity history by clicking the **Activity History** tab, any attachments by clicking the **Attachment** tab, or the audit history by clicking the **Audit History** tab.

Creating a Transaction Set

Every time you add, delete, or modify the Ivanti Service Manager configuration, the system records the changes. These recorded changes are called a transaction set. For example, every time that you save changes for a business object, the system records the changes in a transaction set. The system tracks every change and assigns each change a sequential number.

1. From the Configuration Console, click Build > HEAT Development Project > Projects to display the **Projects** workspace. The system displays a list of all the projects.
2. From the **Project** drop-down menu in the toolbar, select the project to which the changes you are going to make will belong.
3. Make changes to Ivanti Service Manager as normal. For example, you can create a new business object, form, or list.

- In certain cases, when you make changes to Ivanti Service Manager, you must enter your username and password. When you create and later export the transaction set as part of a package, the package file exposes the user name and password.
- To avoid exposing this potentially sensitive information, we recommend using a global constant for the username and password when setting up an integration connection, rather than entering actual values. For more information, see the "Using Global Constants to Hide Login ID and Password Information" topic in the Ivanti Service Manager documentation. For example of using a global constant in this situation, see the "Viewing, Cloning, and Deleting Web Service Connections" topic in the Ivanti Service Manager documentation.
- If you do not use a global constant for the user name and password, the system removes that information from the transaction set.

Viewing Transaction Sets

From the **Transaction Sets** page, you can view the relevant changes that are linked to a transaction set. You can search by transaction set ID. Use the transaction set ID when creating a package.

1. From the Configuration Console, click Build > HEAT Development Project > Transaction Sets to open the **Transaction Sets** workspace. The system displays a list of all the transaction sets.

The system displays the following information:

Field	Description
ID	The ID for this transaction set.
Created On	The date when the transaction set was created.
User ID	The ID of the user who created the transaction set.
Session ID	The ID of the session when the transaction set was created.
DB Revision	The database revision.
Source	Specifies where the transaction was made. *AdminUI* means that the change was made in the Configuration Console, and a tenant URL means that the change was made from a package migration.
Project Name	The name of the Ivanti Service Manager development project to which this transaction set is attached.

2. Double click a transaction set and click the **Transaction Detail** tab to see the following information:

Field	Description
Project Name	The name of the Ivanti Service Manager development project to which this transaction set is attached.
Transaction Set ID	A sequential number linked to the change. A transaction set may contain zero or more changes. (A zero change indicates the change was aborted, and was not recorded.)
Detail ID	A sequential number linked to the change. A detail may contain zero or more changes. (A zero change indicates the change was aborted, and was not recorded.)
User ID	The ID of the user who created the transaction set.
Operation Date Time	The date and time when the change was recorded.
Operation Type	The type of operation. Can be **C** for new record, **U** for modification, or **D** for deleted.
Object Type	The type for which the change applies. The values can be **BusinessObject** for a business object change, **Form** for a form change, or a business object name such as **AccountStatus#** for change in its data, etc.
Business Object	The name of the business object to which the change is linked, if any.
Entity Name	The key for the business object record itself. In some cases, it is the GUID and in other case, it is a name, such as a business object name, a form name, or a role name.
Source ID	The key for the business object record itself. In some cases, it is the GUID and in other case, it is a name, such as a business object name, a form name, or a role name.
To Publish	Specifies if this transaction set can be published: • **Y**: The change is packaged. • **N**: This change is a side effect of another change that is published.

Creating a Package

Follow these steps, on the system (such as staging) on which you have made changes, to create a package:

1. From the Configuration Console, click Build > HEAT Development Package > Packages. The system displays the **Packages** workspace.
2. Click New Record. The system displays a blank package page.
3. Enter a descriptive name for the package. The name must be unique across the system.
4. From the **Type** drop-down list, select if this is an incremental or master package. You cannot edit the type after you save create the package.
    - An incremental package is a single change. It can contain a set of metadata changes and can be imported and exported. There are two types of incremental packages: the first type is exported from the tracked changes in the data and metadata. Every data and metadata change is tracked with a sequence number, which specified the import order. The second type is a result of comparing the full metadata from two tenant instances (such as UAT and staging). There is no sequence associated with these changes. This content comes from the traditional Operations Console migration.
    - A master package contains one or more incremental, master, or file packages. The packages in the master package are called sub-packages. For example, you may want to create a master package with several incremental packages (sub-packages) in it.
5. In Ivanti Service Manager Release 2025, the **Predicate** field is for internal use only. Do not enter anything here.
6. Enter a description.
7. Click Save.
8. Click **Refresh** to see the **Close Package**, **Save As File**, and **Export Package** icons on the top and the tabs at the bottom.

Viewing a Package

1. From the Configuration Console, click Build > HEAT Development Package > Packages to open the **Packages** workspace. The system lists the Ivanti Service Manager development packages.
2. Double click a package to view its details.

The system displays the following tabs:

- **Transaction Set** tab (for incremental packages only). See **Viewing Transaction Sets**.
- **Transaction Detail** tab (for incremental packages only). See **Viewing Transaction Sets**.
- **Package Mapping** tab (for master packages only). See **Adding Packages to a Master Package (Mapping)**.
- **Package Aspect** tab. See **Adding an Aspect to a Package**.
- **Activity History** tab. See the "About Activity History" topic in the Ivanti Service Manager online help.
- **Attachment** tab. See the "Adding an Attachment" topic in the Ivanti Service Manager online help.
- **Audit History** tab. See the "About Accessing the Audit History" topic in the Ivanti Service Manager online help.

Assigning Transaction Sets to a Package

You must create an incremental type package before you can assign transaction sets to it. See **Creating a Package**.

Follow these steps, on the system (such as staging) on which you have made changes, to assign transaction sets to a package:

1. From the Configuration Console, click Build > HEAT Development Project > Transaction Sets to open the **Transaction Set** workspace.
2. Select the transaction sets to include in the package.
3. Click Assign To Package.
4. Use the drop-down menu to select the name of the package to which to assign the transaction sets and click OK.

> You can also assign a transaction set to a package by clicking Build > HEAT Development Package > Packages to open the **Packages** workspace, clicking the **Transaction Set** tab, and clicking Add Change Set.

5. To check that the transactions set are in the package, do the following:
   a. Click Build > HEAT Development Package > Packages. The system displays the **Packages** workspace.
   b. Open the package. (If you do not see it, click Refresh.) The system opens the package.
   c. Click the **Transaction Set** tab to view all of the transaction sets.
6. To remove a transaction set from a package, do the following:
   a. Click Build > HEAT Development Package > Packages. The system displays the **Packages** workspace.
   b. Open the package. (If you do not see it, click Refresh.) The system opens the package.
   c. Click the **Transaction Set** tab to view all of the transaction sets.
   d. Highlight the transaction set.
   e. Click **Remove From Package**.
7. To add a transaction set to another package, do the following:
   a. Click Build > HEAT Development Package > Packages. The system displays the **Packages** workspace.
   b. Open the package. (If you do not see it, click Refresh.) The system opens the package.
   c. Click the **Transaction Set** tab to view all of the transaction sets.
   d. Highlight the transaction set to add to the package.
   e. Click **Assign To Package**.
   f. Select a package name from the drop-down list and click **OK**.

Adding an Aspect to a Package

You can only add aspects to a package if it is open. When you created the package, the system copied the aspects from the original package. You can remove the original aspects and add new ones.

1. From the Configuration Console, click Build > HEAT Development Project > Transaction Sets to open the **Transaction Sets** workspace.
2. Assign one or more transaction sets to a package.
3. Click the **Package Aspect** tab.

A package can have zero or more aspects. When the package is imported into target tenant instance, these aspects are entered into or removed from the target tenant instance to flag the aspect presence.

4. View the following information on the **Package Aspect** tab:

Field	Description
To Uninstall	Specifies if the aspect should be removed.
Version	Numeric whole number for this aspect version. The version number for an aspect grows in its development life.
ID	A unique name for the aspect.
Vendor Name	The name of the vendor who provides the aspect. An example is (such as feature/component), Ivanti Software.
Vendor ID	A unique ID that identifies the aspect provider. Currently, the system uses "Ivantisoftware" for aspects provided by Ivanti Software. In future releases, other providers will have a unique vendor ID.
Created By	The ID of the administrator who added the aspect.
Created On	The date and time when the aspect was created.
Modified By	The ID of the administrator who modified the aspect.
Modified On	The date and time when the aspect was modified.

5. Click **New Package Aspect**. The system displays the **New Package Aspect** dialog box.
6. Check **Is Uninstalled** if this package aspect is uninstalled.
7. Enter the following information:
    - **Version number**: Any number.
    - **ID**: Any text.
    - **Vendor ID**: Any text.
8. Click **Save**.

Adding Packages to a Master Package (Mapping)

If you have a master package, you can add incremental or other master packages to it. In this way, the master package contains many changes. This eliminates the need to export incremental packages one at a time.

1. From the Configuration Console, click Build > HEAT Development Package > Packages to open the **Packages** workspace.
2. Open a master package.
3. Click the **Package Mapping** tab.
4. Click Assign Package. The system displays the **New Master Package Mapping** form.
5. In the **Sub Package** field, click the search icon to display a list of packages that can be mapped.
6. Select a sub-package and click **Use Selected HEAT Release Package**.
7. Enter a value into the **Relative Order** field. This determines the relative order of this sub-package compared to other sub-packages in the master package. The system applies the sub-packages to the tenant in this order (when you export and import

the package).

8. Click **Save**.
9. Repeat steps 4 through 8 to add additional sub-packages to the master package.

Closing a Package

You must close a package before you can use the Operations Console to migrate it from staging to production. Closing a package means that you cannot do any more work on it, such as adding or removing packages, adding or removing predicates, changing aspects, and so on. You do not have to close a package to export it.

We recommend that you only close a package after you have completed testing on it and you are ready to migrate it from either staging or UAT to production. After you close a package, you cannot reopen it.

When a package is closed, the system does not display the **Add Change Set** and **Remove From Package** buttons on the **Transaction Set** tab, nor does it display the **New Package Aspect** and **Delete** buttons on the **Package Aspect** tab.

Follow these steps to close a package.

1. From the Configuration Console, click Build > HEAT Development Package > Packages. The system displays the **Packages** workspace.
2. Highlight the name of the package to close.
3. Click Close Package.
4. At the confirmation prompt, click OK.

About Exporting a Package

You can export a package of any type as an XML file and send it to another system, even if the package is empty. You can export any open or closed package. You can use any text editor to view the exported package.

- About Compacting Export Data
- Exporting a Package

About Compacting Export Data

> Compacting export data is a beta feature. Use it with caution.

Starting in HEAT Service Management Release 2016.1, you can compact the transaction sets for a package when you export it. If you make multiple changes to one business object consecutively, compacting combines the individual changes into one compact change. When you import the package, the system only implements the one compacted change instead of the individual changes, which speeds up the process of importing a package.

For example, if you changed a quick action 12 times consecutively, then without compacting, the package would have 12 transaction sets. With compacting, there is only one transaction set.

There are a few caveats to this:

- The system only compacts changes to metadata.
- The system does not combine creating a business object and changing a business object into one compact package. The system only compacts changes to a business object and in this instance, changing a business object does not include creating it.

- The system only compacts changes to the same business object of the same metadata type. If you change two business objects, the system does not compact these two changes together.

Exporting a Package

1. From the Configuration Console, click Build > HEAT Development Package > Packages to open the **Packages** workspace.
2. Highlight the name of the package to export.
3. Click **Export Package**. The system gives you the option of compacting the transaction sets. This is a beta feature.
4. Click Yes to compact the export data or No to not compact the export data. The system exports the package.

Importing a Package

- You should never import a package into the production environment from the Configuration Console if you have the Operations Console in place, as this could cause system instability.
- We recommend that you always back up your database before importing a package.
- Importing a package is an irreversible operation and cannot be undone.

You can import a package that has been previously exported. For master packages, the system imports the subpackages in the given relative order. The system imports the package if its predicate evaluates to true. If the predicate evaluates to false or an error, the system does not import the package or its subpackages. After importing the package, the system uses the package aspects to update the tenant.

Starting in HEAT Service Management Release 2016.1, the system scans the import package and assigns one of three impact levels (high, medium, and low) to each package.

Impact Level	Definition
High	• Contains a database schema change. • Deletes a business object. • Updates a stored property for a field of a business object. • Updates the **IsLocalized** property for a field of a business object. • Updates the data type, length, or precision for a field of a business object. • Updates a unique property for a field of a business object. • Updates a nullable property for a field of a business object. • Changes the index. • Changes the full-text search index. • Adds or deletes an audit field.
Medium	• Contains changes to the metadata for a business object. • Adds a business object. • Changes the metadata of a business object, other than the changes described above. • Changes or overrides a business rule. • Changes or deletes a form, list, layout, quick action, pick list definition, dashboard, chart part, list part, or special part. • Changes or deletes a role. • Changes or deletes branding. • Changes a workflow, counter, view part, global constant, request offering, or report. • Changes the email configuration. • Changes integration settings. • Changes the LDAP configuration or authentication provider. • Changes the password policy. • Changes an escalation schedule.
Low	• Does not contain a database schema change or changes to the metadata for a business object, or anything listed above. • Adds a form, list, layout, quick action, pick list definition, dashboard, chart part, list part, special part. • Add a role. • Adds branding.

Unlike in previous releases, starting in HEAT Service Management Release 2016.1, you can sometimes import a package even when the system is in metadata read-only mode.

When the system is in metadata read-only mode, note the following:

- For high impact packages, you can only validate the import package. You cannot import the package.
- For medium impact packages, you can validate and import the package, but there may be risks with importing the package. We highly recommend that you back up the database first.

- For low impact packages, you can validate and import the package. We highly recommend that you back up the database first.

Follow these steps to import a package:

1. From the Configuration Console, click Build > HEAT Development Package > Packages to open the **Packages** workspace.
2. Click Import Package.
3. Click **Browse...** to navigate to the import package.
4. Highlight the file and click **Open**. The system displays a dialog box that lists the file name of the import package.
5. Check Show Detail to see the details of the changes in the import package and then click Preview Impact to scan the complexity of the import package. Depending on the impact of the import package, the system displays a screen that lists the details of the import package impact.

If you do not check Show Detail, the system only displays the first high or medium level change in the package.

6. Click Close. The system displays the **Import Package** dialog box stating that you should back up the database before importing the package, and that this action is irreversible.
7. Do one of the following:
    - Select Validate to check the integrity of the package without applying it. If there are any errors, you can go back and fix them before importing the package.
    - Select Validate and apply if no errors to import the package only if there are no errors. Note that the system may import the package even if there are errors on the application side. The system displays a message stating if there are any errors and where they are, either on the application side or on the validation side.
    - Select Apply without validation. We do not recommend this option. With this option, you assume all responsibility to correct any errors after the import. This imports the package even if there are errors. You should always validate the package and understand the errors first, and then select this option only if you agree to ignore the errors.
8. Click Execute
9. At the confirmation prompt, click **Yes**. The system starts the import process.

> If the package contains many changes, the import process can take a long time. Your system may time out or you may accidentally get disconnected from the system due to loss of wifi or VPN. If this happens, log in again. The system continues to import the package and display the importing message. However, the system may not dismiss the importing message, even after the import process has finished. In this case, click Cancel or Refresh to dismiss it.

The system displays a message about the success or failure of the import, including specifying if there are any warnings or errors.

> If there are any errors, you cannot import the package. If you do not understand the impact of the errors, we recommend that you restore your database back to the production environment, as the error may disrupt your production operation. See Troubleshooting Package Import Errors or contact Ivanti Software for further information about the errors.

10. Click **Close**.
11. To view the imported package, do the following:

    a. Click Build > HEAT Development Project > Projects to open the **Projects** workspace.

    b. Under the **Name** column, double-click the project that you just imported, to open it.

Troubleshooting Package Import Errors

If you get any errors while importing a package, the package import fails. You must correct the errors before importing the package. If you get any warnings, the import continues; however, we strongly recommend that you correct the warnings before importing the package.

The system displays warnings in orange text and errors in red text.

Try the following for troubleshooting import package warnings and errors:

- If the error message references a sequence number, open the package in a text editor such as Notepad and look for that sequence number. Examine the package for information.
- From the Configuration Console, click Build > HEAT Development Project > Transaction Sets to open the **Transaction Set** workspace. Open the transaction set associated with the warning or error. Examine the transaction set for information.

Saving a Package as a File

You can save an incremental or master package as a file (but you cannot save a package as a file if it is already a file) so that you can have a record of it at a certain time. This is similar to versioning the package. You can export or import packages saved as files. All aspects in the package are copied to the file, but you can edit them after they are copied to the file.

Follow these steps to save a package as a file:

1. From the Configuration Console, click Build > HEAT Development Package > Packages to open the **Packages** workspace.
2. Highlight the name of the package to save.
3. Click **Save As File**.
4. Enter a new name or description, or leave the default values.
5. Click **Save**. The system saves the package as a file.
6. Click **Refresh** to see the new file in the file list.

Deleting a Package

You can only delete a package if it is open and if it is not being used by another package.

1. From the Configuration Console, click Build > HEAT Development Package > Packages to open the **Packages** workspace.
2. Highlight the name of the package to delete.
3. Click the **delete** icon.

8.9. Appendix D: Working with Packages from the App Store

- About the App Store
- Accessing the App Store
- Downloading and Importing an App Store Package
- Troubleshooting Problems with App Store Packages

About the App Store

Starting in HEAT Service Management Release 2015.1, we released a new applications store (App Store) where you can download optional software created by both Ivanti Software and other companies. The optional software in the App Store is bundled into packages.

The App Store is a collection of apps that can extend your Ivanti Software applications. Some apps include special packages that you can import into Ivanti Service Manager to modify the behavior of the system; other apps are integrations to other applications.

To create your own app, see the link “How to Develop an App” on the home page of the App Store.

The App Store provides different certification levels for the apps, which help you understand the level of support provided for a given app. Some apps are available on an “as-is” basis as a sample, and are available to use at your own risk. Some apps are supported directly by Ivanti Software and some apps come from certified software partners.

Not all apps have an associated package; some require licensing additional software from partners to enable the functionality. These requirements are described in the app description.

Accessing the App Store

Follow these steps to access the App Store:

1. Navigate to https://support.heatsoftware.com.
2. Do the following:

   a. Enter your user name and password.

   b. Press Login.

The system may require you to log in using external authentication. If so, enter your user name and click Sign in with *your_company_name*.

3. If the system prompts you, select the Self Service role and click Submit. The system opens to the Self Service home page.
4. From the top tool bar, click More... and select App Store. You are now in the App Store.

Downloading and Importing an App Store Package

1. Access the App Store as described in **Accessing the App Store**.
2. Open the app to use. Download the documentation associated with the app, and ensure that you meet all of the prerequisites.
3. Download the .metadatapatch files associated with the app. These are the actual package files to be imported.

4. Import the package into the staging instance of your tenant by following the directions in **Importing a Package**. This automatically creates a project for this package in your staging system.

> You should always validate any new package before applying it to a production environment, even partner certified or Ivanti Software-supplied packages. Package behavior can vary based on your specific configuration. By doing the validation in staging, any necessary changes can also be applied.

5. Test the package on the staging instance of your tenant.
6. If you need to make any changes to your system based on your testing, do the following:

   a. Open the project that the system created when you imported the package. Ensure that this is the active project so that the system adds any changes that you make to this project.

   b. Make the required changes.

   c. Create a new package for the changes. See **Creating a Package**.

   d. Export the package. See **Exporting a Package**.

> If you did not make any changes, you do not need to create a new package as you can just use the one that you imported in step 3.

7. Refresh the UAT instance of your tenant. It is very important that the UAT instance accurately reflects the current state of the production instance of your tenant, to get an accurate UAT assessment.
8. Import the package into the UAT instance of your tenant by following the directions in **Importing a Package**. This automatically creates a project for this package in your UAT system.
9. Test the package on the UAT instance of your tenant.
10. If you need to make any changes to your system based on your testing, do the following:

    a. Open the project that the system created when you imported the package. Ensure that this is the active project so that the system adds any changes that you make to this project.

    b. Make the required changes.

    c. Create a new package for the changes. See **Creating a Package**.

    d. Export the package. See **Exporting a Package**.

> If you did not make any changes, you do not need to create a new package as you can just use the one that you imported in step 7.

11. Close the package when you are satisfied with it and know that you will make no more changes.
12. Back up the production instance of your tenant.
13. sample
14. Import the package. If you have locked the production instance of your tenant, enable the Ivanti Service Manager development package and then push the package. (See **The Four Ways to Migrate Data** for a list of the ways to migrate data.)
15. Import the package into the production instance of your tenant by following the directions in **Importing a Package**. The system analyzes the impact of the package.

16. If the impact is green or yellow, you can import the package directly. We recommend doing this during off hours, so users are not impacted by the change.
17. If the impact is red, create an incident with Support requesting that they push the package to the production instance of your tenant. After they receive the request, Support moves the package into the production environment.

Troubleshooting Problems with App Store Packages

- You must import the package into the staging or UAT instance of your tenant. Support cannot import the package for you.
- You must close the package before you send it to Support.
- If Support finds any errors while migrating the package, they will send it back to you with the log that contains the error messages. You must fix the errors and then repeat the process of testing the package in the staging instance of your tenant, exporting the package to the UAT instance of your tenant, and requesting a new migration from Support.
- If there are any problems with the data migration, Support will restore the database from a backup. Report any issues quickly to minimize losing any new transactions because of the restoration from a backup.

9. Web Services

This document describes the Ivanti Service Manager Web Services API, which can be used for integrating your application with Ivanti Service Manager.

By the end of this document, you should be able to create an API for your application that can work with Ivanti Service Manager Web Services.

9.1. Request Offerings and Service Requests

Various Web methods have been added to the FRSHEATIntegration API, to provide access to the Request Offerings available in the Service Catalog, as well as providing the ability to submit and Service Requests, and retrieve them once they are submitted.

The following table summarizes the available Web Methods for this:

Web Method	Description
GetCategories	Get the list of available Categories for the Service Catalog
GetCategoryTemplates	Get the list of Request Offerings in the Service Catalog, which belong to the indicated Category
GetAllTemplates	Get the full list of Request Offerings in the Service Catalog
GetSubscriptionId	Fetches the Subscription ID, corresponding to the Request Offering for the current user
GetPackageData	Retrieve the details of the indicated Request Offering
UserCanAccessRequestOffering	Check whether the current user is entitled to access the given Request Offering
SubmitRequest	Creates and submits the Service Request, on behalf of a given user
GetRequestData	Retrieves the data for the given Service Request
FetchServiceReqValidationListData	Fetch the allowable validation list data for the given Service Request parameter

9.2. Searching for Records

The FRS HEAT Integration Web Service provides several means to search for records in a given tenant. The following table summarizes the four Web Methods providing for searching records:

WebMethod	Applicability
Search	General purpose method for searching for records, using an arbitrary query criteria. Use this WebMethod, if the other three available convenience WebMethods mentioned below, cannot be used to express the desired query criteria.
FindBusinessObject	Search for the business object record, by means of its RecId field value in the database. Since records are uniquely identified by the RecId (i.e. the primary key), this Web Method will return exactly one record, assuming the provided RecId does match one of the existing records in the business object. If the RecId of the record to search for is not readily available, then the other WebMethods should alternatively be used.
FindSingleBusinessObjectByField	Search for the business object record, by means of the provided field / value criteria, and return the exact record match, if it can be found. For example, this WebMethod can be used to identify a Profile.Employee record, by means of either the LoginId or PrimaryEmail field (assuming the field values for the column is unique in the database table). As the name suggests, this Web Method will return exactly one matching record, if it can be located. Note that if the search returns multiple records, it will not return any results. If you are unsure of whether multiple results will be returned, use either the FindMultipleBusinessObjectsByField or FindBusinessObject Web Methods, as an alternative.
FindMultipleBusinessObjectsByField	Search for the business object record, by means of the provided field / value criteria, and return all of the results in an array. For example, this WebMethod can be used to identify all Incident records with a Status of “Resolved”. As the name suggests, this Web Method will return one or more matching results via an array. If it is known beforehand that the search criteria will return exactly one record, the FindSingleBusinessObjectByField Web Method provides a more convenient way of accessing the record directly. Otherwise the FindMultipleBusinessObjecsByFieldWebMethod can still be used, where the record can be retrieved via the first item in the array.

From the above table, it can be seen that the **FindBusinessObject**, **FindSingleBusinessObjectByField**, and **FindMultipleBusinessObjectsByField** Web Methods are all special cases of the Search WebMethod – the latter Web Method can be used to express any arbitrarily complex query.

The following sections will describe the four query Web Methods in further detail.

FindBusinessObject

Retrieves a single business object using its primary identifier (RecId field in the database)

Request syntax:

FRSHEATIntegrationFindBOResponse FindBusinessObject(string sessionKey, string tenantId, string boType, string recId)

Parameters:

- **sessionKey**: Key received in the earlier Connect request
- **tenantId**: tenant for which the key is authenticated.
- **boType**: type of the business object to retrieve, for example Incident or Change.
- **recId**: unique identifier for the object

Return Value:

An FRSHEATIntegrationFindBOResponse object, defined as follows:

```
public class FRSHEATIntegrationFindBOResponse
{
    public string status { get; set; }
    public string exceptionReason { get; set; }
    public WebServiceBusinessObject obj { get; set; }
}
```

The **FRSHEATIntegrationFindBOResponse** class comprises the following fields:

- **status** – this field provides a Status value indicating whether the operation was successful.
  A full description of the available Status values is provided in the table below.
- **exceptionReason** – if there is an exception thrown in the course of running the Connect WebMethod, the exception information will be captured in this field.
- **obj** – if the exact record can be found via the FindBusinessObject WebMethod call (i.e. the value of the status field is "Success"), the business object record can be accessed via this field

The following table lists the available status values, and describes how to interpret them.

Status	Explanation
Success	The business object can be successfully found – access the record via the obj field in the response object.
Error	The business object cannot be successfully found – the obj field will be null, and the exception will be stored in the exceptionReason field. The most typical error is the Table not found exception, which occurs when the specified business object does not exist in the tenant. Double-check to make sure that the name of the business object is spelled properly (e.g. “Incident”, “Profile.Employee”, etc.)
NotFound	The specified business object does exist in the tenant, but the provided RecID value does not match any of the existing records in the object. Since this is not an exceptional condition, there will not be an exception stored in the exceptionReason field, and the obj field will be null. Double-check to make sure the RecID field for the intended record does in fact exist in the tenant. An alternate query Web Method (e.g. “FindSingleBusinessObjectByField”) might be alternatively used for retrieving the record.

Example

```
FRSHEATIntegrationFindBOResponse res = frSvc.FindBusinessObject(authSessionKey, tenantId, "Incident",
"A981FBEBAA8B4EE2820364505855ABC2");

if (res.status == "Success")

{

foreach (WebServiceFieldValue f in res.obj.FieldValues)

{

if (string.Compare(f.Name, "LastModDateTime", true) == 0)

{

DateTime lastMod;

if (f.Value != null)

{

lastMod = (DateTime)f.Value;

Console.WriteLine("The LastModDateTime of the record is " + lastMod.ToString());

}

}

}

}
```

FindSingleBusinessObjectByField

This Web Method retrieves a single business object, by means of the specified field / value criteria.

This is a convenience Web Method introduced for searching for a matching record, by means of a unique field.

Here are some common use cases, where this query Web Method can come in handy:

- Search for a specific Profile.Employee record, by means of the LoginID field
- Search for a specific Profile.Employee record, by means of the PrimaryEmail field
- Search for a specific StandardUserTeam record, by means of the Team field
- Search for a specific OrganizationalUnit record, by means of the Name field

Request syntax:

FRSHEATIntegrationFindBOResponse FindSingleBusinessObjectByField(string sessionKey, string tenantId, string boType, string fieldName, string fieldValue)

Parameters:

- **sessionKey**: Key received in the earlier Connect request
- **tenantId**: tenant for which the key is authenticated.
- **boType**: type of the business object to retrieve, for example Incident or Change.
- **recId**: unique identifier for the object
- **fieldName**: the name of the field in the business object, to search against (e.g. “Status”)
- **fieldValue**: the value for the field to search for in the matching record (e.g. “Active”)

Return Value:

An FRSHEATIntegrationFindBOResponse object, defined as follows:

```
public class FRSHEATIntegrationFindBOResponse
{
    public string status { get; set; }
    public string exceptionReason { get; set; }
    public WebServiceBusinessObject obj { get; set; }
}
```

The FRSHEATIntegrationFindBOResponse class comprises the following fields:

- **status** – this field provides a Status value indicating whether the operation was successful.
  A full description of the available Status values is provided in the table below.
- **exceptionReason** – if there is an exception thrown in the course of running the Connect WebMethod, the exception information will be captured in this field.
- **obj** – if the exact record can be found via the FindBusinessObject WebMethod call (i.e. the value of the status field is “Success”), the business object record can be accessed via this field

The following table lists the available status values, and describes how to interpret them.

Status	Explanation
Success	The business object can be successfully found – access the record via the obj field in the response object.
Error	The business object cannot be successfully found – the obj field will be null, and the exception will be stored in the exceptionReason field. One typical error is the Table not found exception, which occurs when the specified business object does not exist in the tenant. Double-check to make sure that the name of the business object is spelled properly (e.g. “Incident”, “Profile.Employee”, etc.) The other common error encountered, is when the specified field does not exist for the business object – here, the error message would be of the form: ObjectTableMap: field <FieldName> is not found in table <Business Object># Double-check to make sure that the field name is spelled correctly, and is actually defined for the given business object.
NotFound	The specified business object does exist in the tenant, but the provided RecID value does not match any of the existing records in the object. Since this is not an exceptional condition, there will not be an exception stored in the exceptionReason field, and the obj field will be null. Double-check to make sure the fieldValue field for the intended record does in fact exist in the tenant. An alternate query Web Method (e.g. “FindSingleBusinessObjectByField”) might be used for retrieving the record.
Multiple Results	The provided search criteria returned more than one matching result. Since the intent of this WebMethod is to return a single matching business object, the obj field in the response object will remain null, whenever the Status of the response object is “MultipleResults”. If there is a possibility for returning multiple matching results, the “FindSingleBusinessObjectByField” Web Method should be used instead.

Example

```
FRSHEATIntegrationFindBOResponse res = frSvc.FindSingleBusinessObjectByField(authSessionKey, tenantId, "Incident", "IncidentNumber", "10001");

if (res.status == "Success")

{

foreach (WebServiceFieldValue f in res.obj.FieldValues)
```

```
{
if (string.Compare(f.Name, "LastModDateTime", true) == 0)
{
DateTime lastMod;
if (f.Value != null)
{
lastMod = (DateTime)f.Value;
Console.WriteLine("The LastModDateTime of the record is " + lastMod.ToString());
}
}
}
}
```

FindMultipleBusinessObjectsByField

This Web Method retrieves one or more business objects, by means of the specified field / value criteria, and returns the result as an array of business objects.

For example, this Web Method can be used to retrieve all Incident records with Status of “Active”, all Changes with Type of “Major”, etc.

This Web Method only allows for searches based on a single field / value criteria – if more complex queries need to be expressed, the Search Web Method should be used instead.

Also, this Web Method will always return the results in an array, even if there is exactly one matching record returned. In that case, the FindSingleBusinessObjectByField Web Method might be more convenient to use, if the desired query criteria will return exactly one record.

Request syntax:

Bash

```
FRSHEATIntegrationSearchResponse FindMultipleBusinessObjectsByField(string sessionKey, string tenantId, string boType, string fiel
```

Parameters:

- **sessionKey**: Key received in the earlier Connect request
- **tenantId**: tenant for which the key is authenticated.
- **boType**: type of the business object to retrieve, for example Incident or Change.
- **recId**: unique identifier for the object
- **fieldName**: the name of the field in the business object, to search against (e.g. “Status”)
- **fieldValue**: the value for the field to search for the matching record (e.g. “Active”)

Return Value:

A FRSHEATIntegrationSearchResponse object, defined as follows:

```
public class FRSHEATIntegrationSearchResponse
{
    public string status { get; set; }
    public string exceptionReason { get; set; }
    public List<List<WebServiceBusinessObject>> objList { get; set; }
}
```

The FRSHEATIntegrationSearchResponse class comprises the following fields:

- **status** – this field provides a Status value indicating whether the operation was successful.
  A full description of the available Status values is provided in the table below.
- **exceptionReason** – if there is an exception thrown in the course of running the Connect WebMethod, the exception information will be captured in this field.
- **objList** – if one or more records can be found via the WebMethod call (i.e. the value of the status field is “Success”), the results will be returned via this field, which is a List of Lists of WebServiceBusinessObject objects.
  The outer list contains multiple business objects, if the search condition matched more than one object.
  The inner list contains joined business objects if the search condition requested joins.
  Unlike SQL response fields from each joined objects are kept in a separate list, they are not mingled together.

The following table lists the available status values, and describes how to interpret them.

Status	Explanation
Success	The business objects can be successfully found – access the matching records via the objList field in the response object, which returns the results as a List of List of WebServiceBusinessObjects.
Error	The business objects cannot be successfully found – the objList field will be null, and the exception will be stored in the exceptionReason field. One typical error is the Table not found exception, which occurs when the specified business object does not exist in the tenant. Double-check to make sure that the name of the business object is spelled properly (e.g. “Incident”, “Profile.Employee”, etc.) The other common error encountered, is when the specified field does not exist for the business object – here, the error message would be of the form: ObjectTableMap: field <FieldName> is not found in table <Business Object># Double-check to make sure that the field name is spelled correctly, and is actually defined for the given business object.
NotFound	The specified business object does exist in the tenant, but the provided RecID value does not match any of the existing records in the object. Since this is not an exceptional condition, there will not be an exception stored in the exceptionReason field, and the objList field will be null. Double-check to make sure the fieldValue for the intended records does in fact exist in the tenant.

Example

```
FRSHEATIntegrationSearchResponse res = frSvc.FindMultipleBusinessObjectsByField(authSessionKey, tenantId, "Incident", "Status", "Active");

if (res.status == "Success")

{

WebServiceBusinessObject[][] incidentList = res.objList;

foreach (WebServiceBusinessObject[] incidentOuterList in incidentList)

{

foreach (WebServiceBusinessObject incident in incidentOuterList)

{

WebServiceFieldValue[] incidentFieldList = incident.FieldValues;

WebServiceFieldValue incidentNumberField = incidentFieldList.SingleOrDefault(f ⇒ f.Name == "IncidentNumber");

Console.WriteLine("Incident  matches the selection criteria", incidentNumberField.Value);

}
```

```
}

}
```

Search

This Web Method retrieves one or more business objects satisfying the search criteria. This is an SQL-style query.

Compared to the earlier three query Web Methods, this is a general purpose Web Method which can be used to express arbitrarily complex queries.

Request syntax:

FRSHEATIntegrationSearchResponse Search(string sessionKey, string tenantId, ObjectQueryDefinition query)

Parameters:

- **sessionKey**: Key received in the earlier Connect request
- **tenantId**: Tenant for which the key is authenticated.
- **query**: A structure describing the search criteria. It follows the structure of a SQL SELECT request and captures most of the possible parameters in SELECT queries, including TOP, WHERE, JOIN, ORDER BY clauses.

Return Value:

An FRSHEATIntegrationSearchResponse object, defined as follows:

```
public class FRSHEATIntegrationSearchResponse

{

    public string status { get; set; }

    public string exceptionReason { get; set; }

    public List<List<WebServiceBusinessObject>> objList { get; set; }

}
```

The FRSHEATIntegrationSearchResponse class comprises the following fields:

- **status** – this field provides a Status value indicating whether the operation was successful.
  A full description of the available Status values is provided in the table below.
- **exceptionReason** – if there is an exception thrown in the course of running the Connect WebMethod, the exception information will be captured in this field.
- **objList** – if one or more records can be found via the WebMethod call (i.e. the value of the status field is "Success"), the results will be returned via this field, which is a List of Lists of WebServiceBusinessObject objects.
  The outer list contains multiple business objects, if the search condition matched more than one object.
  The inner list contains joined business objects if the search condition requested joins.
  Unlike SQL response fields from each joined objects are kept in a separate list, they are not mingled together.

The following table lists the available status values, and describes how to interpret them.

Status	Explanation
Success	The business objects can be successfully found – access the matching records via the objList field in the response object, which returns the results as a List of List of WebServiceBusinessObjects.
Error	The business objects cannot be successfully found – the objList field will be null, and the exception will be stored in the exceptionReason field. One typical error is the Table not found exception, which occurs when the specified business object does not exist in the tenant. Double-check to make sure that the name of the business object is spelled properly (e.g. “Incident”, “Profile.Employee”, etc.) The other common error encountered, is when the specified field does not exist for the business object – here, the error message would be of the form: ObjectTableMap: field <FieldName> is not found in table <Business Object># Double-check to make sure that the field name is spelled correctly, and is actually defined for the given business object.
NotFound	The specified business object does exist in the tenant, but the provided RecID value does not match any of the existing records in the object. Since this is not an exceptional condition, there will not be an exception stored in the exceptionReason field, and the objList field will be null. Double-check to make sure the fieldValue for the intended records does in fact exist in the tenant.

Example:

The following example will search for Incident records where the Priority is equal to 1, and the Status is equal to “Active”, and retrieves the corresponding IncidentNumber values from the matching results.

```
ObjectQueryDefinition query = new ObjectQueryDefinition();

query.Select = new SelectClass();

// Retrieve just the IncidentNumber field value from the Incident,

// when invoking the search

FieldClass[] incidentFieldObjects = new FieldClass[] {

new FieldClass()

    {

Name = "IncidentNumber",

Type = "Text"

}

};
```

```
query.Select.Fields = incidentFieldObjects;
query.From = new FromClass();
query.From.Object = "Incident";
query.Where = new RuleClass[] {
new RuleClass()
    {
    Join = "AND",
        Condition = "=",
        Field = "Priority",
        Value = "1"
},
    new RuleClass()
    {
    Join = "AND",
        Condition = "=",
        Field = "Status",
        Value = "Active"
}
};
FRSHEATIntegrationSearchResponse searchResponse = frSvc.Search(authSessionKey, tenantId, query);
if (searchResponse.status == "Success")
{
WebServiceBusinessObject[][] incidentList = searchResponse.objList;
    foreach (WebServiceBusinessObject[] incidentOuterList in incidentList)
    {
    foreach (WebServiceBusinessObject incident in incidentOuterList)
        {
        // Since we are just retrieving one field in the selection criteria
            // (i.e. IncidentNumber), this corresponds to
            // incident.FieldValues[0].Value when retrieving the results
            Console.WriteLine("Incident  matches the selection criteria", incident.FieldValues[0].Value);
```

```
        }
    }
}
```

9.3. Understanding the Search Results

As mentioned above, the return value of the Search Web Method is a list of list of WebServiceBusinessobject objects (i.e. a two dimensional array).

The previous section provided a simple search example, where the records are returned against a single business object (e.g. “Incident”).

When performing searches against the current object and related objects, it is important to note that each item in the two-dimensional array represents the matching parent - child combination.

The best way to understand the format of the search results is to consider the following four search scenarios, with the above point in mind:

Scenario 1: Perform a search against a single business object, which returns an exact match

Assume the search is defined against a single business object (e.g. Incident), and the search result returns 1 exact record.

Here, `objList[0][0]` contains the matching Incident record.

Consider the following code sample:

```
ObjectQueryDefinition query = new ObjectQueryDefinition();
FieldClass[] fieldObjects = new FieldClass[] {
   new FieldClass()
   {
      Name = "IncidentNumber",
      Type = "Text"
   },
   new FieldClass()
   {
      Name = "Status",
      Type = "Text"
   }
};
query.Select = new SelectClass();
query.Select.Fields = fieldObjects;
query.From = new FromClass();
query.From.Object = "Incident";
query.Where = new RuleClass[] {
```

```
            new RuleClass()
            {
                Field = "IncidentNumber",
                Condition = "=",
                Value = "10001"
            }
        };
        FRSHEATIntegrationSearchResponse searchResponse = frSvc.Search(authSessionKey, tenantId, query);
        if (searchResponse.status == "Success")
        {
            WebServiceBusinessObject[][] objList = searchResponse.objList;
            foreach (WebServiceBusinessObject[] objOuterList in objList)
            {
                foreach (WebServiceBusinessObject obj in objOuterList)
                {
                    WebServiceFieldValue[] objFieldList = obj.FieldValues;
                    Console.WriteLine(" with  \"\" matches the selection criteria", obj.BusinessObjectName, objFieldList[0].Name,
objFieldList[0].Value);
                }
            }
        }
    }
```

Here we are searching specifically for Incident 10001 in the tenant. Assuming the record exists in the tenant, it will be available via objList[0][0].



Scenario 2: Perform a search against a single business object, which returns several matches

Assume the search is defined against a single business object (e.g. “Incident”), and the search result returns n matching records.

In this case, the matching records can be accessed from objList[0][0] through objList[n-1][0].

For example, assume the search returns 10 Incident records – the Incident records can be accessed from objList[0][0] through objList[9][0].

Here, the first index in the two-dimensional array changes, but the second index remains at 0.

Consider the following code sample:

```
ObjectQueryDefinition query = new ObjectQueryDefinition();
FieldClass[] fieldObjects = new FieldClass[] {
   new FieldClass()
   {
      Name = "IncidentNumber",
      Type = "Text"
   },
   new FieldClass()
   {
      Name = "Status",
      Type = "Text"
   }
};
query.Select = new SelectClass();
query.Select.Fields = fieldObjects;
query.From = new FromClass();
query.From.Object = "Incident";
query.Where = new RuleClass[] {
   new RuleClass()
   {
      Field = "IncidentNumber",
      Condition = "<=",
      Value = "10010"
   }
};
query.OrderBy = new OrderByClass[] {
   new OrderByClass()
   {
      Name = "IncidentNumber",
```

```
            Direction = "ASC"
        }
    };
    FRSHEATIntegrationSearchResponse searchResponse = frSvc.Search(authSessionKey, tenantId, query);
    if (searchResponse.status == "Success")
    {
        WebServiceBusinessObject[][] objList = searchResponse.objList;
        foreach (WebServiceBusinessObject[] objOuterList in objList)
        {
            foreach (WebServiceBusinessObject obj in objOuterList)
            {
                WebServiceFieldValue[] objFieldList = obj.FieldValues;
                Console.WriteLine(" with  \"\" matches the selection criteria", obj.BusinessObjectName, objFieldList[0].Name,
objFieldList[0].Value);
            }
        }
    }
}
```

Compared to the code from scenario 1, only the following has been updated:

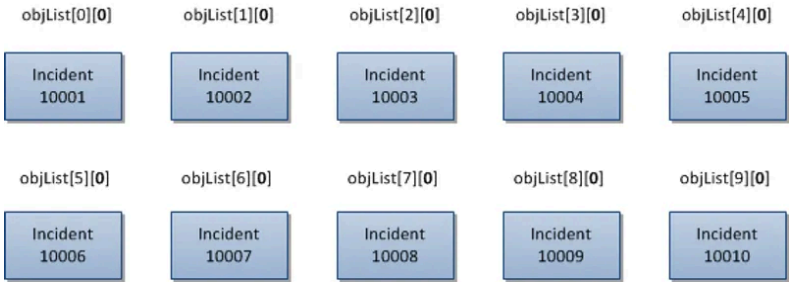
```
    query.Where = new RuleClass[] {
        new RuleClass()
        {
            Field = "IncidentNumber",
            Condition = "<=",
            Value = "10010"
        }
    };
```

Here, we are searching for Incident records where the IncidentNumber is less than or equal to 10010.

Assuming that the tenant has Incident records starting from 10001 through 10010, Incident 10001 can be accessed via objList[0][0], Incident 10002 can be accessed via objList[1][0], up to Incident 10010 which can be accessed via objList[9][0].

The following diagram illustrates how each of the items in the collection can be accessed individually, via the two dimensional array of search results.

Since there are no child objects to be searched against, all of the items can be accessed using objList[n][0].



Scenario 3: Perform a search against a single business object and its related child objects, which returns one exact match for the parent object

Assume we are searching for Incidents with matching Journal.Email records.

Assume that the search returns with one matching Incident record, which contains four related Journal.Email records.

When performing searches against the current object and related objects, it is important to note that each item in the two-dimensional array represents the matching parent / child combination.

In this scenario, since there is only one matching parent Incident record, the search results will contain the same parent record four times, one for each matching Journal.Email record.

So the parent Incident record can be accessed using either objList[0][0], objList[1][0], objList[2][0], or objList[3][0]. Here, the first index value varies, and the second index value will be 0 (to denote the main object).

Each individual matching Journal.Email child record can be accessed using objList[0][1], objList[1][1], objList[2][1], and objList[3][1]. Here, the first index value varies (to correspond to each matching Journal.Email record), and the second index will be 1 (to denote the first child object).

Consider the following code sample:

```
ObjectQueryDefinition query = new ObjectQueryDefinition();

FieldClass[] fieldObjects = new FieldClass[] {
   new FieldClass()
   {
      Name = "IncidentNumber",
      Type = "Text"
   },
   new FieldClass()
   {
      Name = "Journal.Subject",
      Type = "Text"
   }
};
```

```
query.Select = new SelectClass();
query.Select.Fields = fieldObjects;
query.From = new FromClass();
query.From.Object = "Incident";
query.From.Links = new FromLinkClass[] {
    new FromLinkClass {
        Relation = "",
        Object = "Journal#Email"
    }
};
query.Where = new RuleClass[] {
    new RuleClass()
    {
        Field = "IncidentNumber",
        Condition = "=",
        Value = "10001"
    },
    new RuleClass()
    {
        Field = "Journal.Category",
        Condition = "=",
        Value = "Incoming Email"
    },
};
query.OrderBy = new OrderByClass[] {
    new OrderByClass()
    {
        Name = "IncidentNumber",
        Direction = "ASC"
    },
    new OrderByClass()
```

```
        {
            Name = "Journal.Subject",
            Direction = "ASC"
        }
    };
    FRSHEATIntegrationSearchResponse searchResponse = frSvc.Search(authSessionKey, tenantId, query);
    if (searchResponse.status == "Success")
    {
        WebServiceBusinessObject[][] objList = searchResponse.objList;
        foreach (WebServiceBusinessObject[] objOuterList in objList)
        {
            foreach (WebServiceBusinessObject obj in objOuterList)
            {
                WebServiceFieldValue[] objFieldList = obj.FieldValues;
                Console.WriteLine(" with   matches the selection criteria", obj.BusinessObjectName, objFieldList[0].Name,
objFieldList[0].Value);
            }
        }
    }
```

Here, we are searching for Incident records where the IncidentNumber is equal to 10001, and has associated Journal.Email records with Category of “Incoming Email” (i.e. emails attached to the Incident record, via the email listener).

Because the search needs to consider not only the top level object (i.e. “Incident”), but also the related “Journal.Email” records, additional lines of code needs to be written.

First off, the FromLinkClass needs to be specified:

```
    query.From.Links = new FromLinkClass[] {
        new FromLinkClass {
            Relation = "",
            Object = "Journal#Email"
        }
    };
```

Here, we create a new instance of the FromLinkClass, which designates the relationship from Incident to Journal.Email.

The existing “IncidentContainsJournal” relationship (which points to the Journal base object) can be used to associate the Incident with the child Journal records.

Because the relationship has no value specified for the internal reference name parameter, the Relation value is left as an empty string, in the Relation member above.

The actual object to be searched against (in this case, Journal.Email) needs to be specified above in the Object member.

With the Links property specified for the FromClass, the fields from the related business object can be accessed. For example, besides accessing the IncidentNumber field of the Incident, the Subject of the Journal.Email can be accessed:

```
FieldClass[] fieldObjects = new FieldClass[] {
    new FieldClass()
    {
        Name = "IncidentNumber",
        Type = "Text"
    },
    new FieldClass()
    {
        Name = "Journal.Subject",
        Type = "Text"
    }
};
```

Notice that to reference the Subject field in the Journal.Email business object, the field name needs to be specified as “Journal.Subject”.

Similarly, to specify the rule condition using fields in the Journal.Email business object (e.g. “Category”), the field name needs to be specified as “Journal.Category”:

```
query.Where = new RuleClass[] {
    new RuleClass()
    {
        Field = "IncidentNumber",
        Condition = "=",
        Value = "10001"
    },
    new RuleClass()
    {
        Field = "Journal.Category",
```

```
            Condition = "=",
            Value = "Incoming Email"
        },
    };
```

So in the example scenario, assume that Incident 10001 exists in the tenant, with four Journal.Emails attached to it, with subject values of “Email 1”, “Email 2”, “Email 3”, and “Email 4”.

Running the above sample code yields the following results in the console window:

Incident with IncidentNumber "10001" matches the selection criteria

Journal.Email with Journal.Subject "Email 1" matches the selection criteria

Incident with IncidentNumber "10001" matches the selection criteria

Journal.Email with Journal.Subject "Email 2" matches the selection criteria

Incident with IncidentNumber "10001" matches the selection criteria

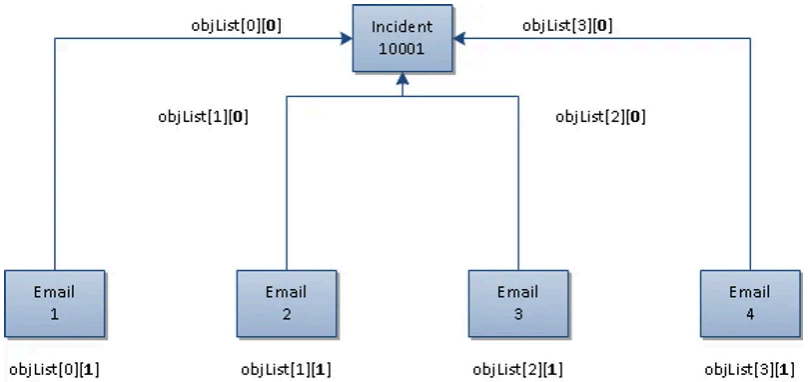
Journal.Email with Journal.Subject "Email 3" matches the selection criteria

Incident with IncidentNumber "10001" matches the selection criteria

Journal.Email with Journal.Subject "Email 4" matches the selection criteria

As mentioned in the previous sections, the results are returned for each parent / child combination. Because the four Journal.Email records are associated with the same Incident record (Incident 10001), this same Incident record will show up four times, once for each Journal.Email child record associated with it.

The following diagram illustrates how each of the items in the parent / child combination can be accessed individually, via the two dimensional array of search results.



Scenario 4: Perform a search against a single business object and its related child objects, which returns several matches for the parent object

Assume we are searching for Incidents with matching Journal.Email records.

Assume that the search returns with two matching Incident records, which contains six related Journal.Email records – four for the first Incident record (e.g. Incident 10001), and two for the second Incident record (e.g. Incident 10008).

As before, recall that when performing searches against the current object and related objects, it is important to note that each item in the two-dimensional array represents the matching parent / child combination.

In this scenario, since there are two matching parent Incident records, and six related Journal.Email records, the search results will return six results.

The first Incident record will show up using either objList[0][0], objList[1][0], objList[2][0], and objList[3][0], and its corresponding Journal.Email records will show up via objList[0][1], objList[1][1], objList[2][1], and objList[3][1].

The second Incident record will show up using either objList[4][0] and objList[5][0], and its corresponding Journal.Email records will show up via objList[4][1] and objList[5][1].

Consider the following code sample:

```
ObjectQueryDefinition query = new ObjectQueryDefinition();
FieldClass[] fieldObjects = new FieldClass[] {
   new FieldClass()
   {
      Name = "IncidentNumber",
      Type = "Text"
   },
   new FieldClass()
   {
      Name = "Journal.Subject",
      Type = "Text"
   }
};
query.Select = new SelectClass();
query.Select.Fields = fieldObjects;
query.From = new FromClass();
query.From.Object = "Incident";
query.From.Links = new FromLinkClass[] {
   new FromLinkClass {
      Relation = "",
      Object = "Journal#Email"
   }
};
```

```
query.Where = new RuleClass[] {
    new RuleClass()
    {
        Field = "IncidentNumber",
        Condition = "<=",
        Value = "10010"
    },
    new RuleClass()
    {
        Field = "Journal.Category",
        Condition = "=",
        Value = "Incoming Email"
    },
};
query.OrderBy = new OrderByClass[] {
    new OrderByClass()
    {
        Name = "IncidentNumber",
        Direction = "ASC"
    },
    new OrderByClass()
    {
        Name = "Journal.Subject",
        Direction = "ASC"
    }
};
FRSHEATIntegrationSearchResponse searchResponse = frSvc.Search(authSessionKey, tenantId, query);
if (searchResponse.status == "Success")
{
    WebServiceBusinessObject[][] objList = searchResponse.objList;
    foreach (WebServiceBusinessObject[] objOuterList in objList)
```

```
        {
            foreach (WebServiceBusinessObject obj in objOuterList)
            {
                WebServiceFieldValue[] objFieldList = obj.FieldValues;
                Console.WriteLine(" with   matches the selection criteria", obj.BusinessObjectName, objFieldList[0].Name,
objFieldList[0].Value);
            }
        }
    }
```

Bash

```
Compared to the code from scenario 3, only the following has been updated:
```

```
    query.Where = new RuleClass[] {
        new RuleClass()
        {
            Field = "IncidentNumber",
            Condition = "<=",
            Value = "10010"
        }
    };
```

Here, we are searching for Incident records where the IncidentNumber is less than or equal to 10010, and has associated Journal.Email records with Category of “Incoming Email” (i.e. emails attached to the Incident record, via the email listener).

So in the example scenario, assume that Incident 10001 exists in the tenant, with four Journal.Emails attached to it, with subject values of “Email 1”, “Email 2”, “Email 3”, and “Email 4”.

Assume that Incident 10008 also exists in the tenant, with two Journal.Email records attached to it, with subject values of “Email 5” and “Email 6”.

Running the above sample code yields the following results in the console window:

Incident with IncidentNumber "10001" matches the selection criteria

Journal.Email with Journal.Subject "Email 1" matches the selection criteria

Incident with IncidentNumber "10001" matches the selection criteria

Journal.Email with Journal.Subject "Email 2" matches the selection criteria

Incident with IncidentNumber "10001" matches the selection criteria

Journal.Email with Journal.Subject "Email 3" matches the selection criteria

Incident with IncidentNumber "10001" matches the selection criteria

Journal.Email with Journal.Subject "Email 4" matches the selection criteria

Incident with IncidentNumber "10008" matches the selection criteria

Journal.Email with Journal.Subject "Email 5" matches the selection criteria

Incident with IncidentNumber "10008" matches the selection criteria

Journal.Email with Journal.Subject "Email 6" matches the selection criteria

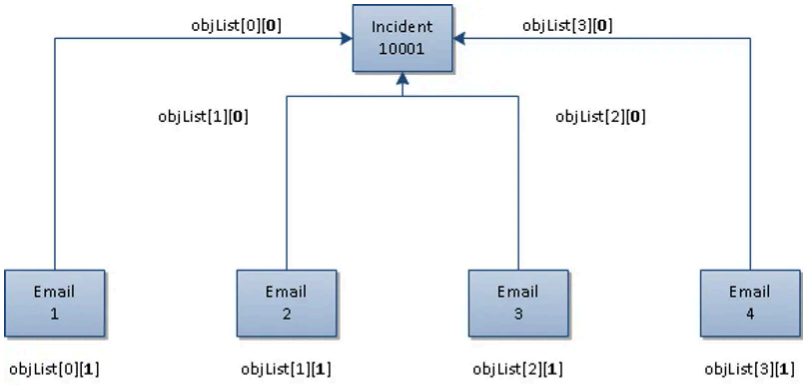
As mentioned in the previous sections, the results are returned for each parent / child combination.

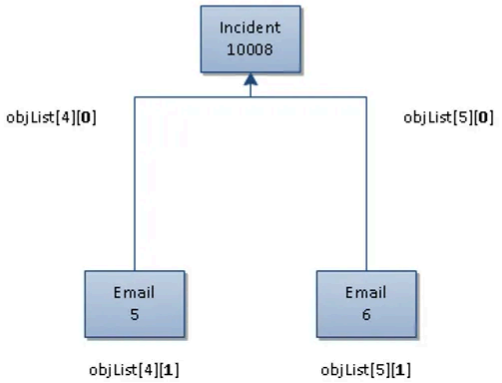
Because there are four Journal.Email records associated with Incident 10001, this same Incident record will show up four times, once for each of the four Journal.Email child records associated with it.

Afterwards, there are two Journal.Email records associated with Incident 10008, so this same Incident will show up two times, once for each of the two Journal.Email child records associated with it.

In total, there are six such parent / child combinations, so there are twelve object records returned (six for the Incident, and six for the distinct Journal.Email records).

The following diagram illustrates how each of the items in the parent / child combination can be accessed individually, via the two dimensional array of search results.





Grouping the Rule Criteria

The previous section describes how to search for records based on the current business object (e.g. “Incident”) and its related child business objects (e.g. “Journal.Email”).

The earlier examples describe how to formulate queries such as the following:

Retrieve all Incident records with IncidentNumber less than 10010, containing related Journal.Email records with Category of “Incoming Email”

The Search WebMethod is actually flexible enough to express searches through the grouping of the rule criteria.

For example, the Search WebMethod allows one to express queries such as the following:

Retrieve all Incident records with IncidentNumber less than 10010, containing related Journal.Email records where

(The Category of the Journal.Email is “Incoming Email”

OR

The Subject of the Journal.Email is “Urgent Request”)

The above search allows the user to search for Incidents containing Journal.Email records, either if the Journal.Email has a Category of “Incoming Email” (i.e. it was created via the Email Listener), OR the Subject of the Journal.Email has a subject line of “Urgent Request” regardless if the email has a Category of “Incoming Email” or “Outgoing Email”.

To express the above search, consider the following code sample:

```
ObjectQueryDefinition query = new ObjectQueryDefinition();

FieldClass[] fieldObjects = new FieldClass[] {

   new FieldClass()

   {

      Name = "IncidentNumber",

      Type = "Text"

   },

   new FieldClass()

   {

      Name = "Journal.Category",

      Type = "Text"

   }

};

query.Select = new SelectClass();

query.Select.Fields = fieldObjects;

query.From = new FromClass();

query.From.Object = "Incident";
```

```
query.From.Links = new FromLinkClass[] {
    new FromLinkClass
    {
        Relation = "",
        Object = "Journal#Email"
    }
};
query.Where = new RuleClass[] {
    new RuleClass
    {
        Join = "AND",
        Field = "IncidentNumber",
        Condition = "<=",
        Value = "10010"
    },
    new RuleClass
    {
        Rules = new RuleClass[] {
            new RuleClass
            {
                Field = "Journal.Category",
                Condition = "=",
                Value = "Outgoing Email"
            },
            new RuleClass
            {
                Join = "OR",
                Field = "Journal.Subject",
                Condition = "=",
                Value = "Urgent Request"
            }
```

```
            }
        }
    };
    FRSHEATIntegrationSearchResponse searchResponse = frSvc.Search(authSessionKey, tenantId, query);
    if (searchResponse.status == "Success")
    {
        WebServiceBusinessObject[][] objList = searchResponse.objList;
        foreach (WebServiceBusinessObject[] objOuterList in objList)
        {
            foreach (WebServiceBusinessObject obj in objOuterList)
            {
                WebServiceFieldValue[] objFieldList = obj.FieldValues;
                Console.WriteLine(" with  \"\" matches the selection criteria", obj.BusinessObjectName, objFieldList[0].Name,
objFieldList[0].Value);
            }
        }
    }
```

This code sample is a variation of the samples from the earlier section, with several important differences.

In particular, notice the update to the Where property for the ObjectQueryDefinition object:

```
    query.Where = new RuleClass[] {
        new RuleClass
        {
            Join = "AND",
            Field = "IncidentNumber",
            Condition = "<=",
            Value = "10010"
        },
        new RuleClass
        {
            Rules = new RuleClass[] {
                new RuleClass
```

```
            {
                Field = "Journal.Category",
                Condition = "=",
                Value = "Outgoing Email"
            },
            new RuleClass
            {
                Join = "OR",
                Field = "Journal.Subject",
                Condition = "=",
                Value = "Urgent Request"
            }
        }
    }
};
```

At the top level are two RuleClass objects – one for expression the condition to search for Incident records with IncidentNumber less than or equal to 10010, as illustrated in the previous sections:

```
        new RuleClass
        {
            Join = "AND",
            Field = "IncidentNumber",
            Condition = "<=",
            Value = "10010"
        },
```

The second RuleClass is used solely to populate the Rules member variable of the RuleClass:

```
        new RuleClass
        {
            Rules = new RuleClass[] {
                ...
                ...
                ...
```

```
        }
    }
```

So inside the second RuleClass, another RuleClass array is being instantiated with two inner RuleClasses:

```
        new RuleClass
        {
            Field = "Journal.Category",
            Condition = "=",
            Value = "Outgoing Email"
        },
        new RuleClass
        {
            Join = "OR",
            Field = "Journal.Subject",
            Condition = "=",
            Value = "Urgent Request"
        }
```

So the inner RuleClass array essentially allows one to express the following portion of the search:

Retrieve the related Journal.Email records where either

(The Category of the Journal.Email is “Incoming Email”

OR

The Subject of the Journal.Email is “Urgent Request”)

To express the following related query:

Retrieve the related Journal.Email records where either

(The Category of the Journal.Email is “Incoming Email”

AND

The Subject of the Journal.Email is “Urgent Request”)

the Join member variable needs to be changed from “OR” to “AND”, as follows:

```
        new RuleClass
        {
            Field = "Journal.Category",
            Condition = "=",
```

```
        Value = "Outgoing Email"
    },
    new RuleClass
    {
        Join = "AND",
        Field = "Journal.Subject",
        Condition = "=",
        Value = "Urgent Request"
    }
```

With the above example, it can be seen how Rules can be grouped together, by populating the Rules member variable of the RuleClass object.

That is, the RuleClass uses composition to allow RuleClasses to be arbitrarily grouped together, using AND or OR operators via the Join property.

Full Text Searching

Besides regular SQL-style searches, the Search WebMethod also supports performing full text searches against a business object (e.g. search for records containing the terms “Email Down”, against the full text catalog of the Incident object).

The RuleClass contains a member called ConditionType, which is of type SearchConditionType - an enumeration with two permissible values:

- ByField - 0 (regular SQL search)
- ByText - 1 (full text SQL search)

```
public enum SearchConditionType
{
    ByField = 0,
    ByText = 1
}
```

Regular searches are performed by setting SearchConditionType to ByField – this is the default mode for searching, so it is not necessary to have this explicitly set during the RuleClass instantiation.

To perform full text searches, it is necessary to explicitly set the ConditionType of the RuleClass to ByText:

```
    new RuleClass()
    {
        Join = "AND",
        Condition = "=",
        ConditionType = SearchConditionType.ByText,
```

```
        Value = "Email Down"
    }
```

Notice in particular, that the Field member (which was present in the earlier Search examples) is not specified in the RuleClass – since the search is now performed against the full text catalog (by virtue of the ConditionType value of ByText), it is an error to also specify Field member in the RuleClass.

The following code sample illustrates how to search for Incident records with the matching terms “Email Down”:

```
ObjectQueryDefinition query = new ObjectQueryDefinition();
query.Select = new SelectClass();
// Retrieve just the IncidentNumber field value from the Incident,
// when invoking the search
FieldClass[] incidentFieldObjects = new FieldClass[] {
   new FieldClass()
   {
      Name = "IncidentNumber",
      Type = "Text"
   },
   new FieldClass()
   {
      Name = "Service",
      Type = "Text"
   }
};
query.Select.Fields = incidentFieldObjects;
query.From = new FromClass();
query.From.Object = "Incident";
query.Where = new RuleClass[] {
   new RuleClass()
   {
      Join = "AND",
      Condition = "=",
      ConditionType = SearchConditionType.ByText,
      Value = "Email Down"
```

```
        },
    };
    FRSHEATIntegrationSearchResponse searchResponse = frSvc.Search(authSessionKey, tenantId, query);
    if (searchResponse.status == "Success")
    {
        WebServiceBusinessObject[][] incidentList = searchResponse.objList;
        foreach (WebServiceBusinessObject[] incidentOuterList in incidentList)
        {
            foreach (WebServiceBusinessObject incident in incidentOuterList)
            {
                // Since we are just retrieving one field in the selection criteria
                // (i.e. IncidentNumber), this corresponds to
                // incident.FieldValues[0].Value when retrieving the results
                Console.WriteLine("Incident  matches the selection criteria", incident.FieldValues[0].Value);
            }
        }
    }
```

Formal Description of the Classes used by the Search WebMethod

From the previous sections, it can be seen that in order to execute the Search WebMethod properly, objects from several classes need to created and populated accordingly, before the web method is called.

Now that the earlier sections have presented several example use cases, this section will formally describe the various classes in greater detail.

ObjectQueryDefinition

The ObjectQueryDefinition class is defined as follows:

```
class ObjectQueryDefinition
{
    int Top;
    bool Distinct;
    FromClass From;
    SelectClass Select;
    List<RuleClass> Where;
```

```
    List<OrderByClass> OrderBy;
}
```

From the class definition, it can be seen that the ObjectQueryDefinition is used to model the various portions of a typical SQL SELECT statement, specifically:

- FROM clause

Class FromClass

- SELECT clause

Class SelectClass

- WHERE clause

Class RuleClass (implemented as a list)

- ORDER BY clause

Class OrderByClass (implemented as a list)

With the exception of RuleClass (used to model the WHERE clause in the SQL SELECT statement), the other classes are named according to the corresponding clause in the SQL SELECT statement.

Besides these classes, notice that there is also an integer member variable called “Top” – this can be used to constrain the number of records being returned by the Search (e.g. return the first 1000 matching results).

There is also a boolean member variable called “Distinct” – this can be used to return the distinct results, to eliminate the repeated values in the search results.

The following sections will describe the various component classes, which are used to model the respective clauses in the SELECT statement.

FromClass

The FromClass class is used to model the FROM clause in a typical SQL SELECT statement, and is defined as follows:

```
class FromClass
{
    string Object;
    List<FromLinkClass> Links;
}
```

The Object member variable needs to be populated with the name of the business object to search against:

```
        query.From = new FromClass();
        query.From.Object = "Incident";
```

If the saved search needs to be performed relative to specific child objects, the Links member variable also needs to be populated, using a List of FromLinkClass objects:

```
class FromLinkClass
```

```
{
    string Relation;
    string Object;
}
```

The FromLinkClass contains two member variables:

- The “Relation” member variable specifies the name of the Internal Reference Name of the relationship between the parent and child object

For example, for the “IncidentContainsJournal” relationship, the internal reference name of the relationship is blank – so to use this relationship, populate the Object member variable with the name of the child business object, and leave the Relation member variable as an empty string

- The “Object” member variable specifies the name of the child business object (e.g. “Journal#Email”)

From the earlier code samples, recall that to search for the Journal.Email records related to the current business object, the Links member variable of the FromClass needs to be populated as follows:

```
query.From.Links = new FromLinkClass[] {
    new FromLinkClass {
        Relation = "",
        Object = "Journal#Email"
    }
};
```

SelectClass

The SelectClass class is used to model the SELECT clause in a typical SQL SELECT statement, and is defined as follows:

```
class SelectClass
{
    bool All;
    List<FieldClass> Fields;
}
```

To select all the fields in the business object, create a new SelectClass object, and set the “All” member variable of the object to true:

```
query.Select = new SelectClass();
query.Select.All = true;
```

Note that if the Search is against the main business object (e.g. “Incident”) and its related child business objects (e.g. “Journal.Email”), setting the “All” member variable to true will return all the fields in the main business object, and all the fields in the child business object.

To restrict the set of fields to be returned by the Search web method, create a new List of FieldClass objects, and initialize it with FieldClass objects, corresponding to the fields of interest.

From the earlier code samples, recall that to return the IncidentNumber field of the Incident business object, and the Category field of the child Journal.Email object, the following statements are used:

```
FieldClass[] fieldObjects = new FieldClass[] {
    new FieldClass()
    {
        Name = "IncidentNumber",
        Type = "Text"
    },
    new FieldClass()
    {
        Name = "Journal.Category",
        Type = "Text"
    }
};
query.Select = new SelectClass();
query.Select.Fields = fieldObjects;
```

In particular, for fields from the main business object, the names of the fields can be provided as is, whereas for fields in the child business objects, the name of the field needs to be prefixed with the name of the child business object in the relationship.

So for example, if the “IncidentContainsJournal” relationship is used, the relationship is defined against the Incident and Journal (base) object, the Category field should be specified as “Journal.Category”.

RuleClass

The RuleClass class is used to model the WHERE clause in a typical SQL SELECT statement, and is defined as follows:

```
class RuleClass
{
    string Join;
    string Condition;
    SearchConditionType ConditionType;
    string Field;
    string Value;
    List<RuleClass> Rules;
```

}

The Field member variable is used to designate the name of the field, and the Value member variable specifies the value corresponding to the field.

In particular, for fields from the main business object, the names of the fields can be provided as is, whereas for fields in the child business objects, the name of the field needs to be prefixed with the name of the child business object in the relationship.

So for example, if the “IncidentContainsJournal” relationship is used, the relationship is defined against the Incident and Journal (base) object, the Category field should be specified as “Journal.Category”.

For example, the following statement can be used to search for Incident records with a Priority value equal to 1, where the Category value of the related Journal.Email records is equal to “Incoming Email”:

```
query.Where = new RuleClass[] {

new RuleClass()

    {

    Join = "AND",

        Condition = "=",

        Field = "Priority",

        Value = "1"

},

    new RuleClass()

    {

    Join = "AND",

        Condition = "=",

        Field = "Journal.Category",

        Value = "Incoming Email"

}

};
```

The Join property can contain a value of either “AND” / “OR”, for specifying how the RuleClass objects are to be related to one another.

The Condition member variable specifies the comparison operator used for relating the Field and the specified Value.

The allowable values for the Condition member variable include the following:

Operator

Meaning

=

Equal to

!=

Not Equal to

>

Greater than

<

Less than

>=

Greater than or Equal

<=

Less than or Equal

For grouping the rule criteria together, notice that within the RuleClass class, there is a member variable called Rules, which can optionally hold a list of RuleClass objects.

The Rules member variable can therefore be used to group related RuleClass objects together, by composing the RuleClass objects.

Recall from the earlier “Grouping the Rule Criteria” section, the following code example:

```
query.Where = new RuleClass[] {
    new RuleClass
    {
        Join = "AND",
        Field = "IncidentNumber",
        Condition = "<=",
        Value = "10010"
    },
    new RuleClass
    {
        Rules = new RuleClass[] {
            new RuleClass
            {
                Field = "Journal.Category",
                Condition = "=",
                Value = "Outgoing Email"
            },
```

```
            new RuleClass
            {
               Join = "OR",
               Field = "Journal.Subject",
               Condition = "=",
               Value = "Urgent Request"
            }
         }
      }
   };
```

From the above example, it can be seen that inside the second RuleClass at the top level, the Rules member variable is being initialized with another, inner List of RuleClass objects, where the criteria for Journal.Email is being expressed.

As explained in the earlier “Full Text Searching” section, normal SQL-style searches are performed, where the ConditionType is set to the enumeration value of SearchConditionType.ByField – this is the default mode for searches, and does not need to be explicitly specified in the RuleClass instantiation.

To support full text searches, set the ConditionType to the enumeration value of SearchConditionType.ByText, and do not include the Field member, when instantiating the RuleClass.

OrderByClass

The OrderByClass class is used to model the Order By clause in a typical SQL SELECT statement, and is defined as follows:

```
class OrderByClass
{
   public string Name;
   public string Direction;
}
```

Here, the Name member variable is used to specify the field to be ordered against, and the Direction member variable specifies whether the records should be specified in ascending or descending order, using the values of “ASC” or “DESC”, respectively.

For example, from the earlier code samples, assume that the search returns the Incident and related Journal.Email records.

If the Incident records should be sorted in ascending order based on the IncidentNumber, and the related Journal.Subject records should be sorted in ascending order based on Subject, the following code can be used for this:

```
      query.OrderBy = new OrderByClass[] {
         new OrderByClass()
         {
            Name = "IncidentNumber",
```

```
        Direction = "ASC"
    },
    new OrderByClass()
    {
        Name = "Journal.Subject",
        Direction = "ASC"
    }
};
```

Record Operations

CreateObject

This WebMethod creates a new instance of a single Business Object, and may also establish relationships with other objects. Runs initialization rules first, then applies the supplied values to the fields and invokes auto-fill, calculated, save and business rules in the same way, if the object was being created interactively via UI. Validation rules are also executed and they might prevent the saving operation, if the resulting object field values do not pass the validation.

Fields are initialized in the order provided.

Request Syntax:

```bash
FRSHEATIntegrationCreateBOResponse CreateObject(string sessionKey, string tenantId, ObjectCommandData commandData)
```

Parameters:

- **sessionKey**: Key received in the earlier Connect request
- **tenantId**: tenant for which the key is authenticated.
- **commandData**: a structure containing information about the creation request:

```
public class ObjectCommandData
{
public string ObjectId;
public string ObjectType;
public List<ObjectCommandDataFieldValue> Fields;
public List<LinkEntry> LinkToExistent;
}
```

- **ObjectId**: recid of the object to be created

- **ObjectType**: type of the object to be created
- **FieldValues**: a list of name-value pairs, containing the field names and new values for the fields of the business object that should be populated.
- **LinkToExistent**: references the LinkEntry class, which controls whether relationships between this object and other objects should be established.

```
public class LinkEntry

{

public string Action;

public string Relation;

public string RelatedObjectType;

public string RelationshipName;

public string RelatedObjectId;

public List<SearchCondition> SearchCriteria;

}
```

The following are the field members of the LinkEntry class:

- **Action**: either “link” or “unlink” – determines whether this object is to be linked with the other objects or unlinked. Only “link” is meaningful in CreateObject operation
- **Relation**: the relationship tag (shown as Internal Reference Name in Admin UI) for the relationship type to be established.
- **RelationshipName**: the relationship name (shown as display name in Admin UI) for the relationship type to be established
- **RelatedObjectType**: the type of the business object in an object reference notation to be linked with
- **RelatedObjectId**: the recId of the business object to be linked or unlinked. Optional, either RelatedObjectId or SearchCriteria must be provided.
- **SearchCriteria**: a list of structures defining search criteria for matching objects that have to be linked with this object.

```
public class SearchCondition

{

public string ObjectId;

public string ObjectDisplay;

public string JoinRule;

public string Condition;

public SearchConditionType ConditionType;

public string FieldName;

public string FieldDisplay;

public string FieldAlias;

public string FieldType;
```

public string FieldValue;

public string FieldValueDisplay;

public string FieldValueBehavior;

public string FieldStartValue;

public string FieldEndValue;

public List<SearchCondition> SubQuery;

}

- **ObjectId**: recId of the object to match
- **ObjectDisplay**: No definition present
- **JoinRule**: determines how individual search criteria combine together. Possible values are “and” and “or”
- **Condition**: how the field value should be matched. Possible values:

  = Equal to

  != Not Equal to

  > Greater than

  < Less than

  >= Greater or Equal

  <= Less or Equal

  → Begin with

  {} Contains

  !{} Does Not Contain

  0 Is Empty

  !0 Is Not Empty

  () In List

  !() Not In list

  >< In range

- **ConditionType**: controls how text fields are searched. Possible values:
  0 – use regular SQL field search (SQL like, contains clauses)
  1 – use fulltext SQL field search
- **FieldName**: field name
- **FieldValue**: field value
- **FieldValueBehavior**: either “single” or “list”
- **FieldStartValue**: start value for “In range” condition only
- **FieldEndValue**: end value for “In range” condition only

Return Value:

An FRSHEATIntegrationCreateBOResponse object, defined as follows:

public class FRSHEATIntegrationCreateBOResponse

{

public string status { get; set; }

public string exceptionReason { get; set; }

public string recId { get; set; }

public WebServiceBusinessObject obj { get; set; }

}

The FRSHEATIntegrationCreateBOResponse class comprises the following fields:

- **status** – this field provides a Status value indicating whether the operation was successful.
  A full description of the available Status values is provided in the table below.
- **exceptionReason** – if there is an exception thrown in the course of running the WebMethod, the exception information will be captured in this field.
- **recId** – the RecId of the newly created record, assuming the status of the Web Method is "Success"
- **obj** – assuming the business object record can be created successfully, this field returns the record as a WebServiceBusinessObject object

The following table lists the available status values, and describes how to interpret them.

Status

Explanation

Success

The business object can be successfully created.

The RecId of the newly created record can be accessed via the recId field of the response object, and the obj field references the newly created WebServiceBusinessObject

Error

The business object cannot be successfully created – the recId and obj fields will be null, and the exception will be stored in the exceptionReason field.

One typical error is the Table not found exception, which occurs when the specified business object does not exist in the tenant. Double-check to make sure that the name of the business object is spelled properly (e.g. "Incident", "Profile.Employee", etc.)

The other common error encountered, is when the specified field does not exist for the business object – here, the error message would be of the form:

ObjectTableMap: field <FieldName> is not found in table <Business Object>#

Double-check to make sure that the field name is spelle d correctly, and is actually defined for the given business object.

A third common error is to specify a value for a field, which does not exist in the associated validation list – in such cases, the following exception would be encountered:

<BusinessObject>.<Field>:`<FieldValue>` is not in the validation list

To specify date/time values, the string value should be specified using ISO 8601 format, and the value itself should be relative to UTC.

So the date/time value can be specified in one of the following two ways:

yyyy-mm-dd hh:mm

or

yyyy-mm-ddThh:mm

Either a space character or “T” character can be used to separate between the date and time values.

The following are two examples of specifying a date/time value of March 26th, 2013, 18:38 UTC, relative to the above two formats:

2013-03-26 18:38

2013-03-26T18:38

Example:

The following example will first create a brand new Change record with specific field values, then locate an existing CI.Computer record, by means of the Search() WebMethod, and will link the two records together, by means of the CreateObject() WebMethod.

```
ObjectCommandData data = new ObjectCommandData();

data.ObjectType = "Change#";

List<ObjectCommandDataFieldValue> dataFields = new List<ObjectCommandDataFieldValue>();

Dictionary<string, object> fields = new Dictionary<string, object>();

fields["RequestorLink"] = "FB884D18F7B746A0992880F2DFFE749C";

fields["Subject"] = "Need to swap out the hard disk";

fields["Description"] = "The hard drive just crashed - need to replace with a new drive from the vendor";

fields["Status"] = "Logged";

fields["TypeOfChange"] = "Major";

fields["OwnerTeam"] = "Operations";

fields["Owner"] = "Admin";

fields["Impact"] = "Medium";

fields["Urgency"] = "Medium";

fields["CABVoteExpirationDateTime"] = "2013-03-26 18:38:30";

foreach (string key in fields.Keys)

{
```

```
dataFields.Add(new ObjectCommandDataFieldValue()
{
Name = key,
Value = fields[key].ToString()
});
}
data.Fields = dataFields.ToArray();
// Here we will identify a CI.Computer record, to link to the
// new Change record
// For this example, we will attempt to locate the CI.Computer record
// with the name of "APAC-DEPOT-SERV01", and retrieve its RecId
ObjectQueryDefinition ciQuery = new ObjectQueryDefinition();
// Just retrieve only the RecId field for the CI.Computer record
FieldClass[] ciFieldObjects = new FieldClass[] {
new FieldClass()
{
Name = "RecId",
Type = "Text"
}
};
ciQuery.Select = new SelectClass();
ciQuery.Select.Fields = ciFieldObjects;
ciQuery.From = new FromClass();
// Search for the record against the CI.Computer member object
ciQuery.From.Object = "CI.Computer";
ciQuery.Where = new RuleClass[]
{
// Provide the criteria to search for the CI.Computer
// Here, we will search for the CI.Computer by its Name
new RuleClass()
{
```

```
Condition = "=",
Field = "Name",
Value = "APAC-DEPOT-SERV01"
}
};
// Pass in the ObjectQueryDefinition for the query
FRSHEATIntegrationSearchResponse searchResponse = frSvc.Search(authSessionKey, tenantId, ciQuery);
WebServiceBusinessObject[][] cilist = searchResponse.objList;
// Assuming that the CI record is uniquely identified by its Name, and
// because the above query does not join with other tables, we should be
// able to locate the CI record, by accessing cilist[0][0], in the
// list of list of WebServiceBusinessObjects
WebServiceBusinessObject ci = cilist[0][0];
string ciRecId = ci.RecID;
// Define the LinkEntry record, to link the new Change record to the CI
// record, by means of the RecId of the Change (i.e. ciRecId), as
// determined above
data.LinkToExistent = new LinkEntry[]
{
new LinkEntry()
{
Action = "Link",
Relation = "",
RelatedObjectType = "CI#",
RelatedObjectId = ciRecId
}
};
// If the record creation succeeds, the result variable will store the
// RecId of the new Change record, otherwise it will be null
FRSHEATIntegrationCreateBOResponse result = frSvc.CreateObject(authSessionKey, tenantId, data);
if (result.status == "Success")
```

```
{

Console.WriteLine("A new Change record is created with RecId of ", result.recId);

}
```

The next example will create a new Profile.Employee record, and link the user to the respective roles and teams.

Notice in particular, that the password value is specified in plain text – it will be automatically converted to the internal hashed value, upon save of the record.

```
ObjectCommandData data = new ObjectCommandData();

data.ObjectType = "Profile#Employee";

List<ObjectCommandDataFieldValue> dataFields = new List<ObjectCommandDataFieldValue>();

Dictionary<string, object> fields = new Dictionary<string, object>();

fields["Status"] = "Active";

fields["FirstName"] = "Brian";

fields["LastName"] = "Wilson";

fields["LoginID"] = "BWilson";

fields["IsInternalAuth"] = true;

// Notice when setting the password for the Employee, that the plain text

// password is specified here - it will be converted to the hashed value

// upon save of the record

fields["InternalAuthPasswd"] = "Manage1t";

fields["PrimaryEmail"] = "BWilson@example.com";

fields["Phone1"] = "14158665309";

// RecId for the "Admin" user, to serve as the Manager for the new Employee

fields["ManagerLink"] = "FB884D18F7B746A0992880F2DFFE749C";

// RecId for the "GMI" Org Unit, for the OrgUnit of the new Employee

fields["OrgUnitLink"] = "4A05123D660F408997A4FEE714DAD111";

fields["Team"] = "IT";

fields["Department"] = "Operations";

fields["Title"] = "Administrator";

foreach (string key in fields.Keys)

{

dataFields.Add(new ObjectCommandDataFieldValue()
```

```
{
    Name = key,
        Value = fields[key].ToString()
});
}
data.Fields = dataFields.ToArray();
data.LinkToExistent = new LinkEntry[]
{
// First we link the new Employee to the "SelfService" and
// "ServiceDeskAnalyst" roles by RecID
// The internal reference name for the relationship between
// Profile.Employee and Frs_def_role is empty, so we leave
// the Relation attribute in the LinkEntry empty in this case
    // Link to "SelfService" role
new LinkEntry()
    {
    Action = "Link",
        Relation = "",
        RelatedObjectType = "Frs_def_role#",
        RelatedObjectId = "0a4724d8478b451abea3fb44d33db1b6"
},
// Link to "ServiceDeskAnalyst" role
    new LinkEntry()
    {
    Action = "Link",
        Relation = "",
        RelatedObjectType = "Frs_def_role#",
        RelatedObjectId = "06d780f5d7d34119be0d1bc8fc997947"
},
    // We then link the new Employee to the "IT" and "HR" teams
    // The internal reference name for the relationship between
```

```
    // Profile.Employee and StandardUserTeam is "Rev2", so we
    // specify this in the Relation attribute in the LinkEntry
    // Link to the "IT" team
    new LinkEntry()
    {
    Action = "Link",
        Relation = "Rev2",
        RelatedObjectType = "StandardUserTeam#",
        RelatedObjectId = "10F60157A4F34A4F9DDB140E2328C7A6"
},
    // Link to the "HR" team
    new LinkEntry()
    {
    Action = "Link",
        Relation = "Rev2",
        RelatedObjectType = "StandardUserTeam#",
        RelatedObjectId = "1FF47B9EDA3049CC92458CE3249BA349"
}
};
FRSHEATIntegrationCreateBOResponse result = frSvc.CreateObject(authSessionKey, tenantId, data);
if (result.status == "Success")
{
Console.WriteLine("A new Employee record is created with recId of ", result.recId);
}
```

UpdateObject

This WebMethod updates a single object by changing its field values, and may also establish or break relationships with other objects. Note that the auto-fill, calculated, save and business rules run during the update and they may trigger additional field changes. Validation rules are also executed and they might block the update operation if the resulting object field values do not pass the validation.

The order of operations is preserved during the update.

Request Syntax:

```bash
FRSHEATIntegrationUpdateBOResponse UpdateObject(string sessionKey, string tenantId, ObjectCommandData commandData)
```

Parameters:

- **sessionKey**: Key received in the earlier Connect request
- **tenantId**: tenant for which the key is authenticated.
- **commandData**: see description of ObjectCommandData in CreateObject request

Return Value:

An FRSHEATIntegrationUpdateBOResponse object, defined as follows:

public class FRSHEATIntegrationUpdateBOResponse

{

public string status { get; set; }

public string exceptionReason { get; set; }

public string recId { get; set; }

public WebServiceBusinessObject obj { get; set; }

}

The FRSHEATIntegrationUpdateBOResponse class comprises the following fields:

- **status** – this field provides a Status value indicating whether the operation was successful.
  A full description of the available Status values is provided in the table below.
- **exceptionReason** – if there is an exception thrown in the course of running the WebMethod, the exception information will be captured in this field.
- **recId** – the RecId of the updated record, assuming the status of the Web Method is “Success”
- **obj** – assuming the business object record can be updated successfully, this field returns the updated record as an WebServiceBusinessObject object

The following table lists the available status values, and describes how to interpret them.

Status

Explanation

Success

The business object can be successfully updated.

The RecId of the updated record can be accessed via the recId field of the response object, and the obj field references the updated WebServiceBusinessObject

Error

The business object cannot be successfully updated – the recId and obj fields will be null, and the exception will be stored in the exceptionReason field.

One typical error is the Table not found exception, which occurs when the specified business object does not exist in the tenant. Double-check to make sure that the name of the business object is spelled properly (e.g. “Incident”, “Profile.Employee”, etc.)

The other common error encountered, is when the specified field does not exist for the business object – here, the error message would be of the form:

ObjectTableMap: field <FieldName> is not found in table <Business Object>#

Double-check to make sure that the field name is spelled correctly, and is actually defined for the given business object.

A third common error is to specify a value for a field, which does not exist in the associated validation list – in such cases, the following exception would be encountered:

<BusinessObject>.<Field>:`<FieldValue>` is not in the validation list

To specify date/time values, the string value should be specified using ISO 8601 format, and the value itself should be relative to UTC.

So the date/time value can be specified in one of the following two ways:

yyyy-mm-dd hh:mm

or

yyyy-mm-ddThh:mm

Either a space character or “T” character can be used to separate between the date and time values.

The following are two examples of specifying a date/time value of March 26th, 2013, 18:38 UTC, relative to the above two formats:

2013-03-26 18:38

2013-03-26T18:38

Example:

The following example will locate an existing Change record and CI.Computer record, by means of the Search() WebMethod, and will link the two records together, by means of the UpdateObject() WebMethod.

```
// First, locate the Change record to update, using the ChangeNumber

// (e.g. Change 21)

ObjectQueryDefinition changeQuery = new ObjectQueryDefinition();

// Just retrieve only the RecId field for the Change record

FieldClass[] changeFieldObjects = new FieldClass[] {

new FieldClass()

{

Name = "RecId",

Type = "Text"
```

```
}
};
changeQuery.Select = new SelectClass();
changeQuery.Select.Fields = changeFieldObjects;
changeQuery.From = new FromClass();
// Search for the record against the Change object
changeQuery.From.Object = "Change";
changeQuery.Where = new RuleClass[] {
new RuleClass()
{
// Provide the criteria to search for the Change
// Here, we will search for the Change by its ChangeNumber
Condition = "=",
Field = "ChangeNumber",
Value = "21"
}
};
// Pass in the ObjectQueryDefinition for the query
FRSHEATIntegrationSearchResponse changeSearchResponse = frSvc.Search(authSessionKey, tenantId, changeQuery);
WebServiceBusinessObject[][] changeList = changeSearchResponse.objList;
// Assuming that the Change record is uniquely identified by the
// ChangeNumber, and because the above query does not join with other
// tables, we should be able to locate the Change record, by accessing
// changeList[0][0], in the list of list of WebServiceBusinessObjects
WebServiceBusinessObject change = changeList[0][0];
string changeRecId = change.RecID;
// Now locate the CI.Computer record, to link with the existing Change
// Here we will attempt to locate the CI.Computer record with
// the name of "APAC-DEPOT-SERV01" and retrieve its RecId
ObjectQueryDefinition ciQuery = new ObjectQueryDefinition();
// Just retrieve only the RecId field of the CI for the matching result
```

```
FieldClass[] ciFieldObjects = new FieldClass[] {
new FieldClass()
{
    Name = "RecId",
        Type = "Text"
}
};
ciQuery.Select = new SelectClass();
ciQuery.Select.Fields = ciFieldObjects;
ciQuery.From = new FromClass();
// Search for the record against the CI.Computer member object
ciQuery.From.Object = "CI.Computer";
ciQuery.Where = new RuleClass[]
{
// Search for the CI.Computer by its Name
new RuleClass()
{
Condition = "=",
Field = "Name",
Value = "EMEA-EXCH-SERV01"
}
};
// Pass in the ObjectQueryDefinition for the query
FRSHEATIntegrationSearchResponse ciSearchResponse = frSvc.Search(authSessionKey, tenantId, ciQuery);
WebServiceBusinessObject[][] cilist = ciSearchResponse.objList;
// Assuming that the CI record is uniquely identified by Name, and
// because the above query does not join with other tables, we should
// be able to locate the CI record, by accessing cilist[0][0], in the
// list of list of WebServiceBusinessObjects
WebServiceBusinessObject ci = cilist[0][0];
// Since we are only retrieving the RecId field for CI, it will appear
```

```
// as the first item in the list of Fields, i.e. ci.FieldValues[0]
string ciRecId = (string)ci.FieldValues[0].Value;
// At this point, we now have the RecId of the Change and CI records,
// and can proceed with the update
// For the ObjectCommandData, use the changeRecId value that was
// determined above, for looking up the record to update
ObjectCommandData data = new ObjectCommandData();
data.ObjectType = "Change#";
data.ObjectId = changeRecId;
List<ObjectCommandDataFieldValue> dataFields = new List<ObjectCommandDataFieldValue>();
Dictionary<string, object> fields = new Dictionary<string, object>();
// To demonstrate that the existing field value can be updated, set the
// Urgency of the existing Change record to "High"
fields["Urgency"] = "Medium";
// Update the CABVoteExpirationDateTime to a specific date/time value
fields["CABVoteExpirationDateTime"] = "2013-03-26 18:38:30";
foreach (string key in fields.Keys)
{
dataFields.Add(new ObjectCommandDataFieldValue()
    {
    Name = key,
        Value = fields[key].ToString()
    });
}
data.Fields = dataFields.ToArray();
data.LinkToExistent = new LinkEntry[]
{
new LinkEntry()
    {
    Action = "Link",
        Relation = "",
```

```
RelatedObjectType = "CI#",
RelatedObjectId = ciRecId
}
};
FRSHEATIntegrationUpdateBOResponse response = frSvc.UpdateObject(authSessionKey, tenantId, data);
if (response.exceptionReason != null)
{
Console.WriteLine("Encountered the following error while updating the record: ", response.exceptionReason);
}
```

DeleteObject

Deletes the specified business object record.

Request Syntax:

FRSHEATIntegrationDeleteBOResponse DeleteObject(string sessionKey, string tenantId, ObjectCommandData commandData)

Parameters:

- **sessionKey**: Key received in the earlier Connect request
- **tenantId**: tenant for which the key is authenticated.
- **commandData**: see description of ObjectCommandData in CreateObject request

Return Value:

An FRSHEATIntegrationDeleteBOResponse object, defined as follows:

```
public class FRSHEATIntegrationDeleteBOResponse
{
    public string status { get; set; }
    public string exceptionReason { get; set; }
}
```

The FRSHEATIntegrationDeleteBOResponse class comprises the following fields:

- **status** – this field provides a Status value indicating whether the operation was successful.
  A full description of the available Status values is provided in the table below.
- **exceptionReason** – if there is an exception thrown in the course of running the WebMethod, the exception information will be captured in this field.

The following table lists the available status values, and describes how to interpret them.

Status

Explanation

Success

The business object record can be successfully deleted.

Note that a value of Success is returned, either if the record is successfully deleted from the tenant, or that the indicated record cannot be found in the tenant.

Error

An error has occurred, in the process of deleting the indicated record from the system.

Typically the error can occur if the specified business object does not exist in the system – here, the error message would be:

definition for business object <BusinessObject># was not found

In such cases, please ensure that the specified business object exists in the tenant.

Example:

The following example will delete the Incident record with the provided RecID value.

```
ObjectCommandData data = new ObjectCommandData();

data.ObjectType = "Incident#";

data.ObjectId = "96F889A8CE6E4F9C8B3A99852F788670";

FRSHEATIntegrationDeleteBOResponse response = frSvc.DeleteObject(authSessionKey, tenantId, data);

if (response.status != "Success")

{

Console.WriteLine("Ran into the following error when deleting the record: " + response.exceptionReason);

}
```

Attachments

AddAttachment

This WebMethod adds an attachment to the specified business object record.

Request syntax:

FRSHEATIntegrationAddAttachmentResponse AddAttachment(string sessionKey, string tenantId, ObjectAttachmentCommandData commandData)

Parameters:

- **sessionKey**: Key received in the earlier Connect request
- **tenantId**: tenant for which the key is authenticated.
- **commandData**: structure containing information about the attachment:

public class ObjectAttachmentCommandData

{

public string ObjectId;

public string ObjectType;

public string fileName;

public Byte[] fileData;

}

- **ObjectId**: Record ID of the new attachment.
- **ObjectType**: Type of the main Business Object to which this attachment is attached in the object reference notation.
- **Filename**: Name of the file.
- **fileData**: Actual file bytes.

Return Value:

An FRSHEATIntegrationAddAttachmentResponse object, defined as follows:

```
public class FRSHEATIntegrationAddAttachmentResponse
{
    public string status { get; set; }
    public string exceptionReason { get; set; }
}
```

The FRSHEATIntegrationAddAttachmentResponse class comprises the following fields:

- **status** – this field provides a Status value indicating whether the operation was successful.
  A full description of the available Status values is provided in the table below.
- **exceptionReason** – if there is an exception thrown in the course of running the WebMethod, the exception information will be captured in this field.

The following table lists the available status values, and describes how to interpret them.

Status

Explanation

Success

The attachment can be successfully added to the business object record.

Error

The attachment cannot be successfully added to the business object record, and the exception information is stored in the exceptionReason field in the response object.

One typical reason for the error, is when the attachment has a file extension which has been disallowed for the given tenant – here, the error message would be of the form:

The file that is uploaded is a restricted file extension. Please contact your System Administrator for a list of allowed file extensions for upload.

Another common reason for the error is when the business object record cannot be successfully found – here, the error message would be of the form:

Attachment upload finished unsuccessfully.

A third reason for the error occurs when the specified business object does not exist in the tenant – here, the error message would be of the form:

definition for business object <Business Object># was not found

Example:

The following example reads in a sample image from the user's local filesystem, and attaches the file to the indicated Incident record.

```
const string fileName = "C:\\Temp\\sample.jpg";

using (FileStream fs = new FileStream(fileName, FileMode.Open, FileAccess.Read))

{

using (BinaryReader r = new BinaryReader(fs))

{

byte[] AttachmentData = new byte[fs.Length];

for (int i = 0; i < fs.Length; i++)

{

AttachmentData[i] = r.ReadByte();

}

ObjectAttachmentCommandData data = new ObjectAttachmentCommandData()

{

ObjectId = "9981FBEBAA8B4EE2820364505855ABC2",

ObjectType = "Incident#",

fileName = "sample.png",

fileData = AttachmentData

};

FRSHEATIntegrationAddAttachmentResponse response = frSvc.AddAttachment(authSessionKey, tenantId, data);

if (response.status != "Success")

{

Console.WriteLine("Encountered the following error while adding the attachment to the record: " + response.exceptionReason);
```

}

}

}

ReadAttachment

This WebMethod is used for reading out the data of a specific Attachment record, identified by its RecId value.

Request syntax:

FRSHEATIntegrationReadAttachmentResponse ReadAttachment(string sessionKey, string tenantId, ObjectAttachmentCommandData commandData)

Parameters:

- **sessionKey**: Key received in the earlier Connect request
- **tenantId**: tenant for which the key is authenticated.
- **commandData**: structure containing information about the attachment:
  `public class ObjectAttachmentCommandData`
  `{`
  `public string ObjectId;`
  `}`
- **ObjectId**: Record ID of the new attachment.

Return Value:

An FRSHEATIntegrationReadAttachmentResponse object, defined as follows:

```
public class FRSHEATIntegrationReadAttachmentResponse
{
    public string status { get; set; }
    public string exceptionReason { get; set; }
    public byte[] attachmentData { get; set; }
}
```

The FRSHEATIntegrationReadAttachmentResponse class comprises the following fields:

- status – this field provides a Status value indicating whether the operation was successful.
  A full description of the available Status values is provided in the table below.
- exceptionReason – if there is an exception thrown in the course of running the WebMethod, the exception information will be captured in this field.
- attachmentData - a byte array, containing the contents of the specified Attachment

The following table lists the available status values, and describes how to interpret them.

Status

Explanation

Success

The specified attachment record can be successfully retrieved from the tenant, and the data is returned via the attachmentData byte array field in the response object.

NotFound

The specified attachment record cannot be located in the tenant.

Double-check the records in the Attachment business object, to confirm that the desired Attachment record does in fact exist.

Error

An unforeseen error was encountered during the retrieval of the attachment record from the tenant, and the exception information is stored in the exceptionReason field in the response object.

Under most circumstances, the attachment retrieval should result in either a Status of “Success” or “NotFound”.

Example:

```
byte[] AttachmentData;

const string fileName = "C:\\Temp\\test.png";

ObjectAttachmentCommandData data = new ObjectAttachmentCommandData()

{

ObjectId = "94069732037142E7BF3D81DB02128289", // RecId of the Attachment record

};

FRSHEATIntegrationReadAttachmentResponse response = frSvc.ReadAttachment(authSessionKey, tenantId, data);

if (response.status == "Success")

{

AttachmentData = response.attachmentData;

if (AttachmentData != null)

{

using (FileStream fs = new FileStream(fileName, FileMode.Create))

        {

        using (BinaryWriter w = new BinaryWriter(fs))

            {

            for (int i = 0; i < AttachmentData.Length; i++)

                {

                    w.Write(AttachmentData[i]);
```

```
                }
}
        }
}
}
else if (response.status == "NotFound")
{
Console.WriteLine("The attachment record with the specified RecId, cannot be located in the tenant");
}
else
{
Console.WriteLine("Encountered the following error while reading the attachment from the record: " + response.exceptionReason);
}
```

> **Important Note**: For .NET based Web Service clients, the application may run into the following error when retrieving large attachments, via the Web Service:

CommunicationException occurred: The maximum message size quota for incoming messages (65536) has been exceeded. To increase the quota, use the MaxReceivedMessageSize property on the appropriate binding element.

Here, the .NET client application which consumes the web service, needs to be configured so that the “MaxReceivedMessageSize” and “MaxBufferSize” attributes are set sufficiently large in the App.config file, to accommodate the large attachment sizes; e.g.

```
<binding name="IPCMServiceSoap" closeTimeout="00:01:00" openTimeout="00:01:00"
receiveTimeout="00:10:00" sendTimeout="00:01:00" allowCookies="false"
bypassProxyOnLocal="false" hostNameComparisonMode="StrongWildcard"
maxBufferSize="65536" maxBufferPoolSize="524288" maxReceivedMessageSize="65536"
messageEncoding="Text" textEncoding="utf-8" transferMode="Buffered"
useDefaultWebProxy="true">
...
</binding>
```

Please note that the size values above are represented in bytes.

Metadata access

You can import an existing set of Business Objects into the FRS HEAT tenant using the Business Object Uploader tool. The tool communicates with the FRS SaaS application server using public web services to facilitate user logins, population of data-entry forms and the upload of Business Object data.

The workflow in the Business Object Uploader is as follows:

1. Log into an instance.
   Load the list of all allowable Business Objects.
   Select a business object,
   Load the metadata for the selected object.
   Complete the data for the fields being imported, if needed.
   Submit new object to system

When designing your API, to know what objects you can create, you need to obtain a list of allowed objects (GetAllAllowedObjectNames)

To know what fields you want to submit to the create object call, you need to obtain the metadata (GetSchemaForObject).

In some cases, you may want to upload objects even with “special” fields visible (like RecID) (GetAllSchemaForObject).

GetSchemaForObject

Retrieves an XML version of the metadata behind a Business Object. In the process, it screens out properties not appropriate for end user consumption, such as the RecID

Request syntax:

string GetSchemaForObject(string sessionKey, string tenantId, string objectName)

Parameters:

- **sessionKey**: Key received in the earlier Connect request
- **tenantId**: Tenant for which the key is authenticated.
- **objectName**: Name of the business object, in object reference notation (e.g. "Incident#")

Return Value:

A string value representing the schema for the given business object, which is represented in XML.

Example:

string schemaDoc = frSvc.GetSchemaForObject(authSessionKey, tenantId, "Incident#");

GetAllSchemaForObject

Retrieves an XML version of the metadata behind an object, including all fields such as RecId.

Request syntax:

string GetAllSchemaForObject(string sessionKey, string tenantId, string objectName)

Parameters:

- **sessionKey**: Key received in the earlier Connect request
- **tenantId**: Tenant for which the key is authenticated.
- **objectName**: Name of the business object, in object reference notation (e.g. "Incident#")

Return Value:

A string value representing the schema for the given business object, which is represented in XML.

Example:

string schemaDoc = frSvc.GetSchemaForObject(authSessionKey, tenantId, "Incident#");

GetAllAllowedObjectNames

Retrieves a list of all business objects.

Request syntax:

List<string> GetAllAllowedObjectNames(string sessionKey, string tenantId);

Parameters:

- **sessionKey**: Key received in the earlier Connect request
- **tenantId**: Tenant for which the key is authenticated.

Return Value:

An array of strings, where each item corresponds to the name of the business object.

Example:

string[] boNameArray = frSvc.GetAllAllowedObjectNames(authSessionKey, tenantId);

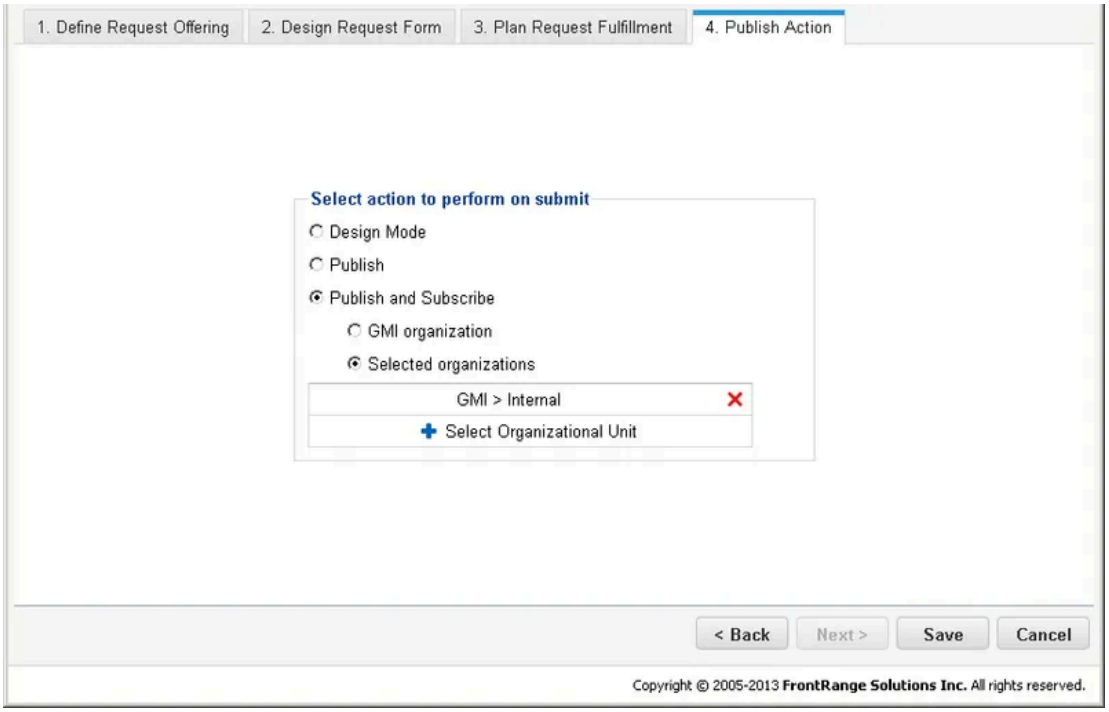
9.4. Web Methods

The following sections describe the full details for these

Important Note Regarding Service Request Subscription

> Note that when accessing the various Service Catalog web methods via the API, most of the Web Methods are invoked with respect to the Id of the Service Request Subscription record, and not with the Id of the Request Offering itself.

Recall that when the Request Offering is defined, the user needs to specify whether it is configured as "Published and Subscribe", or "Publish".



- When configured as **Published and Subscribe**, users belonging to the indicated Organizational Unit and below will have access to the Request Offering.
- When configured as **Published**, user will have access to the Request Offering, based on the Service Level Agreement (SLA) for the Organizational Unit the user belongs to.

The **ServiceReqSubscription** record (used internally by the Service Catalog) contains the information regarding which Organizational Unit is associated with the Request Offering, and the SLA information, if it is configured relative to the SLA.

In the following sections, several of the Web Methods will make use of the **Id value** of the **ServiceReqSubscription** record. For this, a helper web method called **GetSubscriptionId()** can be used, for obtaining the Subscription Id, which can then be used in any subsequent operations, such as **SubmitRequest()**.

GetCategories

Get the list of available Categories for the Service Catalog.

Request syntax:

```Bash
public FRSHEATGetCategoriesResponse GetCategories(string sessionKey, string tenantId)
```

Parameters:

- **sessionKey**: Key received in the earlier Connect request
- **tenantId**: tenant for which the key is authenticated.

Return Value:

An FRSHEATGetCategoriesResponse object, defined as follows:

```
public class FRSHEATGetCategoriesResponse
{
    public string status { get; set; }
    public string exceptionReason { get; set; }
    public List<FRSHEATServiceReqCategory> srCategories { get; set; }
}
```

The FRSHEATIntegrationFindBOResponse class comprises the following fields:

- **status** – this field provides a Status value indicating whether the operation was successful.
  A full description of the available Status values is provided in the table below.
- **exceptionReason** – if there is an exception thrown in the course of running the Connect WebMethod, the exception information will be captured in this field.
- **srCategories** – a list of FRSHEATServiceReqCategory objects, each of which represents the Category value in the Service Catalog.

The FRSHEATServiceReqCategory class is defined as follows:

```
public class FRSHEATServiceReqCategory
{
    public string strRecId;
    public string strName;
    public string strDescription;
}
```

The FRSHEATServiceReqCategory class comprises the following fields:

- **strRecId** – the RecId corresponding to the Category value
- **strName** – the Name of the Category
- **strDescription** – the Description for the Category

The following table lists the available status values, and describes how to interpret them.

Status	Explanation
Success	The list of Categories in the Service Catalog can be successfully retrieved, and are available via the srCategories list.
Error	An Error was encountered during the execution of this Web Method - the corresponding exceptionReason member should be inspected, to determine why the web method has failed.

Example

```
FRSHEATGetCategoriesResponse getCategoriesResponse = frSvc.GetCategories(authSessionKey, tenantId);
if (getCategoriesResponse.status == "Success")
{
   foreach (FRSHEATServiceReqCategory srCategory in getCategoriesResponse.srCategories)
   {
      Console.WriteLine("Category: ", srCategory.strName);
   }
}
```

GetCategoryTemplates

Get the list of Request Offerings in the Service Catalog, which belong to the indicated Category

Request syntax:

```bash
public FRSHEATGetTemplatesResponse GetCategoryTemplates(string sessionKey, string tenantId, string categoryid, string searchString
```

Parameters:

- **sessionKey**: Key received in the earlier Connect request
- **tenantId**: tenant for which the key is authenticated.
- **categoryId**: the RecId of the Category in the Service Catalog, obtained via the GetCategories WebMethod
- **searchString**: a substring for determining the matching Request Offerings

Return Value:

An FRSHEATGetTemplatesResponse object, defined as follows:

```
public class FRSHEATGetTemplatesResponse
{
```

```
    public string status { get; set; }

    public string exceptionReason { get; set; }

    public List<FRSHEATServiceReqTemplateListItem> srtList { get; set; }

}
```

The FRSHEATGetTemplatesResponse class comprises the following fields:

- **status** – this field provides a Status value indicating whether the operation was successful.
  A full description of the available Status values is provided in the table below.
- **exceptionReason** – if there is an exception thrown in the course of running the Connect WebMethod, the exception information will be captured in this field.
- **srtList** – a list of FRSHEATServiceReqTemplateListItem objects, each of which represents the Request Offering matching the search criteria.

The FRSHEATServiceReqTemplateListItem class is defined as follows:

```
public class FRSHEATServiceReqTemplateListItem

{

    public string strRecId;

    public string strName;

    public string strDescription;

    public string strSubscriptionId;

}
```

The FRSHEATServiceReqTemplateListItem class comprises the following fields:

- **strRecId** – the RecId corresponding to the Request Offering
- **strName** – the Name of the Request Offering
- **strDescription** – the Description for the Request Offering
- **strSubscriptionId** – the SubscriptionId

The following table lists the available status values, and describes how to interpret them.

Status	Explanation
Success	The list of matching Request Offerings in the Service Catalog can be successfully retrieved, and are available via the srtList member.
Error	An Error was encountered during the execution of this Web Method - the corresponding exceptionReason member should be inspected, to determine why the web method has failed.

Example

```
FRSHEATGetTemplatesResponse getTemplatesResponse = frSvc.GetCategoryTemplates(authSessionKey, tenantId, categoryid, searchString);
```

```
if (getTemplatesResponse.status == "Success")
{
    foreach (FRSHEATServiceReqTemplateListItem srTemplateListItem in getTemplatesResponse.srtList)
    {
        Console.WriteLine("Request Offering:  ", srTemplateListItem.strSubscriptionId, srTemplateListItem.strName);
    }
}
```

GetAllTemplates

Get the full list of Request Offerings in the Service Catalog

Request syntax:

public FRSHEATGetTemplatesResponse GetAllTemplates(string sessionKey, string tenantId)

Parameters:

- **sessionKey**: Key received in the earlier Connect request
- **tenantId**: tenant for which the key is authenticated.

Return Value:

An FRSHEATGetTemplatesResponse object, defined as follows:

```
public class FRSHEATGetTemplatesResponse
{
    public string status { get; set; }
    public string exceptionReason { get; set; }
    public List<FRSHEATServiceReqTemplateListItem> srtList { get; set; }
}
```

The FRSHEATGetTemplatesResponse class comprises the following fields:

- **status** – this field provides a Status value indicating whether the operation was successful.
  A full description of the available Status values is provided in the table below.
- **exceptionReason** – if there is an exception thrown in the course of running the Connect WebMethod, the exception information will be captured in this field.
- **srtList** – a list of FRSHEATServiceReqTemplateListItem objects, each of which represents the Request Offering matching the search criteria.

The FRSHEATServiceReqTemplateListItem class is defined as follows:

```
public class FRSHEATServiceReqTemplateListItem
```

```
{
    public string strRecId;
    public string strName;
    public string strDescription;
    public string strSubscriptionId;
}
```

The FRSHEATServiceReqTemplateListItem class comprises the following fields:

- **strRecId** – the RecId corresponding to the Request Offering
- **strName** – the Name of the Request Offering
- **strDescription** – the Description for the Request Offering
- **strSubscriptionId** – the Subscription Id

The following table lists the available status values, and describes how to interpret them.

Status	Explanation
Success	The list of matching Request Offerings in the Service Catalog can be successfully retrieved, and are available via the srtList member.
Error	An Error was encountered during the execution of this Web Method - the corresponding exceptionReason member should be inspected, to determine why the web method has failed.

Example

```
FRSHEATGetTemplatesResponse getTemplatesResponse = frSvc.GetAllTemplates(authSessionKey, tenantId);
if (getTemplatesResponse.status == "Success")
{
    foreach (FRSHEATServiceReqTemplateListItem srTemplateListItem in getTemplatesResponse.srtList)
    {
        Console.WriteLine("Request Offering:  ", srTemplateListItem.strSubscriptionId, srTemplateListItem.strName);
    }
}
```

GetSubscriptionId

Fetches the Subscription ID corresponding to the Request Offering for the current user

Request syntax:

```Bash
public FRSHEATGetSubscriptionIdResponse GetSubscriptionId(string sessionKey, string tenantId, string name)
```

Parameters:

- **sessionKey**: Key received in the earlier Connect request
- **tenantId**: tenant for which the key is authenticated.
- **name**: name of the Request Offering

Return Value:

An FRSHEATGetSubscriptionIdResponse object, defined as follows:

```
public class FRSHEATGetSubscriptionIdResponse
{
    public string status { get; set; }
    public string exceptionReason { get; set; }
    public string subscriptionId { get; set; }
}
```

The FRSHEATGetSubscriptionIdResponse class comprises the following fields:

- **status** – this field provides a Status value indicating whether the operation was successful.
  A full description of the available Status values is provided in the table below.
- **exceptionReason** – if there is an exception thrown in the course of running the Connect WebMethod, the exception information will be captured in this field.
- **subscriptionId** – the Subscription ID corresponding to the Request Offering for the current user.

The following table lists the available status values, and describes how to interpret them.

Status	Explanation
Success	The Subscription ID of the indicated Request Offering can be accessed via the subscriptionId member
NotFound	There is no Request Offering available with the given name in the tenant – please ensure that the name of the Request Offering is spelled correctly
Error	An Error was encountered during the execution of this Web Method - the corresponding exceptionReason member should be inspected, to determine why the web method has failed.

Example

FRSHEATGetSubscriptionIdResponse subscriptionIdResponse = frSvc.GetSubscriptionId(authSessionKey, tenantId, offeringName);

string offeringname = "Domain Password Reset";

```
if (subscriptionIdResponse.status == "Success")
{
    string subscriptionId = subscriptionIdResponse.subscriptionId;
    Console.WriteLine("The Subscription Id for the \"\" Request Offering is ", offeringName, subscriptionId);
}
```

GetPackageData

Retrieve the details of the indicated Request Offering

Request syntax:

Bash

```
public FRSHEATGetPackageDataResponse GetPackageData(string sessionKey, string tenantId, string strSubscrRecId)
```

Parameters:

- **sessionKey**: Key received in the earlier Connect request
- **tenantId**: tenant for which the key is authenticated.
- **strSubscrRecId**: the Subscription ID corresponding to the Request Offering for the current user

Return Value:

An FRSHEATGetPackageDataResponse object, defined as follows:

```
public class FRSHEATGetPackageDataResponse
{
    public string status { get; set; }
    public string exceptionReason { get; set; }
    public FRSHEATServiceReqSubscription srSubscription { get; set; }
}
```

The FRSHEATGetPackageDataResponse class comprises the following fields:

- **status** – this field provides a Status value indicating whether the operation was successful.
  A full description of the available Status values is provided in the table below.
- **exceptionReason** – if there is an exception thrown in the course of running the Connect WebMethod, the exception information will be captured in this field.
- **srSubscription** – an FRSHEATServiceReqSubscription object, containing the full details of the Request Offering (such as the parameter details)

The FRSHEATServiceReqSubscription class is defined as follows:

```
public class FRSHEATServiceReqSubscription
{
    public string strSubscriptionRecId;
    public string strDateSubscribed;
    public string strOrgUnitId;
    public string strName;
    public string strDescription;
    public string strServiceName;
    public string strRecId;
    public List<FRSHEATServiceReqTemplateParam> lstParameters;
}
```

The FRSHEATServiceReqTemplateListItem class comprises the following fields:

- **strSubscriptionRecId** – the Subscription ID corresponding to the Request Offering for the current user
- **strDateSubscribed** – date when the Request Offering was marked as Published and Subscribed
- **strOrgUnitId** – RecId of the OrganizationalUnit the Request Offering is associated with
- **strName** – the name of the Request Offering
- **strDescription** – description value of the Request Offering
- **strServiceName** – name of the Service the Request Offering is associated with
- **strRecId** – the RecId of the Request Offering
- **lstParameters** – a list of FRSHEATServiceReqTemplateParam objects, each item representing a parameter in the Request Offering

The FRSHEATServiceReqTemplateParam class is defined as follows:

```
public class FRSHEATServiceReqTemplateParam
{
    public string strName;
    public string strLabel;
    public string strDescription;
    public string strType;
    public bool bAllowSelectByUser;
    public bool bDBValidated;
    public string strRequiredExpression;
    public bool isCalculated;
    public bool autoFillOnlyWhenEmpty;
```

```
    public string strValidationListRecId;
    public string strValidationListTableRef;
    public bool bIsHidden;
    public string strRecId;
    public List<ValidationConstraint> validationConstraints;
    public string strValidationConstraints;
    public string strAutoFillExpression;
    public List<string> triggerFields;
    public string strTriggerFields;
    public string strDefaultValue;
}
```

The FRSHEATServiceReqTemplateListItem class comprises the following fields:

- **strName** – name of the parameter
- **strLabel** – label of the parameter
- **strDescription** – description of the parameter
- **strType** – type of the parameter
- **bAllowSelectByUser** – flag indicating whether the parameter can be selected by the user
- **bDBValidated** – flag indicating whether the parameter is validated
- **strRequiredExpression** – expression indicating how the parameter is (conditionally) required
- **isCalculated** – flag indicating whether the parameter is calculated (i.e. does not take inputs from users)
- **autoFillOnlyWhenEmpty** – flag indicating whether the autofill for the parameter occurs, when the parameter is initially empty
- **strValidationListRecId** – RecId of the named validation list (i.e. pick list), if the parameter is a Dropdown Selection
- **strValidationListTableRef** – the name of the Business Object, from which the dropdown values are retrieved, for the Dropdown Selection
- **bIsHidden** – flag indicating whether the parameter is hidden
- **strRecId** – RecId of the Request Offering parameter
- **validationConstraints** – list of ValidationConstraint objects, used for representing the validation constraints currently set on the parameter
- **strValidationConstraints** – textual representation of the validationConstraints member
- **strAutoFillExpression** – textual representation of the autofill expression
- **triggerFields** – list of dependent parameters, upon which the current parameter is dependent upon
- **strTriggerFields** – textual representation of the triggerFields member
- **strDefaultValue** – textual representation of the default value for the parameter

The following table lists the available status values, and describes how to interpret them.

Status	Explanation
Success	The details of the Request Offering can be successfully retrieved, and can be accessed via the srSubscription member
NotFound	There is no Request Offering available with the given Subscription Id in the tenant – please ensure that the ID of of the Request Offering is specified correctly. Recall that the GetSubscriptionId Web Method is used for fetching the Subscription ID corresponding to the current user
Error	An Error was encountered during the execution of this Web Method - the corresponding exceptionReason member should be inspected, to determine why the web method has failed.

Example

FRSHEATGetPackageDataResponse getPackageDataResponse = frSvc.GetPackageData(authSessionKey, tenantId, strSubscrRecId);

FRSHEATServiceReqSubscription scsrs;

if (getPackageDataResponse.status == "Success")

{

scsrs = getPackageDataResponse.srSubscription;

// Go ahead and access the relevant properties of interest from the Request Offering ..

}

UserCanAccessRequestOffering

Check whether the current user is entitled to access the given Request Offering

Request syntax:

```Bash
public FRSHEATUserCanAccessReqOfferingResponse UserCanAccessRequestOffering(string sessionKey, string tenantId, string loginId, st
```

Parameters:

- **sessionKey**: Key received in the earlier Connect request
- **tenantId**: tenant for which the key is authenticated.
- **loginId**: loginId of the user
- **reqOfferingName**: name of the Request Offering

Return Value:

An FRSHEATUserCanAccessReqOfferingResponse object, defined as follows:

public class FRSHEATUserCanAccessReqOfferingResponse

```
{
    public string status { get; set; }
    public string exceptionReason { get; set; }
    public bool canAccess { get; set; }
}
```

The FRSHEATUserCanAccessReqOfferingResponse class comprises the following fields:

- **status** – this field provides a Status value indicating whether the operation was successful.
  A full description of the available Status values is provided in the table below.
- **exceptionReason** – if there is an exception thrown in the course of running the Connect WebMethod, the exception information will be captured in this field.
- **canAccess** – boolean field, indicating whether the user is entitled to access the given Request Offering

The following table lists the available status values, and describes how to interpret them.

Status	Explanation
Success	The indicated user and Request Offering can be successfully located in the tenant - the canAccess field can then be checked, to determine whether the indicated user is entitled to access the Request Offering
UserNotFound	No user record can be located in the tenant, with the given LoginID value. Check the LoginID value that was passed in, to ensure that it is spelled correctly
OfferingNotFound	No Request Offering can be found with the given name. Check the name of the Request Offering that is passed in, to ensure that it is spelled correctly
Error	An Error was encountered during the execution of this Web Method - the corresponding exceptionReason member should be inspected, to determine why the web method has failed.

Example

```
        string loginId = "ASimon";
        string reqOfferingName = "Domain Password Reset";
        FRSHEATUserCanAccessReqOfferingResponse canAccessReqOfferingResponse =
frSvc.UserCanAccessRequestOffering(sessionKey, tenantId, loginId, reqOfferingName);
        if (canAccessReqOfferingResponse.status == "Success")
        {
            if (canAccessReqOfferingResponse.canAccess)
            {
                Console.WriteLine("User  can access the Request Offering ", loginId, reqOfferingName);
            }
```

```
        else
        {
            Console.WriteLine("User  has no access to the Request Offering ", loginId, reqOfferingName);
        }
    }
```

SubmitRequest

Creates and submits the Service Request, on behalf of a given user

Request syntax:

```
Bash
public FRSHEATSubmitRequestResponse SubmitRequest(string sessionKey, string tenantId, string subscriptionId, List<FRSHEATServiceRe
```

Parameters:

- **sessionKey**: Key received in the earlier Connect request
- **tenantId**: tenant for which the key is authenticated
- **subscriptionId**: the Subscription ID corresponding to the Request Offering for the current user
- **srparameters**: list of FRSHEATServiceReqParam objects, representing the key / values for the parameters, for submission for the Service Request
- **loginId**: loginId of the user

An FRSHEATServiceReqParam class is defined as follows:

```
public class FRSHEATServiceReqParam
{
    public string strName;
    public string strValue;
}
```

The FRSHEATServiceReqParam class comprises the following fields:

strName: name of the parameter

strValue: value for the parameter

Return Value:

An FRSHEATSubmitRequestResponse object, defined as follows:

```
public class FRSHEATSubmitRequestResponse
{
```

```
    public string status { get; set; }

    public string exceptionReason { get; set; }

    public FRSHEATServiceReqRequest reqData { get; set; }

}
```

The FRSHEATSubmitRequestResponse class comprises the following fields:

- **status** – this field provides a Status value indicating whether the operation was successful.
  A full description of the available Status values is provided in the table below.
- **exceptionReason** – if there is an exception thrown in the course of running the Connect WebMethod, the exception information will be captured in this field.
- **reqData** – an FRSHEATServiceReqRequest object, containing the Service Request that is successfully submitted

The FRSHEATServiceReqRequest class is defined as follows:

```
public class FRSHEATServiceReqRequest

{

    public string strRequestRecId;

    public string strRequestNum;

    public decimal fTotalPrice;

    public decimal fTotalRecurPrice;

    public string strByEmployee;

    public string strFulfillmentPlanType;

    public string strUrgency;

    public DateTime? DateSubmitted;

    public DateTime? DateUpdated;

    public string strUpdatedBy;

    public DateTime? DateDelivery;

    public string strSubscriptionRecId;

    public bool isFulfilled;

    public string strName;

    public string strDescription;

    public decimal fPrice;

    public decimal fRecurPrice;

    public int nRecurPeriod;

    public int nDeliveryCommitment;
```

```
    public string strStatus;

    public string strCreatedBy;

    public string strRecId;

    public List<FRSHEATServiceReqTemplateParam> lstParameters;

}
```

The FRSHEATServiceReqRequest class comprises the following fields:

- **strRequestRecId** – RecId of the Service Request
- **strRequestNum** – ID of the Service Request
- **fTotalPrice** – total non-recurring price on the Service Request
- **fTotalRecurPrice** – total recurring price on the Service Request
- **strByEmployee** - LoginId of the requestor of the Service Request
- **strFulfillmentPlanType** – type of fulfillment
- **strUrgency** – urgency value on the Service Request
- **DateSubmitted** – date when the Service Request was submitted
- **DateUpdated** - date when the Service Request was last updated
- **strUpdatedBy** – LoginId of the user who last updated the Service Request
- **DateDelivery** – delivery date for the Service Request
- **strSubscriptionRecId** – RecId of the Subscription corresponding to the Service Request
- **isFulfilled** – flag indicating whether the Service Request has been fulfilled or not
- **strName** – name of the Request Offering, from which the Service Request was created from
- **strDescription** – description of the Service Request
- **fPrice** – Price on the Service Request
- **fRecurPrice** – Recurring Price on the Service Request
- **nRecurPeriod** – recurrence period for the Service Request
- **nDeliveryCommitment** – delivery commitment period for the Service Request
- **strStatus** – Status of the Service Request
- **strCreatedBy** - LoginId of the user who created the Service Request
- **strRecId** – RecId of the Service Request
- **lstParameters** – list of FRSHEATServiceReqTemplateParam objects, where each item in the list represents the parameter value for the Service Request

The following table lists the available status values, and describes how to interpret them.

Status	Explanation
Success	The list of matching Request Offerings in the Service Catalog can be successfully retrieved, and are available via the srtList member.
UserNotFound	No user recordcan be located in the tenant, with the given LoginID value. Check the LoginID value that was passed in, to ensure that it is spelled correctly
ReqParamsNotFilled	One or more required parameters are not filled out. Check the exceptionReason member ,to determine which specific required parameters were not filled out properly
ValidationValuesMismatch	The value that was passed in for a drop-down selection parameter, does not belong in the validation list. For example, assume there is a drop-down selection parameter which is validated against the Departments validation list. If a value of “Billing” is supplied to the SubmitRequest WebMethod, but “Billing” does not exist in the validation list, the return Status in this case will be “ValidationValuesMismatch”. Check the exceptionReason member, to determine which specific validated parameters did not have the proper values filled out for it
ParameterTypeMismatch	One or more parameters were specified with values, which are inappropriate for the underlying parameter data type. For example, the Text Field parameter type can be used to store various types of values. If the data type of the parameter is Number, and a text value is specified for the parameter, the return status in this case will be “ParameterTypeMismatch”. Check the exceptionReason member, to determine which specific parameters were specified with values with the incorrect type.
Error	An Error was encountered during the execution of this Web Method - the corresponding exceptionReason member should be inspected, to determine why the web method has failed.

Service Request Submission Validation

Before submitting the Service Request, the SubmitRequest Web Method performs three types of validation:

- **Required Parameters** – ensures that all parameters marked as required are filled out. Currently the code will handle unconditional required parameters, as well as simple conditional required parameters.
  If there are one or more required parameters which are not filled out properly, the SubmitRequest WebMethod will return a status value of “ReqParamsNotFilled”. In such cases, the exceptionReason member can be consulted, for determining which specific parameter values have not been filled out.
- **Validation Values** – ensures that the supplied values are allowable values, based on the existing records in the validation list table.
  Currently the code will handle regular validation (i.e. validation against a single field) as well as simple constrained validation (i.e. validation which is constrained, based on the value of an earlier field).
  If there are one or more parameters which do not satisfy the validation, the SubmitRequest WebMethod will return a status value of “ValidationValuesMismatch”. In such cases, the exceptionReason member can be consulted, for determining which specific parameter values do not match up with the validation list.

- **Parameter Type Checking** – ensures that the supplied value is appropriate with respect to the data type of the given parameter.
  If there are one or more values which do not match the corresponding parameter type, the SubmitRequest WebMethod will return a status value of "ParameterTypeMismatch". In such cases, the exceptionReason member can be consulted, for determining which specific parameter values do not match the corresponding data type.
  This will be defined in detailed in the next section.

Type Checking of Parameters in the Service Request

Prior to submitting the Service Request, the SubmitRequest Web Method will perform the data type checking of the values, with respect to the data types of the respective parameters.

The following table summarizes the various parameter types. If data type validation is performed, the details section will describe this information in greater detail.

Data Type	Details
Text	Currently not validated (matches the current Service Catalog behavior)
CheckBox	Allowed values are those which can be parsed as either true or false. Uses the Boolean.TryParse() method in .NET for determining if the value for checkbox is valid or not.
Date / Time / DateTime	Allowed values are those which can be parsed as date/time values. Uses the DateTime.TryParse() method in .NET for determining if the value for date / time / datetime is valid or not. To specify date/time values, the string value should be specified using ISO 8601 format, and the value itself should be relative to UTC. So the date/time value can be specified in one of the following two ways: or Either a space character or "T" character can be used to separate between the date and time values. The following are two examples of specifying a date/time value of March 26th, 2013, 18:38 UTC, relative to the above two formats: 2013-03-26 18:38 2013-03-26T18:38
Number / Money	Allowed values are those which can be parsed as decimal values. Uses the Decimal.TryParse() method in .NET for determining if the numerical value is valid or not.
Phone	Allowed values are phone numbers which match a US phone number; e.g. (415)555-1212 This matches the current Service Catalog behavior. Do not use this parameter type for validating International phone numbers.
SSN	Allowed values are US Social Security Numbers; e.g. 123-45-6789
Email	Currently not validated (matches the current Service Catalog behavior)
URL	Currently not validated (matches the current Service Catalog behavior)

Bash

```
yyyy-mm-ddThh:mm
```

Bash

```
yyyy-mm-dd  hh:mm
```

Example

```
string strSubscrRecId = frSvc.GetSubscriptionId(authSessionKey, tenantId, "Domain Password Reset").subscriptionId;
List<FRSHEATServiceReqParam> srparamListItems = new List<FRSHEATServiceReqParam>();
srparamListItems.Add(new FRSHEATServiceReqParam
{
   strName = "RequesterDepartment",
   strValue = "IT"
});
srparamListItems.Add(new FRSHEATServiceReqParam
{
   strName = "Requester",
   strValue = "Ashley Simon"
});
FRSHEATSubmitRequestResponse srResponse = frSvc.SubmitRequest(authSessionKey, tenantId, strSubscrRecId, srparamListItems.ToArray(), loginId);
FRSHEATServiceReqRequest srres;
if (srResponse.status == "Success")
{
   srres = srResponse.reqData;
}
else
{
   if (srResponse.exceptionReason != null)
   {
      Console.WriteLine("Error: " + srResponse.exceptionReason);
```

```
        }
    }
```

GetRequestData

Retrieves the data for the given Service Request

Request syntax:

public FRSHEATGetRequestDataResponse GetRequestData(string sessionKey, string tenantId, string strReqNumber)

Parameters:

sessionKey: Key received in the earlier Connect request

tenantId: tenant for which the key is authenticated.

strReqNumber: numeric ID value for the Service Request

Return Value:

An FRSHEATGetRequestDataResponse object, defined as follows:

```
public class FRSHEATGetRequestDataResponse
{
    public string status { get; set; }
    public string exceptionReason { get; set; }
    public FRSHEATServiceReqRequest reqData { get; set; }
}
```

The FRSHEATGetRequestDataResponse class comprises the following fields:

- **status** – this field provides a Status value indicating whether the operation was successful.
  A full description of the available Status values is provided in the table below.
- **exceptionReason** – if there is an exception thrown in the course of running the Connect WebMethod, the exception information will be captured in this field.
- **reqData** – an FRSHEATServiceReqRequest object, containing the Service Request that is successfully submitted

Refer to the earlier SubmitRequest section, regarding the definition of the FRSHEATServiceReqRequest class.

The following table lists the available status values, and describes how to interpret them.

Status	Explanation
Success	The Service Request with the given numeric ID can be located, and is available via the reqData member.
NotFound	The Service Request with the given numeric ID cannot be located. Confirm that the numeric ID of the Service Request was specified correctly .
Error	An Error was encountered during the execution of this Web Method - the corresponding exceptionReason member should be inspected, to determine why the web method has failed.

Example

FRSHEATGetRequestDataResponse getRequestDataResponse = frSvc.GetRequestData(authSessionKey, tenantId, strReqRecId);

FRSHEATServiceReqRequest srRequest;

if (getRequestDataResponse.status == "Success")

{

srRequest = getRequestDataResponse.reqData;

// Go ahead and access the relevant properties of interest from the Service Request ..

}

FetchServiceReqValidationListData

Fetch the allowable validation list data for the given Service Request parameter

Request syntax:

```Bash
public FRSHEATFetchSRValListDataResponse FetchServiceReqValidationListData(string sessionKey, string tenantId, string offeringName
```

Parameters:

- **sessionKey**: Key received in the earlier Connect request
- **tenantId**: tenant for which the key is authenticated.
- **offeringName**: Name of the Request Offering
- **paramName**: Name of the parameter in the Request Offering
- **depValItem**: an FRSHEATDepValItem object, representing the dependent validation item
- **substrMatch**: substring to match against the returned validation list items

Return Value:

An FRSHEATFetchSRValListDataResponse object, defined as follows:

```
public class FRSHEATFetchSRValListDataResponse
{
    public string status { get; set; }
    public string exceptionReason { get; set; }
    public List<FRSHEATValListValue> validationValuesList { get; set; }
}
```

The FRSHEATFetchSRValListDataResponse class comprises the following fields:

- **status** – this field provides a Status value indicating whether the operation was successful.
  A full description of the available Status values is provided in the table below.
- **exceptionReason** – if there is an exception thrown in the course of running the Connect WebMethod, the exception information will be captured in this field.
- **validationValuesList** – a List of FRSHEATValListValue objects, each item representing the records returned from the underlying validation list, based on the specified search criteria

The FRSHEATValListValue class is defined as follows:

```
public class FRSHEATValListValue
{
    public string strRecId;
    public string strStoredValue;
    public string strDisplayValue;
}
```

The FRSHEATValListValue class comprises the following fields:

- **strRecId** – RecId of the validation list record
- **strStoredValue** – the stored value of the validation list record (e.g. “ASimon”, if the validation list record corresponds to the “Ashley Simon” Employee record)
- **strDisplayValue** - the display value of the validation list record (e.g. “Ashley Simon”, if the validation list record corresponds to the “Ashley Simon” Employee record)

The following table lists the available status values, and describes how to interpret them.

Status	Explanation
Success	The Service Request with the given numeric ID can be located, and is available via the reqData member.
NotFound	The Service Request with the given numeric ID cannot be located. Confirm that the numeric ID of the Service Request was specified correctly .
Error	An Error was encountered during the execution of this Web Method - the corresponding exceptionReason member should be inspected, to determine why the web method has failed.

Example

```
// Fetch the matching validation list records for Requester, in the "Domain Password Reset" Request Offering

// Since the Requester is constrained by the "RequesterDeparment" parameter, this needs to be specified as an input parameter to the Web Method

string offeringName = "Domain Password Reset";

string paramName = "Requester";

FRSHEATDepValItem depValItem = new FRSHEATDepValItem()

{

   strParName = "RequesterDepartment",

   strParValue = "IT"

};

FRSHEATFetchSRValListDataResponse validationValuesResponse = frSvc.FetchServiceReqValidationListData(authSessionKey, tenantId, offeringName, paramName, depValItem, subStrQuery);

FRSHEATValListValue[] valListValues;

if (validationValuesResponse.status == "Success")

{

   valListValues = validationValuesResponse.validationValuesList;

   Console.WriteLine("Here are the matching validation list records:\n");

   foreach (FRSHEATValListValue valListItem in valListValues)

   {

      Console.WriteLine("Stored Value: \"\"\t\tDisplay Value: \"\"", valListItem.strStoredValue, valListItem.strDisplayValue);

   }

}
```

Appendix A: Creating a Test Console Application using Visual Studio

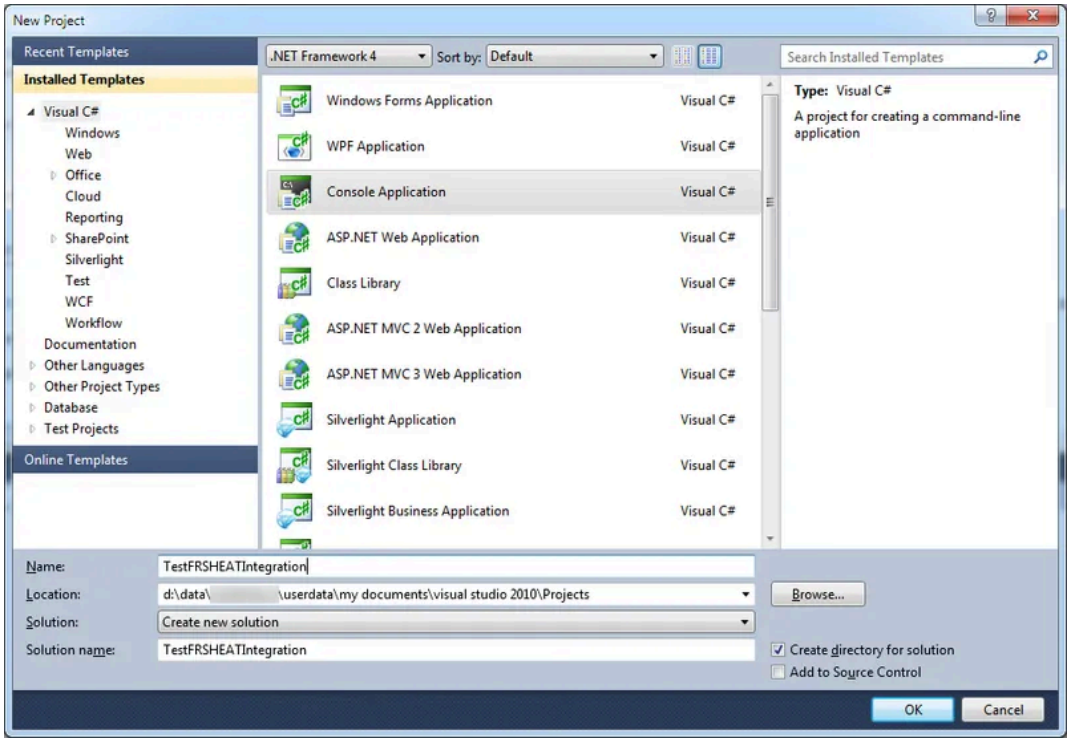
As explained in the earlier Introduction section, the FRSHEATIntegration Web Service uses standard Web Service technologies, and can be consumed using various programming languages / platforms such as Microsoft .NET, Java, etc.

This section describes how one would create a test console application using Visual Studio 2010, specifically for creating the Web Reference to access the SOAP-based FRSHEATIntegration API.

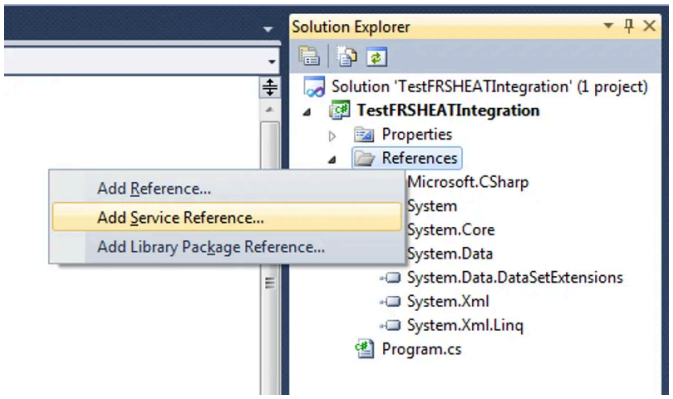
Note that alternative .NET based clients such as WPF, ASP.NET, and Windows Forms clients can also be created, using the same concepts as presented here.

To create a new test console application in Visual Studio, follow these steps:

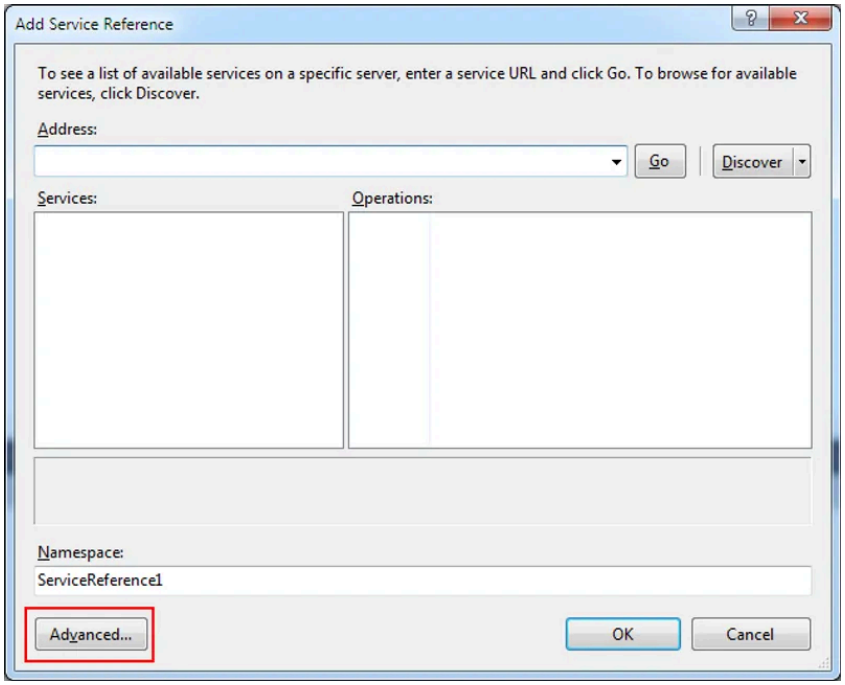
1. Launch Visual Studio 2010, and then create a new Visual C# Console Project.



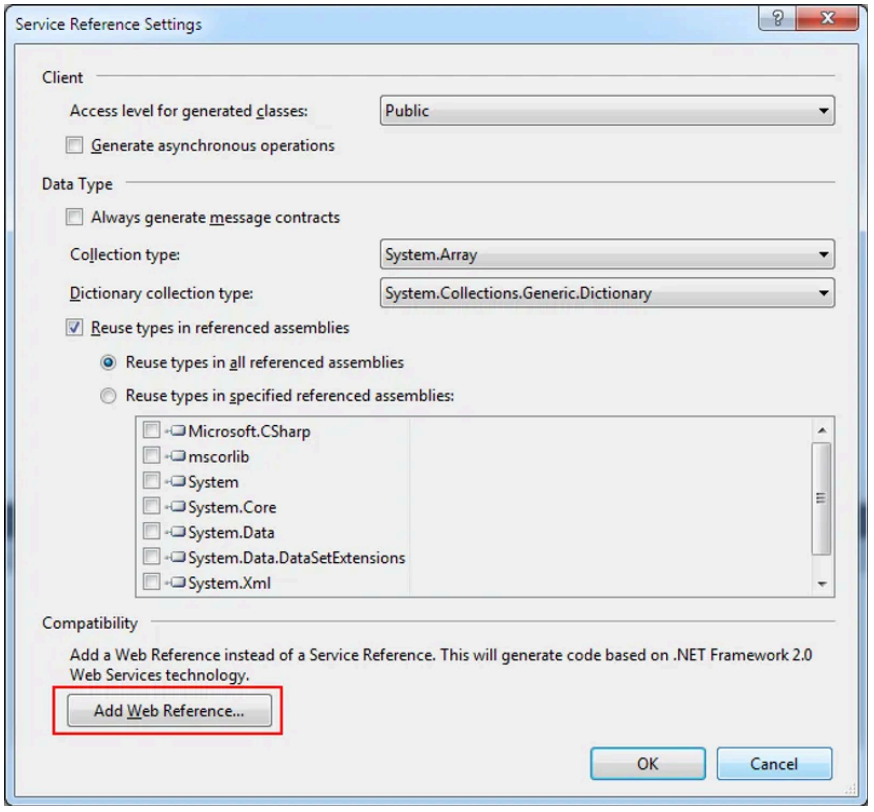
2. In the Solution Explorer, right click the References folder, then select **Add Service Reference** from the context menu.



The Add Service Reference dialog box appears.



3. Click **Advanced** in the lower left hand corner, to launch the Service Reference Settings dialog box.



4. Click **Add Web Reference** in the lower left hand corner, to launch the Add Web Reference dialog box.

   At this point, you are now ready to author the necessary code, for exercising the FRSHEATIntegration Web Service.

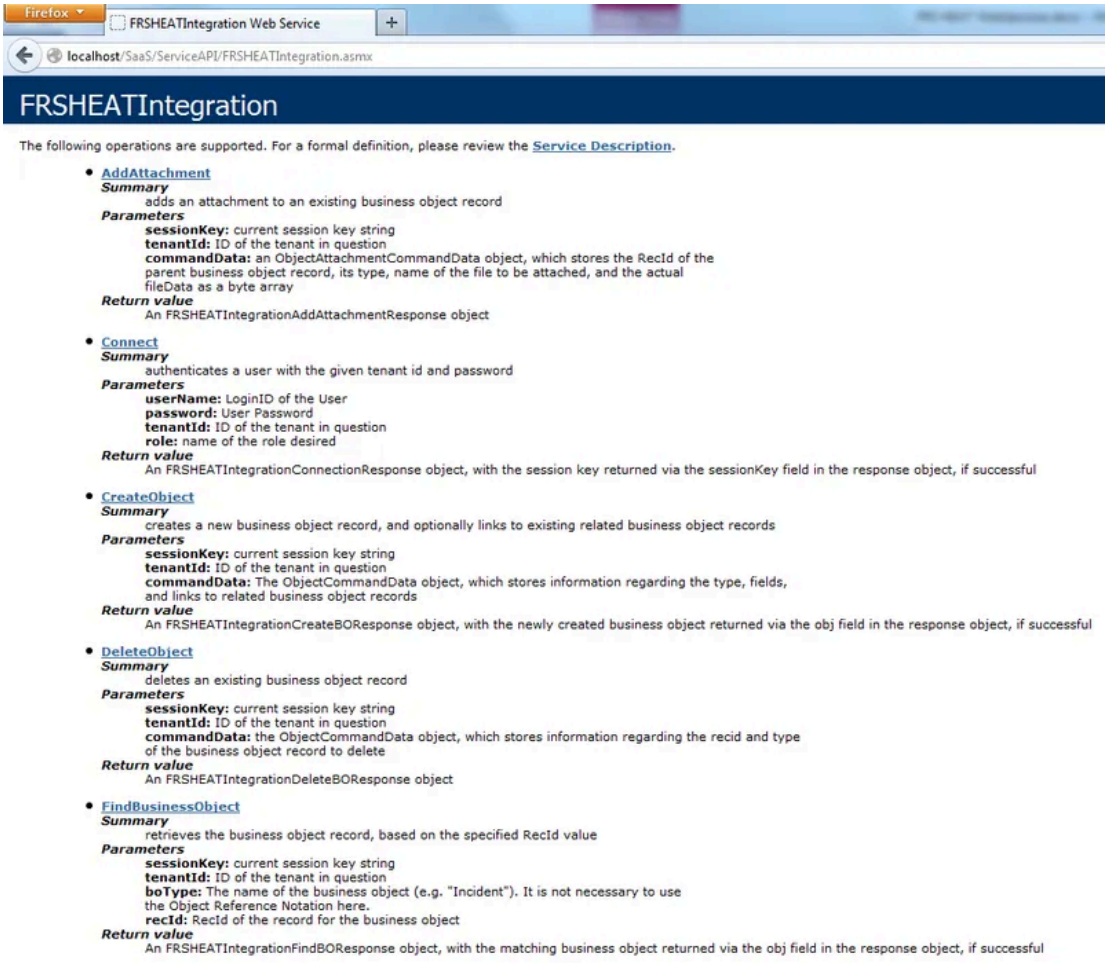
9.5. FRSHEATIntegration Web Service URL

The FRSHEATIntegration Web Service can be accessed at a link similar to the example shown below:

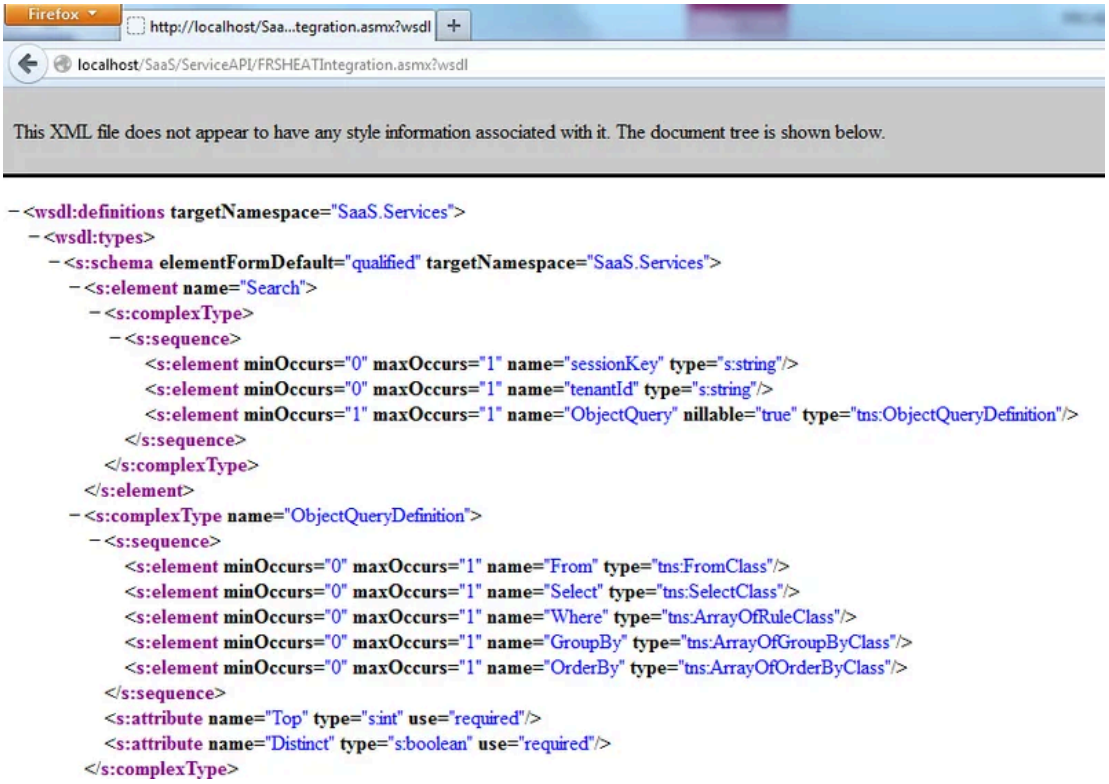
https://<TenantName>/ServiceAPI/FRSHEATIntegration.asmx

Replace *<TenantName>* with the hostname corresponding to your particular tenant.

Notice when accessing the above URL, information is provided for the available Web Methods.



The corresponding WSDL file can be accessed at by adding “?wsdl” at the end of the previous URL, as seen in the example link shown below:https://<TenantName>/ServiceAPI/FRSHEATIntegration.asmx?wsdl



Establishing the Connection and Role Selection

Before an application can access a Web Service API on the SaaS platform, it has to be properly authenticated and authorized, with respect to the tenant. This is achieved by first invoking the Connect() WebMethod, before performing any subsequent Web Service operations.

Connect

The Connect WebMethod is responsible for performing both the authentication and authorization operations, to ensure that the Web Service user is properly authenticated, and belongs to the specified role.

Request syntax:

FRSHEATIntegrationConnectionResponse Connect(string userName, string password, string tenantId, string role)

Parameters:

userName: loginId for which the session is to be created. The loginId is searched against LoginID column in the Profile.Employee business object (Profile table).

password: user password. Either internal or external (Active Directory) password can be used, depending on the Employee record configuration.

tenantId: tenant for which the session is to be created. The tenant ID is matched against the “TenantURL” field, in the Tenants business object in the ConfigDB database. The tenant record must be in “Active” state in order for authentication to succeed.

role: the role that the indicated user will be logging in as.

Return Value:

An FRSHEATIntegrationConnectionResponse object, defined as follows:

```
public class FRSHEATIntegrationConnectionResponse
{
    public string connectionStatus;
    public string sessionKey;
    public string exceptionReason;
}
```

The FRSHEATIntegrationConnectionResponse class comprises the following fields:

- **connectionStatus** – this field provides a Status value indicating the state of the Connection.
  A full description of the available Status values is provided in the table below.
- **sessionKey** – the sessionKey which needs to be used in all subsequent Web Service operations. This field will only be populated when the connectionStatus has a value of "Success", otherwise the value will be null.
- **exceptionReason** – if there is an exception thrown in the course of running the Connect WebMethod, the exception information will be captured in this field.

If the Connect WebMethod operation completes successfully, the sessionKey will be stored in the corresponding member in the FRSHEATIntegrationConnectionResponse object.

However, if there is an error encountered during the Connect operation (either during the authentication or authorization phases), a SOAP Exception will be thrown by the server, and the Web Services Client will be required to handle the exception accordingly.

The following table lists the possible kinds of exceptions which can be encountered, while executing the Connect WebMethod.

SOAP Exception	Explanation
AccessDenied	Either the specified username does not exist in the tenant, or the password provided is invalid for the username. Double-check the credentials passed into the Connect Web Method, to ensure that they are specified properly.
TenantNotFound	Either the specified TenantURL cannot be found, or that the tenant is not currently in “Active” status. Double-check the TenantURL to ensure that it is the correct URL for accessing the tenant. If the TenantURL is correct, please double check with FRS Operations regarding the status of the given Tenant.
TenantInMaintenance	The specified Tenant is currently in maintenance mode. Confirm whether the Tenant is in maintenance mode, and double-check with FRS Operations, as to when the tenant will be set back to Active status.
InvalidRole	Either the specified Role definition is not available in the tenant, or that the user is not currently associated with the specified role. Confirm whether the Role actually does exist in the tenant, and that the user does belong to the given role.
SessionExpired	The SessionKey refers to a session which has since expired. This typically occurs when the API operation uses the SessionKey from an earlier Connect webmethod call, and the session has expired due to inactivity. Whenever this occurs, it is necessary to execute the Connect webmethod call again, so that a new sessionkey can be obtained, for performing any further actions. This will be explained in the next section.

Important Note: For all subsequent WebMethods described in this document, the user may run into authentication / authorization errors, when reusing the same sessionKey value – the Web Services client will need to account for the possibility of connection failures, while exercising the remaining WebMethods described in this document.

Example

```
FRSHEATIntegration frSvc = new FRSHEATIntegration();

FRSHEATIntegrationConnectionResponse connectresponse = frSvc.Connect(username, password, tenantId, role);

if (connectresponse.connectionStatus == "Success")

{

    ...

}
```

Handling Session Expirations

After making the initial call to the Connect WebMethod to obtain the sessionkey, typically Web Service Clients would reuse the same sessionKey value for invoking all subsequent WebMethod operations, such as Search, UpdateObject, etc.

If the Web Services Client remains idle for a period of time, which exceeds the session timeout value that has been set for the tenant, the session will expire - the sessionkey that was obtained earlier will no longer be valid.

Once the session has expired, if the client tries to use the expired sessionkey for any subsequent operation (such as Search), a SOAP Exception will be returned by the server, alerting the client that the session has indeed expired.

By throwing the explicit SOAP Exception back to the client, the client can then use standard exception handling techniques, for calling the Connect WebMethod again, to obtain a brand new sessionkey, which can then be used in subsequent operations.

For .NET-based web services clients, the reference to the FRSHEATIntegration API can be established using either a Web Reference or a Service Reference – the code to handle the SOAP exception differs slightly based on the type of reference.

The following code sample illustrates how this can be achieved, if the Web Services Client is defined using a Web Reference to the FRSHEATIntegration API:

```
ObjectQueryDefinition query = new ObjectQueryDefinition();
query.Select = new SelectClass();
FieldClass[] incidentFieldObjects = new FieldClass[] {
   new FieldClass()
   {
      Name = "IncidentNumber",
      Type = "Text"
   },
   new FieldClass()
   {
      Name = "Service",
      Type = "Text"
   }
};
query.Select.Fields = incidentFieldObjects;
query.From = new FromClass();
query.From.Object = "Incident";
query.Where = new RuleClass[] {
   new RuleClass()
   {
      Join = "AND",
      Condition = "=",
      Field = "IncidentNumber",
      Value = "10001"
```

```
    }
};
FRSHEATIntegrationSearchResponse searchResponse = null;
try
{
  searchResponse = frSvc.Search(authSessionKey, tenantId, query);
}
catch (SoapException soapException)
{
  if (soapException.Actor == "SessionExpired")
  {
     connectresponse = frSvc.Connect(username, password, tenantId, role);
     if (connectresponse.connectionStatus == "Success")
     {
        authSessionKey = connectresponse.sessionKey;
        searchResponse = frSvc.Search(connectresponse.sessionKey, tenantId, query);
     }
  }
}
if (searchResponse != null && searchResponse.status == "Success")
{
   WebServiceBusinessObject[][] incidentList = searchResponse.objList;
   foreach (WebServiceBusinessObject[] incidentOuterList in incidentList)
   {
      foreach (WebServiceBusinessObject incident in incidentOuterList)
      {
         Console.WriteLine("Incident  matches the selection criteria", incident.FieldValues[0].Value);
      }
   }
}
```

In particular, note the following code fragment:

```
try
{
  searchResponse = frSvc.Search(authSessionKey, tenantId, query);
}
catch (SoapException soapException)
{
  if (soapException.Actor == "SessionExpired")
  {
    connectresponse = frSvc.Connect(username, password, tenantId, role);
    if (connectresponse.connectionStatus == "Success")
    {
      authSessionKey = connectresponse.sessionKey;
      searchResponse = frSvc.Search(authSessionKey, tenantId, query);
    }
  }
}
```

Notice that the Search WebMethod operation is wrapped inside the try-catch block.

If the SOAP Exception occurs corresponding to the session expiration, the Actor property in the SoapException can be inspected, to determine whether the connection error is due to the session timeout.

If the exception is due to the session timeout, it then goes ahead and invokes the Connect WebMethod again, to obtain a brand new sessionkey.

Assuming that authSessionKey is a globally accessible string, it is now updated to the sessionkey property of the connectresponse object, and the Search WebMethod can be successfully run the second time.

The following code sample now illustrates how this can be achieved, if the Web Services Client is defined using a Service Reference to the FRSHEATIntegration API:

```
ObjectQueryDefinition query = new ObjectQueryDefinition();
query.Select = new SelectClass();
FieldClass[] incidentFieldObjects = new FieldClass[] {
   new FieldClass()
   {
     Name = "IncidentNumber",
     Type = "Text"
```

```
    },
    new FieldClass()
    {
        Name = "Service",
        Type = "Text"
    }
};
query.Select.Fields = incidentFieldObjects;
query.From = new FromClass();
query.From.Object = "Incident";
query.Where = new RuleClass[] {
    new RuleClass()
    {
        Join = "AND",
        Condition = "=",
        Field = "IncidentNumber",
        Value = "10001"
    }
};
FRSHEATIntegrationSearchResponse searchResponse = null;
try
{
    searchResponse = frSvc.Search(authSessionKey, tenantId, query);
}
catch (FaultException faultException)
{
    MessageFault messageFault = faultException.CreateMessageFault();
    if (messageFault.Actor == "SessionExpired")
    {
        connectresponse = frSvc.Connect(username, password, tenantId, role);
        if (connectresponse.connectionStatus == "Success")
```

```
        {
            authSessionKey = connectresponse.sessionKey;
            searchResponse = frSvc.Search(connectresponse.sessionKey, tenantId, query);
        }
    }
}
if (searchResponse != null && searchResponse.status == "Success")
{
    WebServiceBusinessObject[][] incidentList = searchResponse.objList;
    foreach (WebServiceBusinessObject[] incidentOuterList in incidentList)
    {
        foreach (WebServiceBusinessObject incident in incidentOuterList)
        {
            Console.WriteLine("Incident  matches the selection criteria", incident.FieldValues[0].Value);
        }
    }
}
```

Bash

In particular, note the following code fragment:

```
try
{
    searchResponse = frSvc.Search(authSessionKey, tenantId, query);
}
catch (FaultException faultException)
{
    MessageFault messageFault = faultException.CreateMessageFault();
    if (messageFault.Actor == "SessionExpired")
    {
        connectresponse = frSvc.Connect(username, password, tenantId, role);
        if (connectresponse.connectionStatus == "Success")
```

```
            {
                authSessionKey = connectresponse.sessionKey;
                searchResponse = frSvc.Search(connectresponse.sessionKey, tenantId, query);
            }
        }
    }
```

Again, notice that the Search WebMethod operation is wrapped inside the try-catch block.

However, for Service References, the exception that is caught is now a FaultException, and not the SOAP exception (as was the case with the earlier Web Reference).

Here, once the faultException is caught, it is first necessary to obtain the MessageFault from it.

From the MessageFault, the Actor property can then be the Actor property can then be inspected, to determine whether the connection error is due to the session timeout.

The reconnect logic is then the same as the case for Web Reference based clients, as explained earlier.

The above example illustrates how to properly handle session expirations. The same technique can be used for handling any other kinds of exceptions, such as AccessDenied, InvalidRole, etc.

For example, for the “AccessDenied” exception, the Web Services client might log the error locally to a log file, or send an email to the administrator, alerting that the username / password is not valid, when used in the client.

GetRolesForUser

This Web Method retrieves a list of roles, which are available to the current Web Service user.

The response object returned from this WebMethod, contains a field which stores an array of Roles which are associated with the Profile.Employee record of the current user.

Request Syntax:

FRSHEATIntegrationGetRolesResponse GetRolesForUser(string sessionKey, string tenantId)

NameDisplayPair[] GetRolesForUser(string sessionKey, string tenantId, string userName)

Parameters:

- **sessionKey**: Key received in the earlier Connect request.
- **tenantid**: ID of the tenant in question.

Return Value:

An FRSHEATIntegrationGetRolesResponse object, defined as follows:

```
public class FRSHEATIntegrationGetRolesResponse
{
    public string status { get; set; }
```

```
    public List<NameDisplayPair> roleList { get; set; }

    public string exceptionReason { get; set; }

}
```

The **FRSHEATIntegrationGetRolesResponse** class is comprises the following fields:

- **Status** – this field provides a Status value, indicating whether the WebMethod was able to successfully retrieve the list of roles for the current user.
  Assuming the sessionKey (returned from the previous Connect WebMethod call) is valid, this field should return a value of "Success" under most circumstances.
- **roleList** – contains an array of NameDisplayPair objects. Each NameDisplayPair entry corresponds to a role which the current user belongs to.
  For each NameDisplayPair record in the array, the Name property returns the name of the role, whereas the DisplayName property returns the DisplayName of the role.
- **exceptionReason** – if there is an exception thrown in the course of running the GetRolesForUser WebMethod, the exception information will be captured in this field.

Example

```
FRSHEATIntegrationGetRolesResponse getRolesResponse = frSvc.GetRolesForUser(authSessionKey, tenantId);

NameDisplayPair[] rolePairList = null;

if (getRolesResponse.status == "Success")

{

        rolePairList = getRolesResponse.roleList;

        Console.WriteLine("The current user belongs to the following roles:");

        foreach (NameDisplayPair rolePair in rolePairList)

        {

           Console.WriteLine("\tRole:  ⇒ DisplayName: \"\"", rolePair.Name, rolePair.DisplayName);

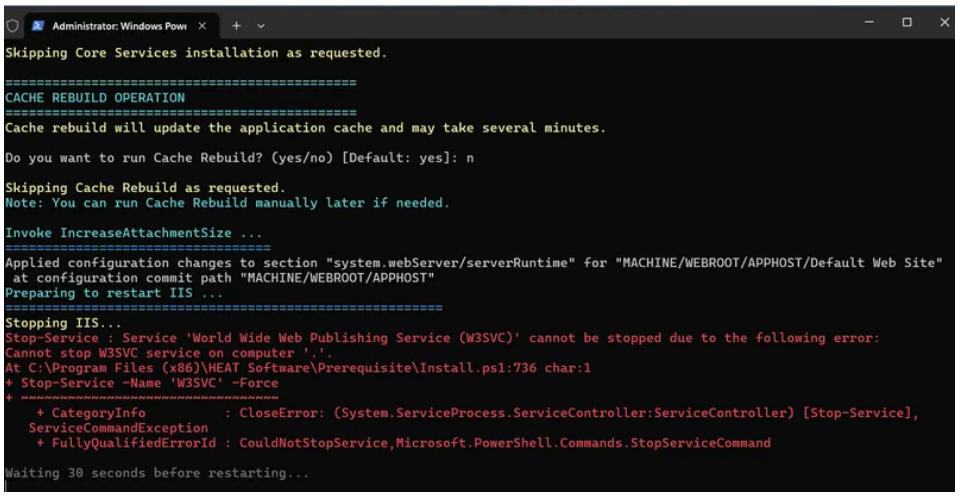
        }

}
```

10. Known Issues

Release Version 2025.4

Bug ID: 1742513

Issue	IIS (W3SVC) Service Stop fails with PowerShell error during SCW configuration
Expected Result	IIS restarts successfully without throwing an error and Powershell window to close automatically.
Actual Result	IIS World Wide Web Publishing Service (W3SVC) fails to stop and does not restart during SCW configuration. Also, throws a PowerShell error. 
Workaround	Manually restart the IIS (W3SVC) Service from the Windows Server settings or control panel.

Release Version 2025.3

Bug ID: 1674149

Issue	You cannot access the **Service Catalog** business object under the **Service Owner** role if you are not using the **English (United States)** locale. If you have changed your locale to another and try to access any request under **Service Catalog**, you might receive an error stating: Access denied to table Frs_def_self_service_literals#. This issue is occurred because the Object Permissions are not set properly for the **Frs_def_self_service_literals** business object.
Expected Result	For any locale, you should be able to access any request under the **Service Catalog** business object.
Actual Result	For non-**English (United States)** locales, you can't access some of the requests under the **Service Catalog** business object.
Workaround	1. Log in to ITSM as an Administrator. 2. Open the Configuration console, and go to **Configure** > **Users and Permissions** > **Roles and Permissions** > **ServiceOwner** role > **Object Permissions**. 3. Find **Frs_def_self_service_literals** business object, and then select the **View** checkbox. 4. Click **Save**.

Release Version 2025.2

No known issues in this release.

Release Version 2024.3

No known issues in this release.

Release Version 2024.2

No known issues in this release.

Release Version 2023.4

Bug ID: 1369214

Issue	When using HTTP - unable to create incident due to console errors while creating an issue in Self Service Mobile UI V3.
Workaround	Use HTTPS.

Bug ID: 1369214

Issue	Inbound web service request fails with null reference object error and the message is stuck in **Dispatched** state.
Workaround	Verify and update the dispatch method name for the type **Web Service Handler** to **HandleWebServiceMessage**. 5. Login to Config DB of the on-premise environment. 6. Open the Message Queue Handler Workspace. 7. Open the handler name **Web Service Handler**. 8. Update the Dispatch method name to **HandleWebServiceMessage**. 9. Restart the Ivanti Message Queue Service. In case the Dispatch Name is **HandleMessage**, then incoming web service will not work.

Release Version 2023.3

No known issues in this release.

Release Version 2023.2

Issue	Whenever Asset Processor displays the message "*SQL Error 245: Data Access Error. Conversion failed*".
Workaround	Under CI Business Object select the field **PercentFreeDiskspace** and set **Precision** as 2 and save. Run the device sync again for failed Asset Processor.

Release Version 2023.1

Bug: 1041094:

Issue	ICDP web.release.configfile has 2 endpoints which has the same contract name and binding configuration. Due to this the files generated by SCW for OnPrem has 2 nodes with incorrect addresses that causes software messages to timeout.
Details	1. Log in to Neurons for ITSM - Administrator and User Help. 2. Open the Configuration console. 3. Go to **Integration Tools** > **Data Import Connections** > **Ivanti Neurons Connector –Device**. 4. Validate the test connection and proceed till the final screen. 5. Click **Run now**.
Expected Result	The status message in the Integration queue is displayed as *Timed out*.
Actual Result	The status message in the Integration queue should be displayed as *Completed*
Workaround	Remove the redundant entry that includes the actual hostname and update the "localhost" entry in the web.config file to specify the port number as 7200.

Release Version 2022.3

No known issues in this release.

Release Version 2022.2

No known issues in this release.

Release Version 2022.1

Bug: 923520:

Issue	When the **Archive Mailbox Name** has uppercase text, the email services do not work.
Details	6. Log in to ITSM as an Administrator. 7. Click **more** and search for **Inbox** Business Object and open it. or 8. From the Configuration console, select **Email Configuration** > **Mail Boxes**. 9. Select any Protocol except **IMAP4**. 10. Set the following options to **Move to Folder**: • **Action To Perform On Processing Success** • **Action To Perform On Processing Error** 11. Enter the value as **Absolute** in the **Archive Mailbox Name** field.
Expected Result	Email services should work.
Actual Result	Email Service do not work.
Workaround	Enter only lowercase letter in the **Archive Mailbox Name** field.

Bug: 923520:

Issue	Email services do not work for **IMAP4** Protocol when the **Action To Perform On Processing Success** and **Action To Perform On Processing Error** options are set to **Move to Folder**.
Details	1. Log in to ITSM as an Administrator. 2. Click **more** and search for **Inbox** Business Object and open it. or 3. From the Configuration console, select **Email Configuration** > **Mail Boxes**. 4. Set the following options to **Move to Folder**: • **Action To Perform On Processing Success** • **Action To Perform On Processing Error**
Expected Result	Email services should work.
Actual Result	Email Service do not work.
Workaround	Delete the Inbox settings record and create a new one. Ensure the following options are set to **Delete**: • **Action To Perform On Processing Success** • **Action To Perform On Processing Error**

Bug: 923520:

Issue	Email services do not work when the **Action To Perform On Processing Success** and **Action To Perform On Processing Error** options are changed from **Move to Folder** to **Delete** without deleting the data in the **Archive Mailbox Name** field.
Details	1. Log in to ITSM as an Administrator. 2. Click **more** and search for **Inbox** Business Object and open it. or 3. From the Configuration console, select **Email Configuration** > **Mail Boxes**. 4. Select the **Protocol** as **IMAP4**. 5. Set the following options to **Move to Folder**: • **Action To Perform On Processing Success** • **Action To Perform On Processing Error** 6. Enter some value in the **Archive Mailbox Name** field. 7. Without deleting the data in the **Archive Mailbox Name**, change the following options to **Delete**. • **Action To Perform On Processing Success** • **Action To Perform On Processing Error**
Expected Result	Email services should work.
Actual Result	Email Service do not work.
Workaround	Before changing the following options to **Delete**, ensure to delete the data from the **Archive Mailbox Name** field. • **Action To Perform On Processing Success** • **Action To Perform On Processing Error**

Release Version 2021.4

No known issues in this release.

Release Version 2021.3

No known issues in this release.

Release Version - 2021.2

Following known issue is in Ivanti Neurons for ITSM 2021.2 release:

1. Bug 845546: When upgrading from 2021.1 to 2021.2, the application throws an error.
2. Bug 811955: Request Offerings do not display in alphabetical order when the Global Constant **EnableUKEnglishLocalization** is set to **True**.

Issue Details

1. Bug: 845546:

2. Bug: 811955:

Issue	Request Offerings do not display in alphabetical order when the Global Constant **EnableUKEnglishLocalization** is set to **True**.
Details	1. Log in to Neurons for ITSM 2. Open the Configuration console > **Global Constants** > set the **EnableUKEnglishLocalization** option as **True**. 3. Open **Languages** and add English UK Culture. If it is already added, remove and re-add. 4. Open the Service Desk console > **Request Offerings**. The list of Request Offerings are in alphabetical order. 5. Edit one of the offering except the first one from list, save and exit the page.
Actual Result	The recently edited offering is on the top of the list.
Expected Result	The Request Offerings should still be displayed in alphabetical order.

Release Version - 2021.1

Following known issue is in Ivanti Neurons for ITSM 2021.1 release:

Bug 822605: - Provisioning report on SSRS 2019 for custom authentication throws a 401 unauthorized error.

Issue Details

Bug: 822605:

Issue	Provisioning report on SSRS 2019 for custom authentication throws a 401 unauthorized error.
Details	Steps: 1. Install 2021.1 on windows 2019 and sql 2019. 2. Provision report on SSRS 2019. 3. Select **Custom Authentication** for **Microsoft SSRS Authentication Type**. 4. Under **Service Account**, select **Use built-in account** and **Network Service** from the drop down list. 5. Click **Next** and provide the necessary information and click **Provision Report Now**.
Actual Result	The application throws a 401 error.
Expected Result	Report provision should be successful.
Workaround	Stop and start the Report Server Configuration Manager (SSRS) and provision the report again.

Release Version - 2020.4

Following known issue is in Ivanti Neurons for ITSM 2020.4 release:

Bug 774221: - Enabling chat fails.

Issue Details

Bug: 774221:

Issue	Enabling chat fails.
Details	Enabling chat fails with the error message - "Unable to register in chat" or "Login prompted".
Expected Result	Chat feature should be enabled without any errors.
Workaround	Review the Redis configuration inside AppServer directory of the installation and restart the Redis service.

Release Version - 2020.3

Following known issue is in Ivanti Neurons for ITSM 2020.3 release:

Bug 739326 - When trying to install Ivanti Service and Asset Manager, the installer automatically closes without installing upon clicking the **Install** button.

Issue Details

Bug: 739326

Issue	When trying to install Ivanti Service and Asset Manager, the installer automatically closes without installing upon clicking the Install button.
Details	1. Setup a machine with Window authentication. 2. Download the Ivanti Service and Asset Manager installer and run as Administrator. 3. Select the product you want to install, that is ISM, ITAM, or ITxM. 4. Follow the next steps and finally click the **Install** button.
Actual Result	The installer closes automatically.
Expected Result	The product should be installed.
Workaround	Run the installer again.

Release Version - 2020.2

Following is the list of known issues in Ivanti Neurons for ITSM 2020.2 release:

1. Bug 697328 - After session timeout clicking on received notification it redirects to Home page of SSM user in Android and iOS device.
2. Bug 700504 - Enable Biometric Authentication checkbox is not displaying in Android device which has Face ID.
3. Bug 701732 - While renewing session using password, it throws time stamp exception in Android device.
4. Bug 702094 - Navigating to Ivanti Cloud Workspace throws an error.
5. Bug 703679 - TRANSLATION-Japanese-Administrator-Chat Configuration-"Seconds" is mistranslated as "2nd".
6. Bug 711161 - While upgrading from 2020.1, SCW throws an error if the value for Privileged User is set as 1 under Concurrent User section.

Issue Details

1. Bug: 697328

Issue	After session timeout clicking on received notification it redirects to Home page of SSM user in Android and iOS device.
Details	**Prerequisite**: • Ensure ISM application is configured to receive push notifications on your device. 1. Login to ISM mobile app as SSM role in Android or IOS device 2. Wait for session timeout (configure session time out in Admin UI) 3. In desktop create an incident to get notification in the mobile device 4. Click on the newly received notification 5. It redirects to Login page, enter valid credential to login.
Actual Result	After successful login it redirects to Home page of SSM user.
Expected Result	After successful login it should redirect to Notification Corner.
Workaround	After landing on the home page, user can click on the bell icon on top right to navigate to notification corner to view notification.

2. Bug: 700504

Issue	Enable Biometric Authentication checkbox is not displaying in Android device which has Face ID.
Actual Result	**Prerequisite**: • Ensure Biometric authentication is enabled on your device. 1. Install latest ISM app in Android device which has Face ID enabled in the device. 2. Launch the App and access the url. 3. In the login page observe for Enable Biometric Authentication check box.
Expected Result	Enable Biometric Authentication checkbox is not displayed to the user.
Actual Result	Enable Biometric Authentication checkbox should be displayed to the user.
Workaround	Use Phone PIN to login or enable fingerprint authentication on your mobile device.

3. **Bug: 701732**

Issue	While renewing session using password, it throws time stamp exception in Android device.
Actual Result	**Prerequisite**: • Session timeout value should be set in the Admin UI. e.g – set the value to 1 minute. 1. Login to ISM using passowrd in Android device. 2. Wait for session timeout. 3. Click on any workspace. 4. In the "You were logged out. Would you like to continue working?" pop up click the Yes button. 5. In the login page enter password and click the login button.
Expected Result	It throws time stamp exception.
Actual Result	It should login successfully to ISM mobile app.
Workaround	Close the app, relaunch and login again.

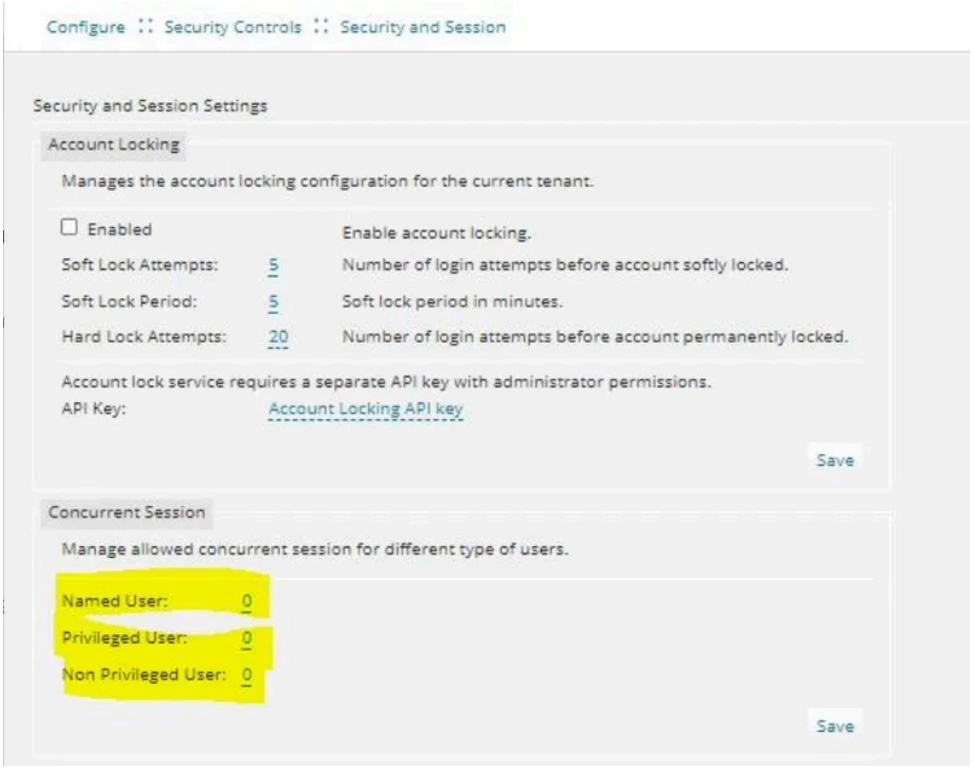
4. **Bug: 702094**

Issue	Navigating to Ivanti Cloud Workspace throws an error.
Actual Result	Note: This workspace newly added in 2020.2 and was not present in previous releases. 1. Login to ISM as Administrator. 2. Open Ivanti Cloud workspace. 3. In the login page enter password and click the login button.
Expected Result	Workspace doesn't load and displays error.
Actual Result	Workspace should load without any error.
Workaround	No workaround available.

5. **Bug: 703679**

Issue	TRANSLATION-Japanese-Administrator-Chat Configuration-"Seconds" is mistranslated as "2nd".
Actual Result	1. Login to the tenant as Administrator. 2. Go to the Chat Configuration workspace. 3. Click UI under Session Configuration Settings.
Expected Result	"Seconds" is translated as 2nd in time.
Actual Result	Seconds should be properly translated.
Workaround	No workaround available.

6. **Bug: 711161**

Issue	While upgrading from 2020.1, SCW throws an error if the value for Privileged User is set as 1 under Concurrent User section.
Workaround	1. Log in to Neurons for ITSM with the Administrator role. 2. Open the Configuration console. 3. Under Security Controls, select Security and Sessions. 4. Under Concurrent Session, change the value of Named User, Privileged User, and Non Privileged User to 0. 5. Save the changes and refresh the page and re-validate the values are changed. 6. Now you can install 2020.2 Neurons for ITSM. 7. After installation is complete, you can change the concurrent users values as it was earlier. **Note:** The user session information is stored in Frs_system_settings in case of sql update. 

Release Version - 2020.1

Following is the list of known issues in Ivanti Neurons for ITSM 2020.1 release:

1. **Bug 670687** - ISM SSO: Session renew fails for OpenID and SAML OKTA for Android device.
2. **Bug 670665** - ISM SSO: After session timeout user is redirected to different browser while default authentication is activated if user don't want to work in Android device.
3. **Bug 670661** - ISM SSO: After session timeout while renewing the session, application opens default external authentication login page in different browser in iOS device.
4. **Bug 636223** - Ivanti cloud transfer: Blank record is getting created in Hardware workspace for device.
5. **Bug 648276** - Software Asset creation has intermittent issues.
6. **Bug 671029** - Username and roles are not visible as the text is in white color by default.

7. Bug 431449 - On-premise : Installation of Metric server for the first time fails.
8. Bug 678308 - On-premise: Unable to Login to Tenant once SCW completed for ITAM/IxM/ISM clean windows authentication.
9. Bug 676494 - Default value of Request Offering text field set to Current User Function (Primary Phone) is not reflecting in Service Catalog.

Issue Details

1. Bug: 670687

Issue	ISM SSO: Session renew fails for OpenID and SAML OKTA for Android device.
Details	**Prerequisite**: • Set any one of the external authentication as default. And set timeout session as 1 minute in Admin UI for validation purpose. 1. ISM Android Mobile App using OpenID or SAML OKTA as external authentication provider. 2. Wait for more than 1 minute. 3. Click on any one of the workspace. 4. The message "*You were logged out. Would you like to continue working?*" is displayed. 5. Click **Yes**.
Actual Result	The same message "*You were logged out. Would you like to continue working?*" is displayed.
Expected Result	The session should renew without any error.
Workaround	No workaround available.

2. Bug: 670665

Issue	ISM SSO: After session timeout user is redirected to different browser while default authentication is activated if user don't want to work in Android device.
Actual Result	On timeout of your session, if you select **No** button for the "*You were logged out. Would you like to continue working?*" the application redirects to different browser and opens default authentication login page in different browser and application continues loading in Android Mobile App.
Expected Result	The application should logout the user session from the Android app.
Workaround	No workaround available.

3. Bug: 670661

Issue	ISM SSO: After session timeout while renewing the session, application opens default external authentication login page in different browser in iOS device.
Actual Result	**Scenario 1:** When **Yes** button is clicked for the timeout session message, default authentication provider login page opens in different browser and in mobile app ISM keeps loading. **Scenario 2:** On clicking the external authentication hyperlink, the external authentication login page is opened in different browser instead of opening within the mobile app.
Expected Result	The application should open in the ISM app.
Workaround	No workaround available.

4. **Bug: 636223**

Issue	Ivanti cloud transfer: Blank record is getting created in Hardware workspace for device.
Workaround	No workaround is available.

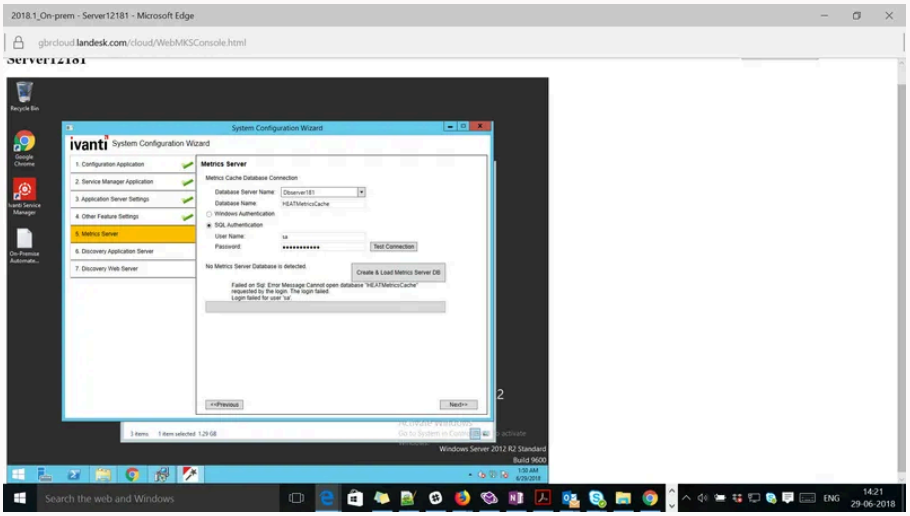
5. **Bug: 648276**

Issue	Software Asset creation has intermittent issues. For example, when a software is installed on different system, after sometime the same software asset is created twice.
Workaround	No workaround is available.

6. **Bug: 671029**

Issue	Username and roles are not visible as the text is in white color by default.
Details	1. Login to ISM with Admin role. 2. Verify the logged in username and role.
Actual Result	The username and role are not visible as the text is in white color.
Expected Result	The logged in username and role should be visible.
Workaround	Navigate to the Configuration console > Style Editor and change the role text's color to black.

7. **Bug: 431449**

Issue	On-premise - Installation of Metric server for the first time fails with db connection error.  **Steps:** 1. Install the latest 2020.1 build on Windows 2014 R2 or any OS. 2. Select the Metric Server configuration in SCW.
Actual Results	The application fails to create the Metric Server database after the "Create" action though the db test connection is successful.
Expected Results	The application should create the Metrix Server database.
Workaround	Recreate works fine.

8. **Bug: 678308**

Issue	On-premise: Unable to Login to Tenant once SCW completed for ITAM/IxM/ISM clean windows authentication. **Steps:** 1. Set up ITAM clean windows authentication. 2. Once SCW completed open tenant enter valid credentials to login to tenant.
Actual Results	The application throws an error.
Workaround	Re-run the SCW and try to login.

9. **Bug: 676494**

Issue	Default value of Request Offering text field set to Current User Function (Primary Phone) is not reflecting in Service Catalog. **Prerequisite:** • Add Phone number to the current user. **Steps:** 1. Login to the application as Admin or Service Owner. 2. Create a new Request Offering. 3. Go to **Design Request** form and add a text field. 4. Edit the field and set the default value to **Function** and select **Current User: Primary Phone** from the drop down list. 5. Publish the Request Offering and save. 6. Open the created Request Offering in Service Catalog and verify the current user phone number is populated.
Actual Results	Current User Phone number is not displayed in the service catalog.
Workaround	This is specific to 2019.1.0 OOTB onwards. No workaround available.

Release Version - 2019.3.1

Following is the list of known issues in Ivanti Neurons for ITSM 2019.3 release:

1. Bug 636223 - Ivanti cloud transfer: Blank record is getting created in Hardware workspace for device.
2. Bug 648276 - Software Asset creation has intermittent issues.

Issue Details

1. Bug: 636223

Issue	Ivanti cloud transfer: Blank record is getting created in Hardware workspace for device.
Workaround	No workaround is available.

2. Bug: 648276

Issue	Software Asset creation has intermittent issues. For example, when a software is installed on different system, after some time the same software asset is created twice.
Workaround	No workaround is available.

Release Version - 2019.3

Following is the list of known issues in Ivanti Neurons for ITSM 2019.3 release:

1. Bug 616198 - Heat Mobile app may crash when iOS is upgraded from version 12.3.1 to 13.1.

2. **Bug 610706** - Post 2019.2 upgrade, Audit History is not updated based on the relationship link for FRS_Approval business object.
3. **Bug 615685** - Ivanti Cloud (UNO) Login App V1 & V2 integration does not work on ISM tenant which has Pre-MCT database.
4. **Bug 620323** - Duplicate entries for workflow and task on creating a service request from a backup database.
5. **Bug 635269** - IP Whitelisting shows "Invalid IP" for all FRSHeat integration API requests when session key from "Authenticate User" is used.
6. **Bug 644185** - The application throws an error when trying to link new and edit existing records in Telemetry Logging Configuration tab in config db tenant.
7. **Bug 70551** - Embedding content using iFrame in HTML UI Control is not working as expected.
8. **Bug 518404** - Request Offering is not getting created in the tenant, when imported via the Release Tool patch file.
9. **Bug 632194** - IP whitelisting - Integration trusted host is not allowing any IP to send/receive request, when the load balancer IP is configured.

Issue Details

1. Bug: 616198

Issue	After upgrading iOS version from 12.3.1 to 13.1, some users may experience crashing of the Heat Mobile app.
Workaround	Un-install and re-install the Heat Mobile app.

2. Bug: 610706

Issue	Post 2019.2 upgrade, Audit History is not updated based on the relationship link for FRS_Approval business object.
Actual Results	1. Upgrade Neurons for ITSM to version 2019.2. 2. Log in to Neurons for ITSM with Admin user role. 3. Open the Configuration console. 4. Navigate to **Business Objects** > **Service Request** > Relationships > ServiceReqAssociatedFRS_Approval. 5. Select the Relationship is Audited check box. 6. Save and exit. 7. Create a Service Request that generates an approval, for example, Mailbox Quota Request. 8. Save the Service Request and exit. 9. Re-open the Service Request and click the Audit History tab. 10. No link to the Approval record is added in the Audit History.
Expected Results	A link to the Approval record should be added in the Audit History.

3. Bug: 615685

Issue	Ivanti Cloud (UNO) Login App V1 & V2 integration does not work on ISM tenant which has Pre-MCT database.
Workaround	Upgrade the pre-MCT tenant using MCT migration tool before performing IvantiCloud (UNO) Integration.

4. Bug: 620323

Issue	Duplicate entries for workflow and task on creating a service request from a backup database.
Workaround	Note: The issue is re-producible when you restore the 2019.3 OOTB database from any existing tenant. You need to run a clean up script to remove the unwanted entries from specific table so that no duplicate workflow is triggered. Run the below query in the restored tenant database and "isvisble=0" condition is mandatory. Make sure the workflow service is stopped before running this script and start after running. "select * from Frs_def_businessrules where isvisible = 0"

5. Bug: 635269

Issue	IP Whitelisting shows "Invalid IP" for all FRSHeat integration API requests when session key from "Authenticate User" is used.
Workaround	Use the SID (from ISM application > **Developer Tools**) to authenticate API requests.

6. Bug: 644185

Issue	The application throws an error when trying to link new and edit existing records in Telemetry Logging Configuration tab in config db tenant.
Actual Results	1. Log in to a config db tenant of Neurons for ITSM with Admin user role. 2. Open a tenant record and navigate to Telemetry Logging Configuration tab. 3. Click **Link** > select a service record to link > **Add** or **Edit** button. The application displays an error message.
Expected Results	The selected service record should be linked to the Telemetry Logging Configuration.
Workaround	Go to the Telemetry Logging Configuration workspace, add new service records or existing records here. The updated services will be visible in the Telemetry Logging Configuration tab of the tenant page which you can now link.

7. Bug: 70551

Issue	Embedding content using iFrame in HTML UI Control is not working as expected.

8. Bug: 518404

Issue	Request Offering is not getting created in the tenant, when imported via the Release Tool patch file.

9. Bug: 632194

Issue	IP whitelisting - Integration trusted host is not allowing any IP to send/receive request, when the load balancer IP is configured.

Release Version - 2019.2

There are no known issues in Ivanti Neurons for ITSM 2019.2 release.

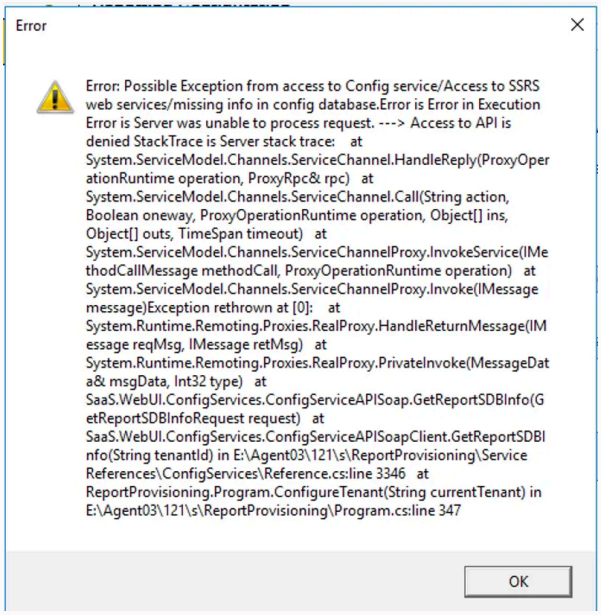
Release Version - 2019.1 and earlier

Given below is the list of known issues in Ivanti Neurons for ITSM 2019.1 and earlier releases.

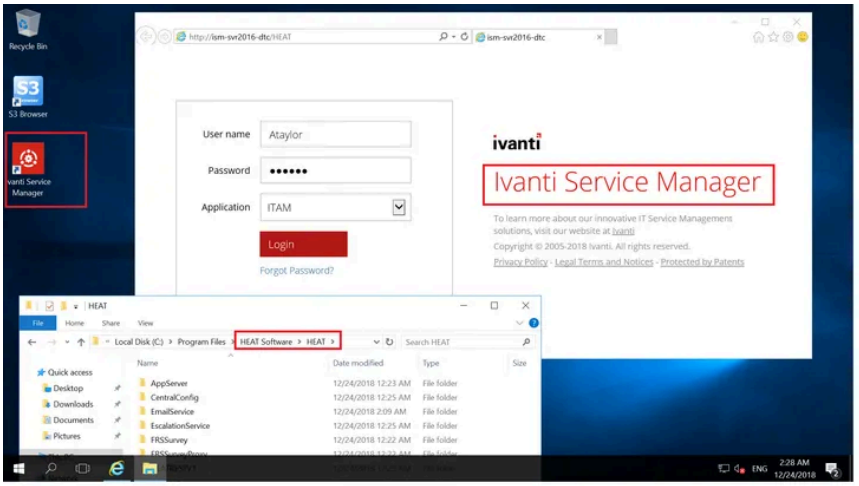
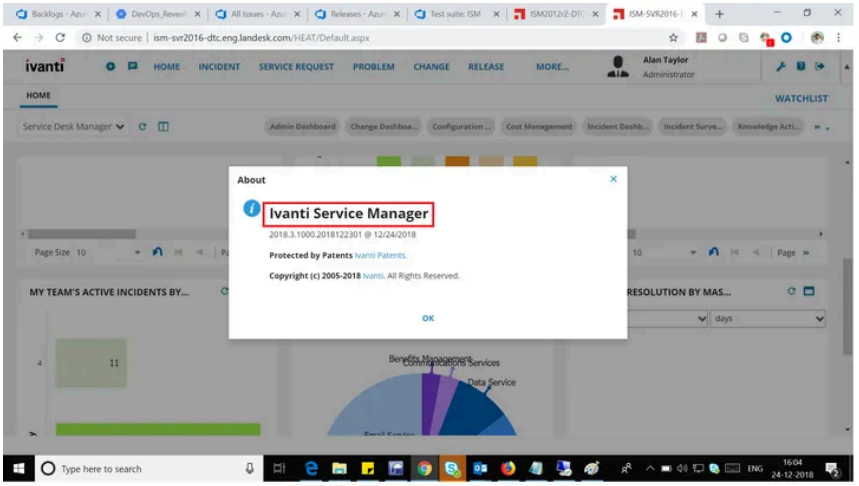
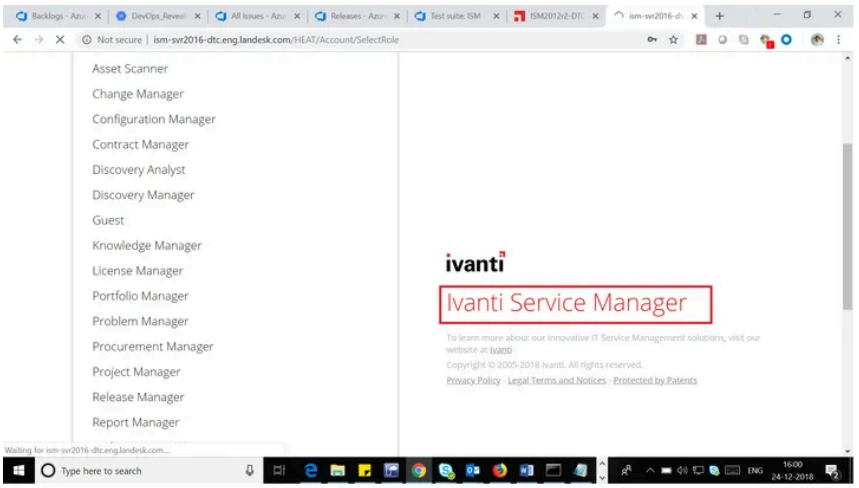
1. **Bug 476581** - Configuration fails if the App server cache does not have trusted host information of database server.
2. **Bug 492307** - On-premise - Re-branding is pending in the ITAM application as it still refers to "Ivanti Service Manager".
3. **Bug 492520** - SSRS configuration with custom authentication and domain credentials may throw the 1326 error sometimes.
4. **Bug 492643** - Ops console is not re-branded for Ivanti, it still shows HEAT brand names.
5. **Bug 333924** - Configure OpenIDConnect Auth Provider for MS Azure AD.
6. **Bug 341934** - [Request for Information] Report: Distribution option "Download".
7. **Bug 505677** - Multiple records are displayed on the home page search for a single request.
8. **Bug 502699** - SSRS configuration fails while trying to add the provision report in SCW.
9. **Bug 504840** - There is a browser compatibility issue for Self Service Mobile, in the UI price list/ Cost panel showing Zero(0).
10. **Bug 383101** - Japanese characters are shown as ? in the FRS_SurveyAnswer.
11. **Bug 359153** - Logout from the Self Service Mobile UI takes users to the ISM login instead of the Auth provider login.
12. **Bug 508376** - "FrontRange Solution" installer certificate is not trusted by Microsoft anymore.
13. **Bug 519086** - Upgrading 2019.1 directly from 2018.1.1 does not work.
14. **Bug 522220** - Unable to create new CI due to the Asset Processor Configuration.
15. **Bug 525479** - Login page not loading by throwing Timestamp Error while Internal Services user is disabled.
16. **Bug 522214** - Config db/tenant db metadata version field left blank during upgrade/install.
17. **Bug 431449** - On-premise : Installation of Metric server for the first time fails.
18. **Bug 526262** - Upgraded ISM 2018.3.1 to 2019.1 and post upgrade accessing the below workspaces as LicenseManager role throws exception.
19. **Bug 485595** - The Sting "Self" is UNLOC shown under Self Service User - My Items > Benefits Package Claim.
20. **Bug 235039** - The Service Request Review and Submit page displays the incorrect employee details if the login IDs are similar.
21. **Bug 533210** - 2019.1 premise installer seems to silently exit or crash after installing one of the pre-req modules.
22. **Bug 576301** - Errors on Form and Grid for BO AuthenticationPrincipal when upgrading from 2018.3.1 to 2019.1.0 premise.

Issue Details

Bug: 476581

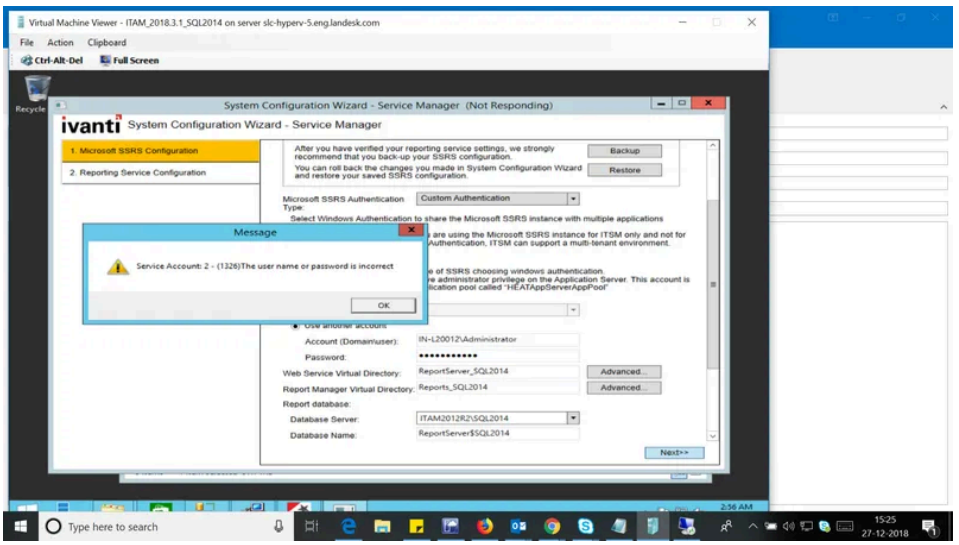
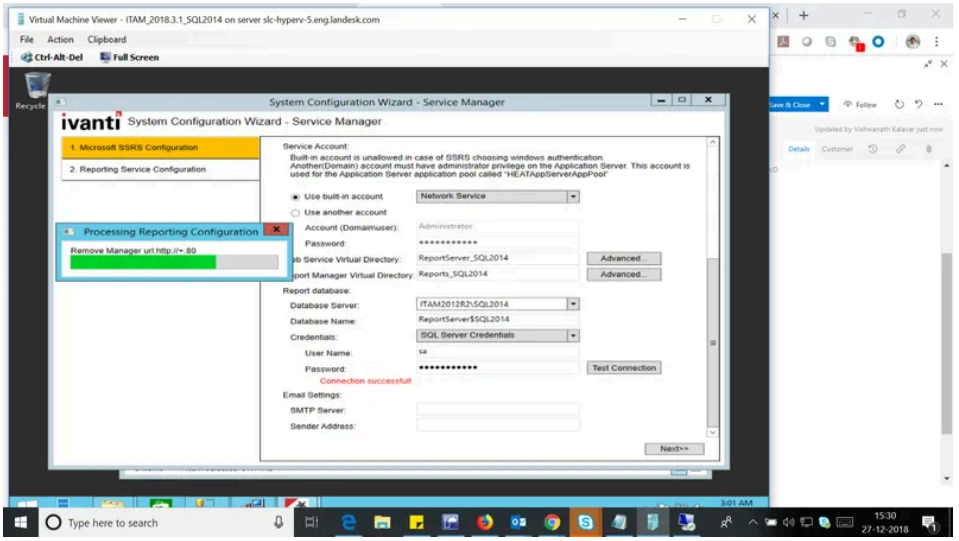
Issue	Configuration fails if App server cache does not have trusted host information of database server 1. Install or upgrade ISM application to latest build version of 2018.1.1 2. Install or upgrade the reporting service 3. Provision the report as part of configuration Use Custom Authentication method for SSRS configuration and SQL authentication for database connection type. 
Workaround	There is a timing issue on provisioning. If the application cache is not rebuilt after updating the Trusted Host Entry in the config database (central config webapp), then provisioning of reports will fail as post config calls fails (Exception to access config service). Hence, it is advised to make sure Trusted Host Address are already added in the config database and IIS is reset (or wait for 15 mins to rebuild cache).

Bug 492307

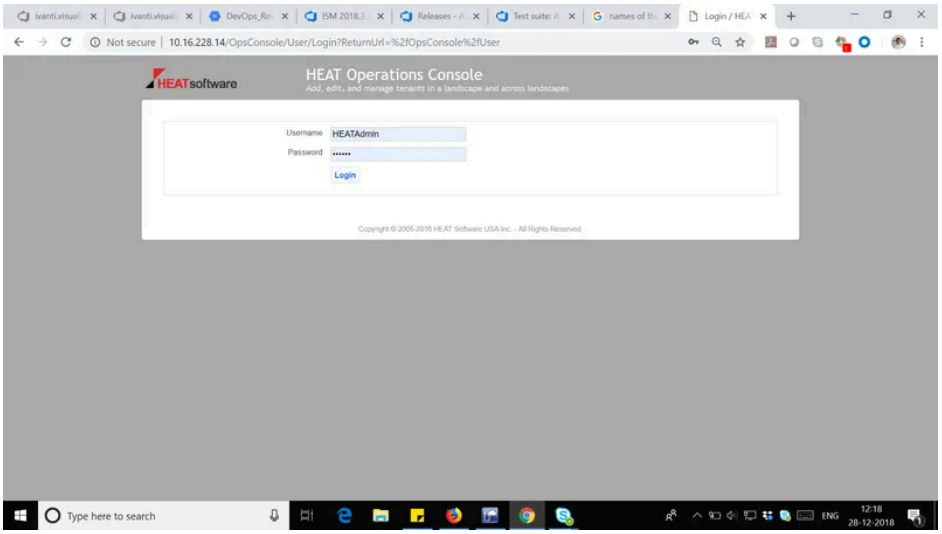
Issue	On-premise - Re-branding is pending in the ITAM application as it still refers to "Ivanti Service Manager". **Steps to reproduce** Install ITAM application and check the product version and other branding-related information.
Actual Results	  

Issue	On-premise - Re-branding is pending in the ITAM application as it still refers to "Ivanti Service Manager". **Steps to reproduce** Install ITAM application and check the product version and other branding-related information.
Expected Results	Currently, the 2018.3.1 on-premise version is built on top of the ISM platform and continues to refer ISM in many places. This needs to be fixed.
Workaround	No application functionality should be lost.

Bug 492520

Issue	SSRS configuration with Custom Authentication may fail sometimes if the domain user credentials provided is working. You will see the 1326 error. **Steps to reproduce** Configure SSRS with Custom Authentication with Domain credentials.
Actual Results	SSRS configuration with Custom Authentication is shown below: 
Expected Results	Provided domain credentials should work and users should be able to proceed.
Workaround	Use Custom authentication with the User in-built account type "Network Service" to proceed. 

Bug 492643

Issue	2018.3.1 ops console is not re-branded for Ivanti, it still shows HEAT brand names. **Steps to reproduce** Login to ops console by accessing the following path: http://10.16.228.14/OpsConsole/User
Actual Results	

Bug 333924

Issue	When configuring OpenIDConnect Authentication Provider for MS Azure AD and making "Test Authentication", he follwoing message appears: "Use Sub as external login as e-mail is not available at this step." Azure AD does not return "email" claim in response and there is no "email" in "Extracted JWT token". Instead of returning email in "email" claim, Azure returns an email in "upn" claim: The following errors occur: User not found for external login qX_OE8OkmVyYRXYI7chX4qqSWPyCQLVRI8CU_YtYOY8. Try to use auto-provisioning ... Value is missing for 'email' System.ArgumentException: Value is missing for 'email' ?? at HEAT.Authentication.Sso.Oidc.OidcResultHandler.GetDictStringValue(IDictionary`2 dict, String ... Authentication failed unexpected error
Actual Results	Configuration to install Ivanti SM handler to use "upn" instead of "email.

Bash


```
"ver":"1.0"}
```


Bash


```
"uti":"ZmsLjPCsLEqe3Kbp8qkDAA",
```

Bash

```
"upn":"andrejs@synergy.lt",
```

Bash

```
"unique_name":"andrejs@synergy.lt",
```

Bash

```
"tid":"23afecaf-6440-44aa-a6f2-b6e38bd02ba9",
```

Bash

```
"sub":"qX_OE8OkmVyYRXYI7chX4qqSWPyCQLVRl8CU_YtYOY8",
```

Bash

```
"onprem_sid":"S-1-5-21-1293440551-2949271503-1314497110-1132",
```

Bash

```
"oid":"83f1b61f-1e77-4ce5-85ad-c76bed5a1247",
```

Bash

```
"name":"Andrej Sandyrov",
```

Bash

```
"ipaddr":"88.119.194.198",
```

Bash

```
"given_name":"Andrej",
```

Bash

```
"family_name":"Sandyrov",
```

Bash

```
"amr":["pwd"],
```

Bash

```
"aio":"Y2NgYGi+6xjp02/t1/erTNy+LKVPa8dC4+/TP3xennmSm1+8+z4A",
```

Bash

```
"exp":1517389749,
```

Bash

```
"nbf":1517385849,
```

Bash

```
"iat":1517385849,
```

Bash

```
"iss":"https://sts.windows.net/23afecaf-6440-44aa-a6f2-b6e38bd02ba9/",
```

Bash

```
"kid":"z44wMdHu8wKsumrbfaK98qxs5YI"}.{"aud":"022c4d20-1d56-4a73-a421-86b2b423f344",
```

Bash

```
"x5t":"z44wMdHu8wKsumrbfaK98qxs5YI",
```

Bash

```
"alg":"RS256",
```

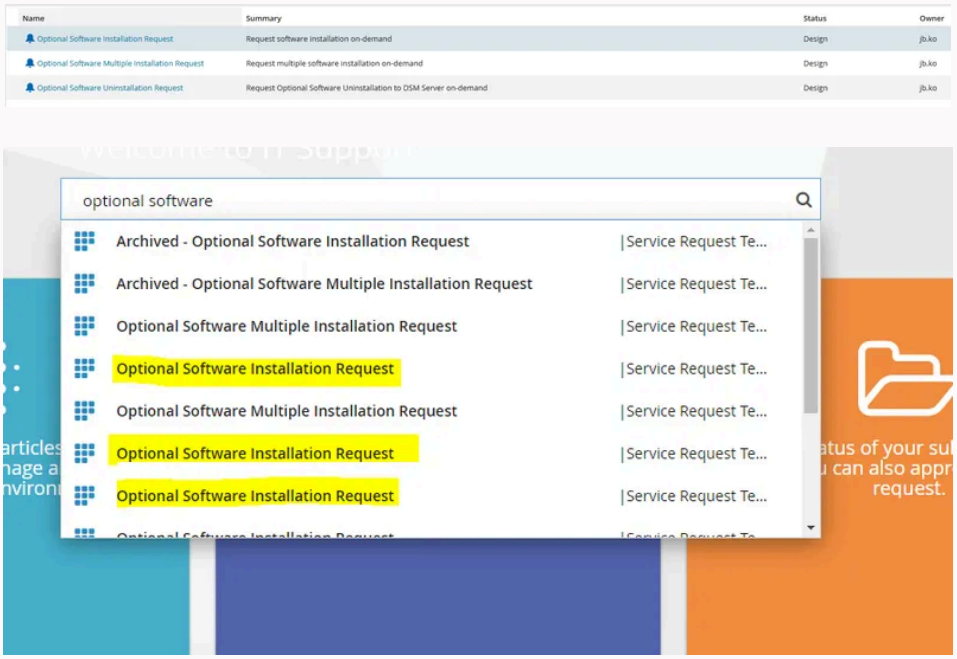
Bash

```
{"typ":"JWT",
```

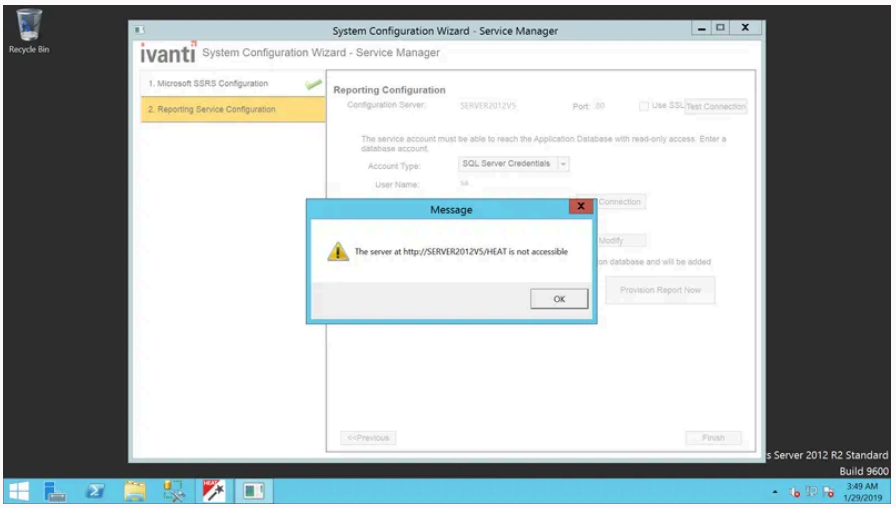
Bug 341934

Issue	When choosing the "Download" option in Distribution Information of the report from the homepage dashboard or report list, download is not happening.
Expected Results	Reports should be downloaded from homepage dashboard or report list.

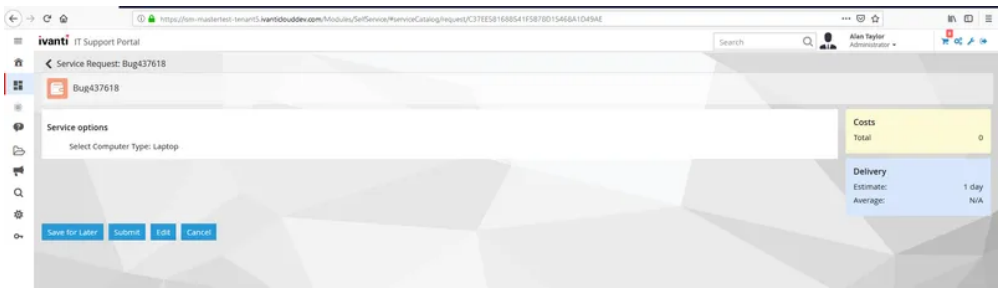
Bug 505677

Issue	In self-service mobile, when searching with the string "optional software" in the home page search box, multiple records that are displayed for a single record and the request offerings are in the "Design" state. **Steps to reproduce** 1. Log in to ISM in the "Self Service Mobile" role. 2. On the home page, search for "optional software". 
Actual Results	The system displays multiple records for a single request that is in design state.
Expected Results	A single record should get displayed for a single request that are in the "Published" state.

Bug 502699

Issue	SSRS configuration fails while trying to add the Provision report in SCW. **Steps to reproduce** 1. Configure the Reporting service configuration. 2. Click on "Provision report now". 
Actual Results	Error message appears as shown above.
Expected Results	There should be no error and the URL should be accessible.

Bug 504840

Issue	There is a browser compatibility issue (in Firefox and Microsoft Edge) for Self Service Mobile, in the UI price list/ Cost panel showing Zero(0). If a user selects an item, for example, Desktop, which is $595 when he submits the request, the cost panel shows $0. Hence, the severity must be kept to 2. In addition, customer bugs (which are linked) are getting blocked and hence, the priority should be set to 2. **Steps to reproduce** 1. Login as an **Admin**. 2. Go to the **Request Offerings** workspace and import a .ROF file, publish, and save. 3. Switch to the **Self Service** (old UI) role. 4. Go to **Service Catalog** and open this offering. 5. Select **Desktop** type, then a **Desktop** and a **Quantity**. Notice that the price list looks correct for the selection. 6. Without submitting or saving, go back and select **Laptop** type, a **Laptop** and a **Quantity**. Observe that the **Costs** panel is still correct and the originally selected **Desktop** item is removed. 7. Switch to the **Self Service Mobile** UI. 8. Go to **Service Catalog** and open this offering. 9. Select **Desktop** type, then a **Desktop** and a **Quantity**. Observe that the price list looks correct for the selection. 10. Without submitting or saving, go back and select **Laptop** type, a **Laptop** and a **Quantity**. Notice that the **Costs** panel retains the previously selected **Desktop** even though it should not.
Actual Results	The **Cost** panel shows zero even if the product is selected.  This is working fine in Chrome, but not working in Firefox and Microsoft Edge.
Expected Results	The **Cost** panel should display the correct value for the selected type.

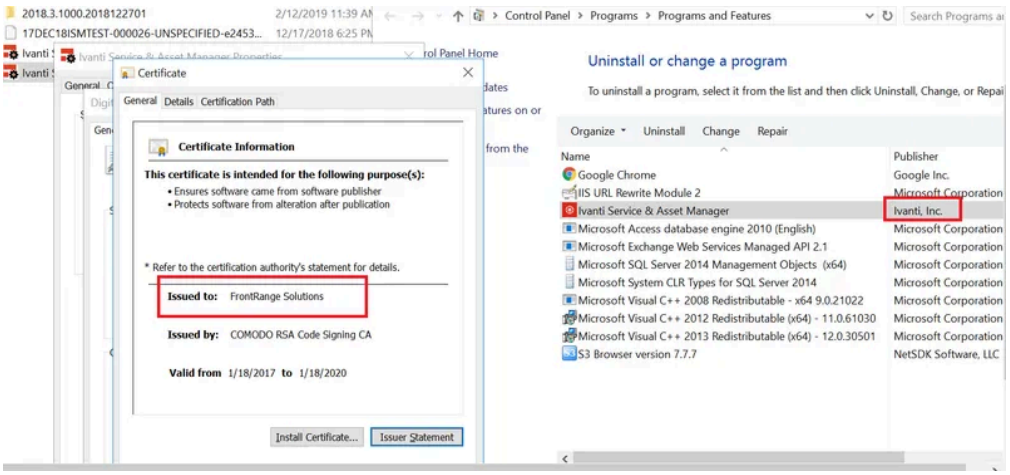
Bug 383101

Issue	Japanese characters are shown as ? in the FRS_SurveyAnswer **Steps to reproduce** This error can be replicated in Demo 2017.3.1 and 2018.1 tenant: https://tenant-dev2.saasitpilot.com/. 1. Open FRS_SurveyAnswer# 2. Go to field QuestionText or AnswerText 3. Change type to unicode, 4. Save.
Actual Results	Error: Unhanded system exception: The given key was not present in the dictionary.
Expected Results	Possibility to set this field to be unicode text.

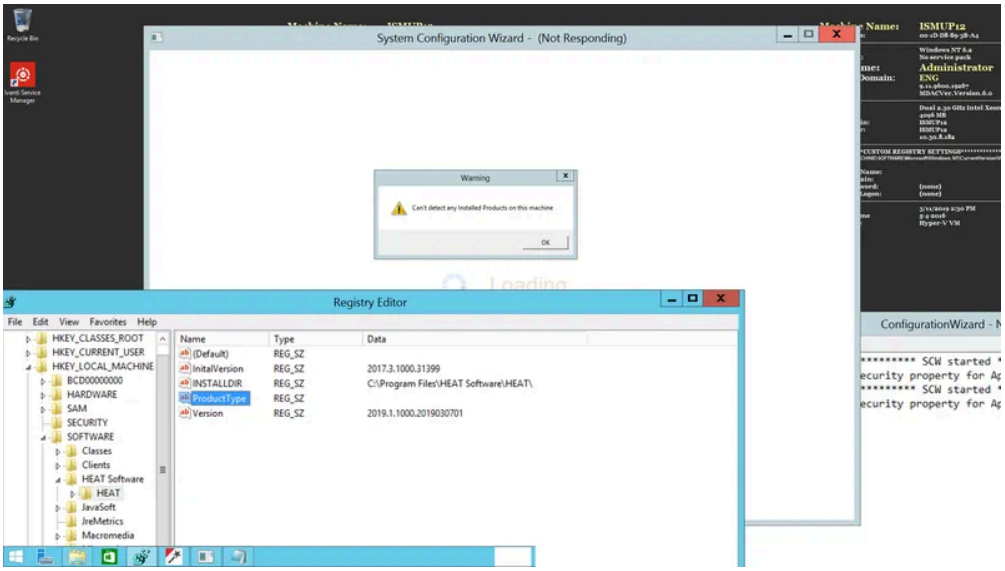
Bug 359153

Issue	Logout from the Self Service Mobile UI takes users to the ISM login instead of the Auth provider login. **Steps to reproduce** 1. Set up the environment with SAML authentication. 2. Set up a logout URL as seen in the screenshot. 3. Log in as admin with the role of Self Service Mobile. 4. Log out later using the logout button.
Actual Results	Users are taken back to the ISM login page for only Self Service Mobile role.
Expected Results	Users should be taken back to the SAML login page.

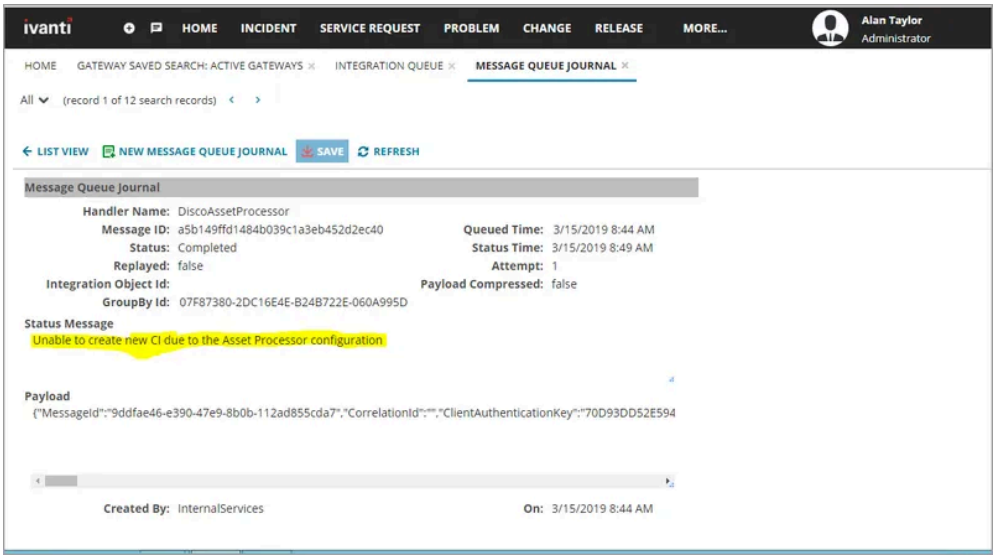
Bug 508376

Issue	The certificate issued for installer is "FrontRange Solution" and Publisher name is "Ivanti Inc." since the last couple of releases. The Windows certificate trust store never questioned the authenticity of the certificate earlier. However, somehow it is prompting the certificate trust question now on Windows OS like 2016, 2012 etc. **Steps to reproduce** 1. Take the latest installer and install on vanilla OS (Win2016, 2012)
Actual Results	
Expected Results	Issue a new certificate to installer with "Ivanti Inc."
Workaround	Accept the certificate to add to MS trust store.

Bug 519086

Issue	If a user tries to directly upgrade from 2018.1.1 to 2019.1.0 or the latest version, the upgrade will fail. This is because SCW fails to launch the post installer update due to the missing registry key for ProductType. This entry should be retained during the upgrade, which is currently not working. 2018.1.1 build: 2018.1.1000.2018103001 @ 10/29/2018 2019.1.1 build: 2019.1.1000.2019030701 The supported upgrade paths for 2018.1.1 to 2019.1.0 application version must be 2018.1.1 > 2018.3.1 > 2019.1.0. **Steps to reproduce** 1. Upgrade the existing 2018.1.1 on-premise version to the 2019.1.0 on-premise version by skipping the 2018.3.1 upgrade path.
Actual Results	
Expected Results	The product reg key must be created during an upgrade and SCW should launch without any error.

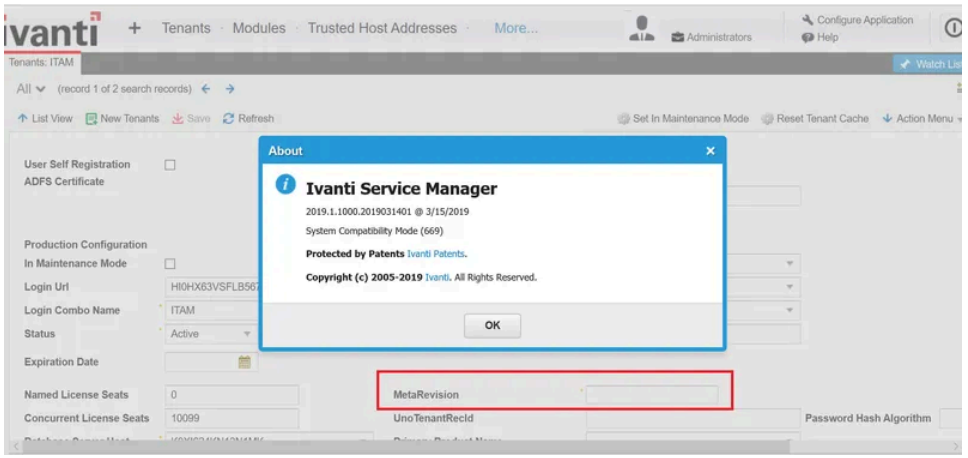
Bug 522220

Issue	Unable to create new CI due to the Asset Processor Configuration. After installing the GatewayDC(Discovery), Gateway CI is not getting generated and getting error message as "Unable to create new CI due to the Asset Processor configuration". This is can be found only on OOTB(IxM) on 2019.1 Builds Onwards. **Steps to reproduce** 1. Pre-requisite: This issue can only be reproduced on OOTB(IxM). SO to reproduce OOTB(IxM) environment is required. 2. Login to the application. 3. Go to the Admin UI and download the GatewayDC/StandardGateway and install with proper credential on the above mentioned kind on tenants environment.
Actual Results	Gateway is not getting created in under Gateway workSpace and Highlighted Error will comes in Message Queue Journal. 
Expected Results	-

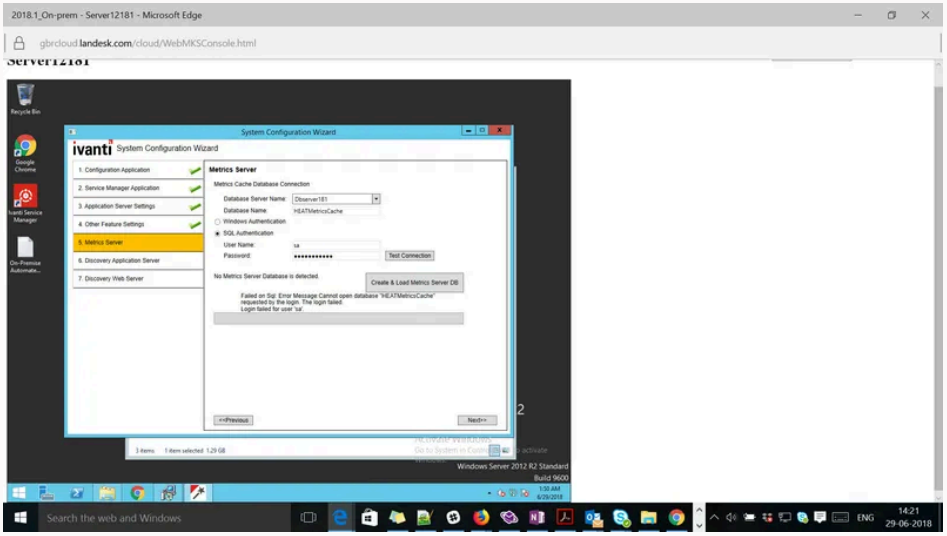
Bug 525479

Issue	When Internal Services user is disabled in a tenant, then login page is not loading by throwing timestamp error. It is a regression issue as the 2018.3.1 environment login page is loading successfully if an Internal Services user is disabled. After 2019.1 deployment to FFT and NVD PLT landscapes where few customers login page is not loading by throwing the Timestamp Error. On Analyzing, ops team found Login "Internal Services" were disabled in those tenants. After enabling the login, they were able to login to the Tenant without any issues. **Steps to reproduce** 1. Login to ISM as the Admin role. 2. Open Employee workspace and search for "Internal Services" user. 3. Disable "Internal Services" user and close the browser or try to logout.
Actual Results	The system is throwing the Timestamp error while loading the login page. 
Expected Results	After disabling "Internal Services" user, login page should be loaded properly without throwing any timestamp error.
Workaround	To work properly, we need to enable the "Internal Services" user.

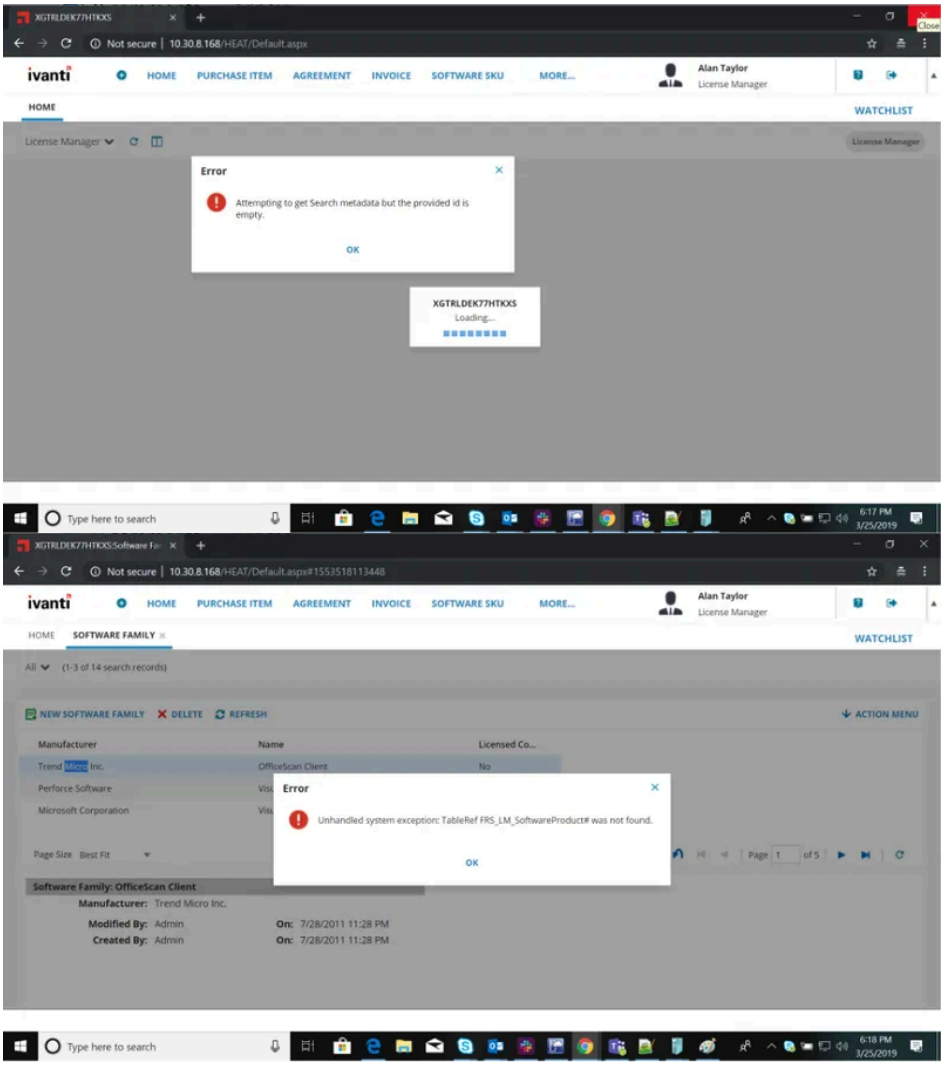
Bug 522214

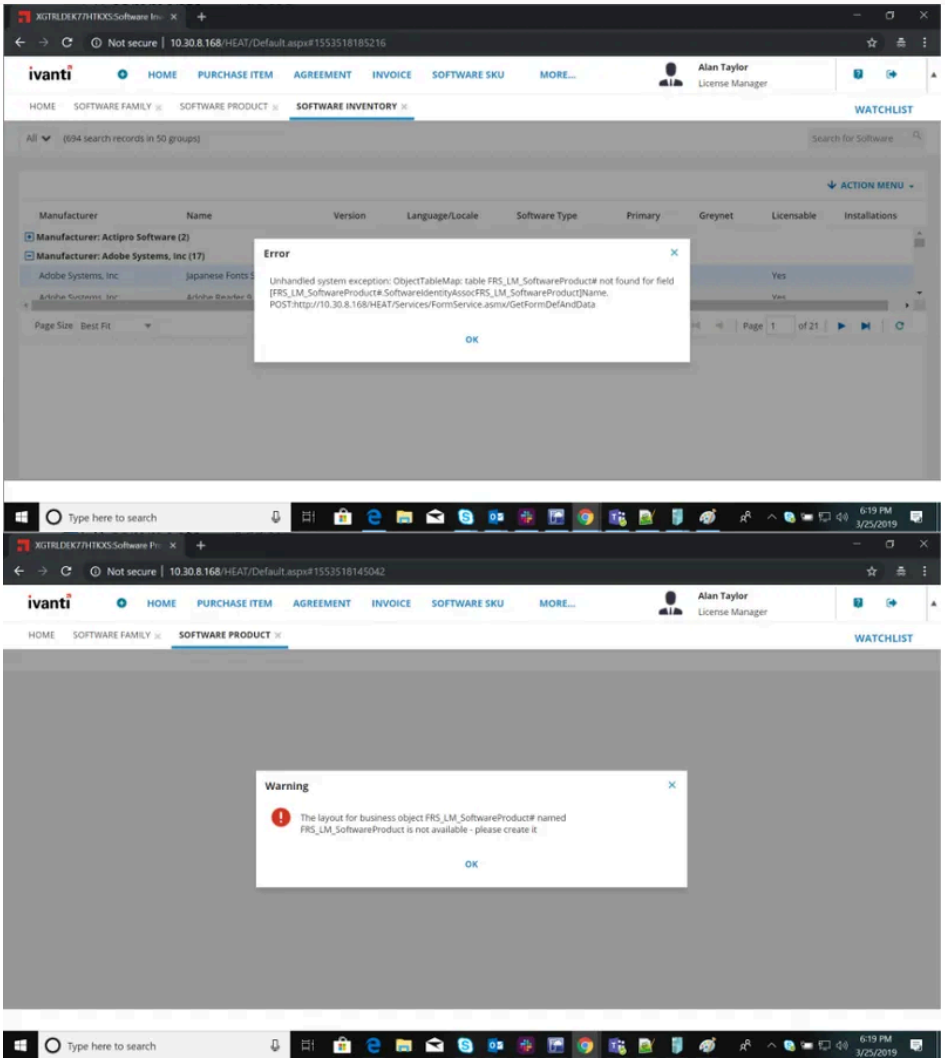
Issue	After installation or upgrade of ISM/AM/IXM application, the metaRevision field is not updated. Regression: NO as it is observed in 2018.3.1 or an earlier version. **Steps to reproduce** 1. Install & upgrade any of the above application 2. Check the MetaRevision version of the application tenant in configuration.
Actual Results	
Expected Results	MetaRevision should be updated or should retain the old value (in case of no system table upgrade) as it is a mandatory field.
Workaround	User should manually enter "670" in this field and save it. This is a workaround only for 2019.1.

Bug 431449

Issue	Version: 2018.1.1 On-premise OS: Windows 2016 SP1 & IIS version 10.0.14393.0 Installation of Metric server for the first time fails with db connection error, however subsequent attempt of "recreate" works fine.  **Steps to reproduce** 1. Install the latest 2018.1.1 build on Windows 2014 R2 or any OS. 2. Select the Metric Server configuration in SCW.
Actual Results	The application fails to create the Metric Server database after the "Create" action though the db test connection is successful. However, subsequent attempt on "recreate' works fine.
Expected Results	-

Bug 526262

Issue	Upgraded ISM 2018.3.1 to 2019.1.0 and post upgrade accessing the below workspaces as LicenseManager role throws exception. **Steps to reproduce** Upgrade ISM 2018.3.1 to 2019.1.0.
Actual Results	LicenseManager home page fails to load. Accessing the Software Inventory, Software Product, Software Family and other workspaces as a LicenseManager throws error in UI. 

Issue	Upgraded ISM 2018.3.1 to 2019.1.0 and post upgrade accessing the below workspaces as LicenseManager role throws exception. **Steps to reproduce** Upgrade ISM 2018.3.1 to 2019.1.0.
	
Expected Results	The above mentioned modules should not be exposed to LicenseManager role.

Bug 485595

Issue	The Sting "Self" is UNLOC shown under Self Service User - My Items > Benefits Package Claim. **Steps to reproduce** 1. Log in to ISM and change the role to **Self Service User**. 2. Open the **My Items** workspace. 3. Select **Benefits Package Claim**. 4. Under **Item Details**, there is one unloc sting "Self" from "PATIENT INFORMATION".
Actual Results	The Sting "Self" is UNLOC shown under Self Service User - My Items > Benefits Package Claim.
Expected Results	The UNLOC string should translated as "自我".

Bug 235039

Issue	Upgraded ISM 2018.3.1 to 2019.1.0 and post upgrade accessing the below workspaces as LicenseManager role throws exception. **Steps to reproduce** Upgrade ISM 2018.3.1 to 2019.1.0.
Actual Results	LicenseManager home page fails to load. Accessing the Software Inventory, Software Product, Software Family and other workspaces as a LicenseManager throws error in UI.
Expected Results	The above mentioned modules should not be exposed to LicenseManager role.

Bug 533210

Issue	2019.1 premise installer seems to silently exit or crash after installing one of the pre-req modules. **Steps to reproduce** 1. Run installer exe as an admin. 2. Click **Install** to installer pre-reqs. A dialog box will appear, "Extracting Ivanti Service Asset Manager.msi. 3. Follow the wizard to where you select SM / AM / both. 4. Select any installation type, click next, then again next, select **Complete**, click next, and then click install. Just after the VC 2013 redist x64 finishes installing the installer exits silently.
Actual Results	Just after the VC 2013 redist x64 finishes the installing process, the installer exits silently.
Expected Results	Installer should progress normally.
Workaround	Re-run the installer EXE.

Bug 576301

Issue	During Update System on Update Application Database in SCW, the system receives a notification that it was updated with visible errors. Example: Patch 'AuthenticationPrincipal' has 5 error(s) Error during metadata commit operation for seq = 12482: System.Exception: Nom de colonne non valide : 'GlobalId'.
Workaround	If a customer receives the error for GlobalID and/or AuthenticationPrincipal, then: 1. Close SCW. 2. Restore the Application DB back to 2018.3.1. 3. Go to the AdminUI and import the “GlobalID_Contract_Identity.MetadataPatch” package. 4. Re-launch SCW. 5. Re-validate and re-select upgrade. The customer should be able to upgrade ISM DB with no errors. Optimally, the customer should run through the following steps so they do not have to wade through a failing upgrade and then have to restore the 2018.3.1 Application DB, thus saving possibly hours of time. 1. After installing 2019.1.0 and before walking through the SCW, launch AdminUI. 2. Import the “GlobalID_Contract_Identity.MetadataPatch” package. 3. Walk through SCW and the customer should be able to upgrade DB.