Redpoint Interaction v6.6

System Administration Guide

Redpoint Global Inc., 888 Worcester Street, Suite 200, Wellesley, MA 02482 T: 1 781-725-0250 F: 1 1 781-583-0464 www.redpointglobal.com

Contents

Preface

Overview

The Redpoint Interaction (RPI) v6.5 System Administration Guide provides installation, configuration and maintenance instructions for IT professionals responsible for implementing the RPI system. Familiarity with database configuration, Windows, IIS and .NET are required to successfully complete all of the setup and maintenance.

Related Documentation

In addition to this Administration Guide, RPI also provides a comprehensive online/offline Reference Guide, and Technical Architecture documentation.

Introduction

What is Redpoint Interaction?

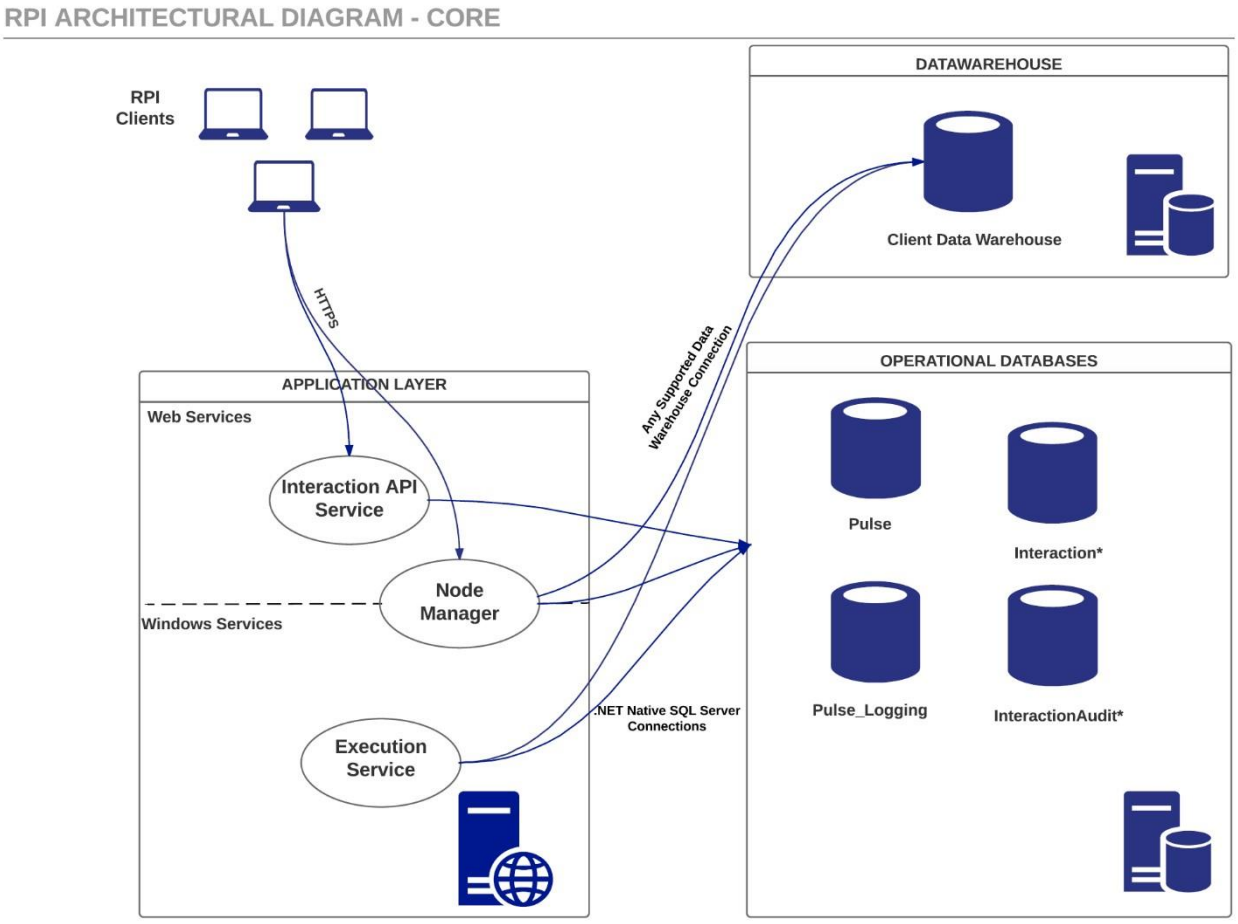
RPI provides the only fully integrated, all-in-one communication platform that enables an always-on, bi-directional, real-time, omni-channel dialog with the customer. With multiple deployment options, ability to scale and rapid implementations, RPI is the most complete and technologically advanced marketing platform available.

Interaction consists of client and server applications. The RPI client application consists of an executable, library files and configuration files. It runs on a local workstation and communicates with the server via the latter's web services. The RPI server can be installed with multiple nodes (cluster) that span several physical or virtual machines. Many RPI client applications can communicate with a single RPI server.

In addition, RPI features true multi-tenancy support. To meet marketing needs for a series of client organizations, each is represented as a separate client within the RPI server. Each client's operational information and communication history are stored entirely separately and securely.

Architecture Diagrams

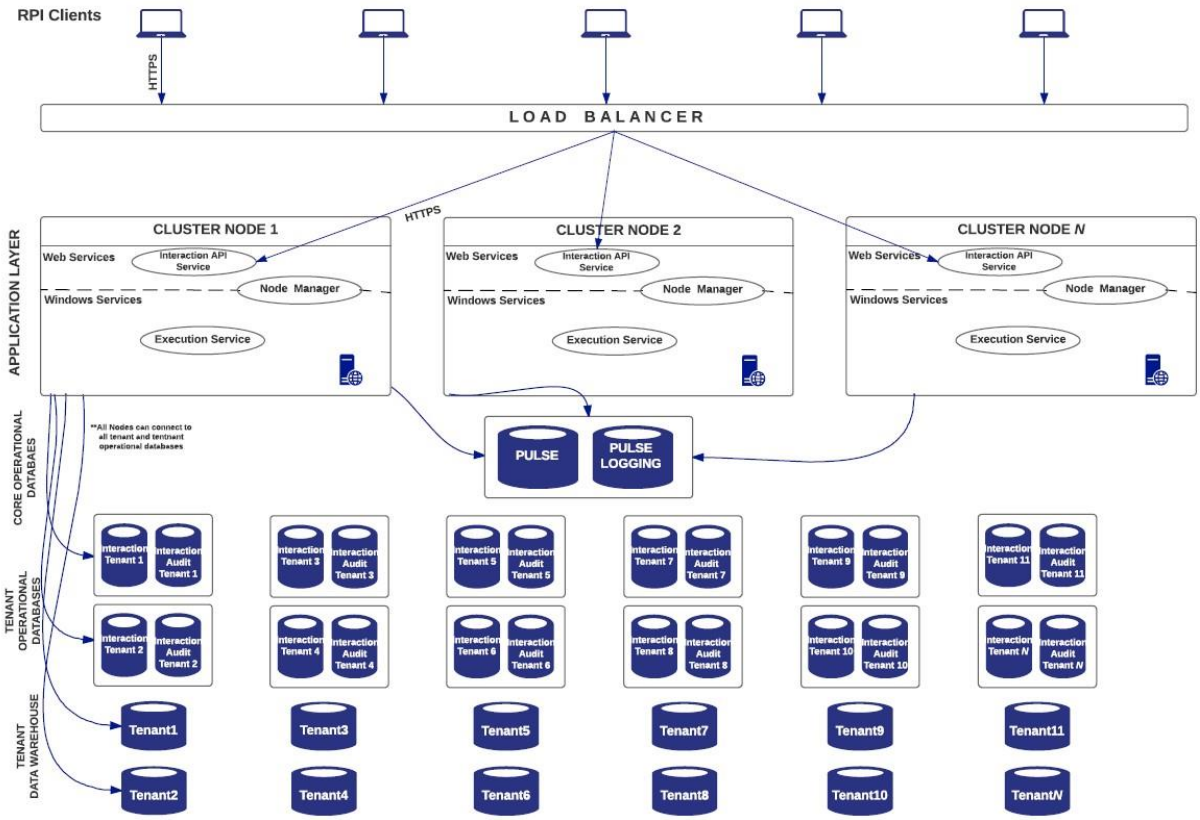
A simple RPI cluster implementation can be represented as follows:



Each of the elements in the architecture diagram is discussed separately below.

A more complex example, illustrating a multi-node, multi-tenant cluster, might appear as follows:

RPI ARCHITECTURAL DIAGRAM - MULTINODE CLUSTER



Key Concepts

Node

A single machine in a single stack or cluster implementation is referred to as a node. An RPI server installation must contain at least one node.

a. Node Manager: The node manager Windows service role is supported by all nodes within a cluster. Only one node at a time assumes the status of Master node manager; this service controls the allocation of work to Windows service roles within the cluster. In addition, the node manager service facilitates connection of the Server Workbench application itself to the cluster.

b. Node Roles: Nodes within a cluster fulfill a number of roles:

i. Windows services

- Execution service ii. Web services
- Interaction API Services
- Help files

Cluster

An RPI server installation can be installed upon several physical or virtual machines. This collection of machines, and the services (or "roles") that they provide, are collectively known as a "cluster,".

At the heart of an RPI server lies its "core". The core consists of:

1. An initial cluster node (installed on a single physical or virtual machine). By default, the initial node always supports the Node manager role.
2. Core operational databases (which can be installed upon the same or another machine).

Databases

Core Operational Databases

The operational databases store information necessary for RPI to function. There are two core operational databases:

1. Pulse: the central core operational database, which is used to persist clusterlevel operational data.
2. Pulse_Logging: stores error log records.

It is recommended that Trusted Authentication be used in preference to SQL Authentication at the core operational databases.

If Trusted Authentication is not used, database passwords might be viewable from Server Workbench and configuration files. Also, system management is made easier when using Trusted Authentication. In organizations that enforce password changes (e.g. every 6 months for SSAE-16 compliance), it is easier to only have to change the service account password rather than a number of separate connection strings.

Client Operational Databases

The following databases are created for each client installed within the cluster. Note that each client's version of the database will be assigned a unique name by the appending of a database suffix (captured at client creation).

Interaction_[suffix]: stores:

- Details of the entities created by a single client's users running the RPI client application.
- RPI users, groups and permissions.

- The RPI file system, including its folders and the files within them, such as attributes, selection rules, audiences, offers and interactions. Administration objects such as audience definitions, resolution levels, joins and database keys.

InteractionAudit_[suffix]: stores audit records created with respect to actions undertaken by a single client's users only.

It is recommended that Trusted Authentication be used in preference to SQL Authentication at the client operational databases.

Please see comments on the benefits of using Trusted Authentication above.

Data Warehouse

Each RPI client is associated with a data warehouse—a SQL Server, SQL Server PDW, Netezza, Oracle, Teradata, GreenPlum, MySQL, Sybase IQ, AWS Redshift, PostgreSQL, Actian VectorH, DB2, Splice Machine, Vertica, MariaDB, Azure SQL Database, Azure Synapse Analytics, Azure Database for MySQL, Azure Database for PostgreSQL, Google BigQuery or Snowflake marketing database that is the source of the data required to engage in conversational dialogue with customers or prospects. The data warehouse is also used as the repository within which details of the execution of such communications ("offer history") are stored.

Auxiliary Databases

Each RPI client can be configured to support one or more auxiliary databases, in addition to the main data warehouse. Data stored within auxiliary databases can be used for targeting and segmentation purposes. Note that appropriate joins between the data warehouse and auxiliary databases must exist – it is not possible to use an auxiliary database table as a resolution table, nor is it possible to write offer history data to an auxiliary database.

All of the supported database management systems suitable for data warehouse use can be used to host an auxiliary database. In addition, the Apache Hive, MongoDB (SQL), Apache Spark, Salesforce.com and Apache Cassandra platforms can be used to host auxiliary databases only.

Roles

Windows Services

At least one node in the cluster must support the Windows services role. This role provides a single Windows service: the Execution service.

The Execution service is responsible for executing server jobs generated by the undertaking of activities in the RPI client application—for example, the testing of an audience or execution of an interaction workflow. It is also responsible for the execution of asynchronous server-side jobs (e.g. catalog synchronization).

Note that, when multiple nodes expose the Windows services role, the node to be assigned the responsibility for executing a server job will be determined by the Master node manager (the job will be assigned to the Windows services node with the fewest executing server jobs).

Web Services

At least one node in the cluster must support the Web services role. This role is responsible for controlling access to the RPI server from the RPI client application. In addition, once connected, all communications between client application and server are affected by this role.

Help Files

If online help is to be made available to users, one node in the cluster can be used to support the Help files role. This role is responsible for serving online help content in a browser, as requested by RPI client application users.

Client

A client (not to be confused with the RPI client application) is an entity within an RPI server that represents a discrete and independent usage of the tool by an organization. All clients' data are stored securely and entirely independently of one another—a user who is able to access only Client A will be completely oblivious to the existence of Client B within the same server installation.

User

RPI users can be managed centrally within Server Workbench (as well as directly within the RPI client application). A user can be granted access to a single—or more than one—client and provided with appropriate permissions in each such context.

RPI Licenses

When installing the RPI server, you must specify a valid license.

If a server license expires, you can upload a new, valid license file in the Plugins tab. When you do so, the new license will be installed automatically. You will then be able to access RPI functionality as required. Note a license must be more recent than the original to be uploaded in this way.

Deployment Scenarios

Single stack (single node)

A single stack implementation is an initial node (installed on a single physical or virtual machine). By default, the initial node always supports the Node Manager role

Cluster (multiple nodes)

A cluster implementation contains an initial node plus a series of additional nodes. Each cluster node, including the initial node, exposes the Node Manager role.

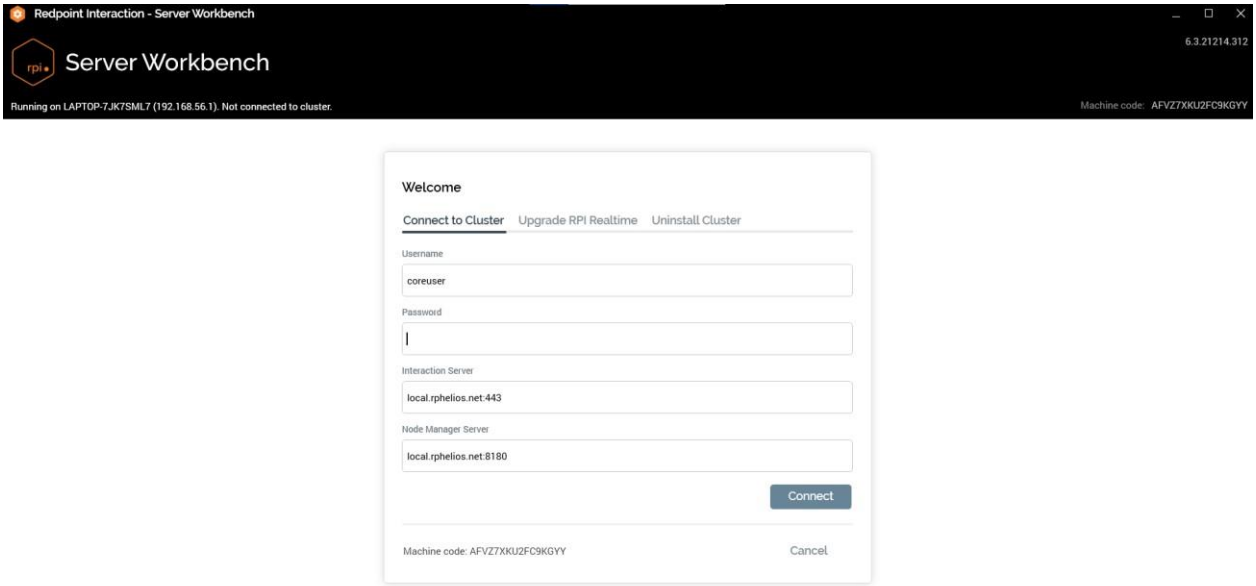
At any given time, one of the nodes is acting as the "Master Node" and each node can contain one or more roles. However, all roles must be exposed at least once in the cluster. Refer to **Roles** for more information.

Server Workbench

The RPI Server Workbench is used to set up and configure the system. The Server Workbench provides a series of dedicated interfaces within which configuration activities can be undertaken. The majority of this document is dedicated to the step-by-step use of the Server Workbench to install and configure RPI.

Double-click ServerWorkbench.exe to run Server Workbench.

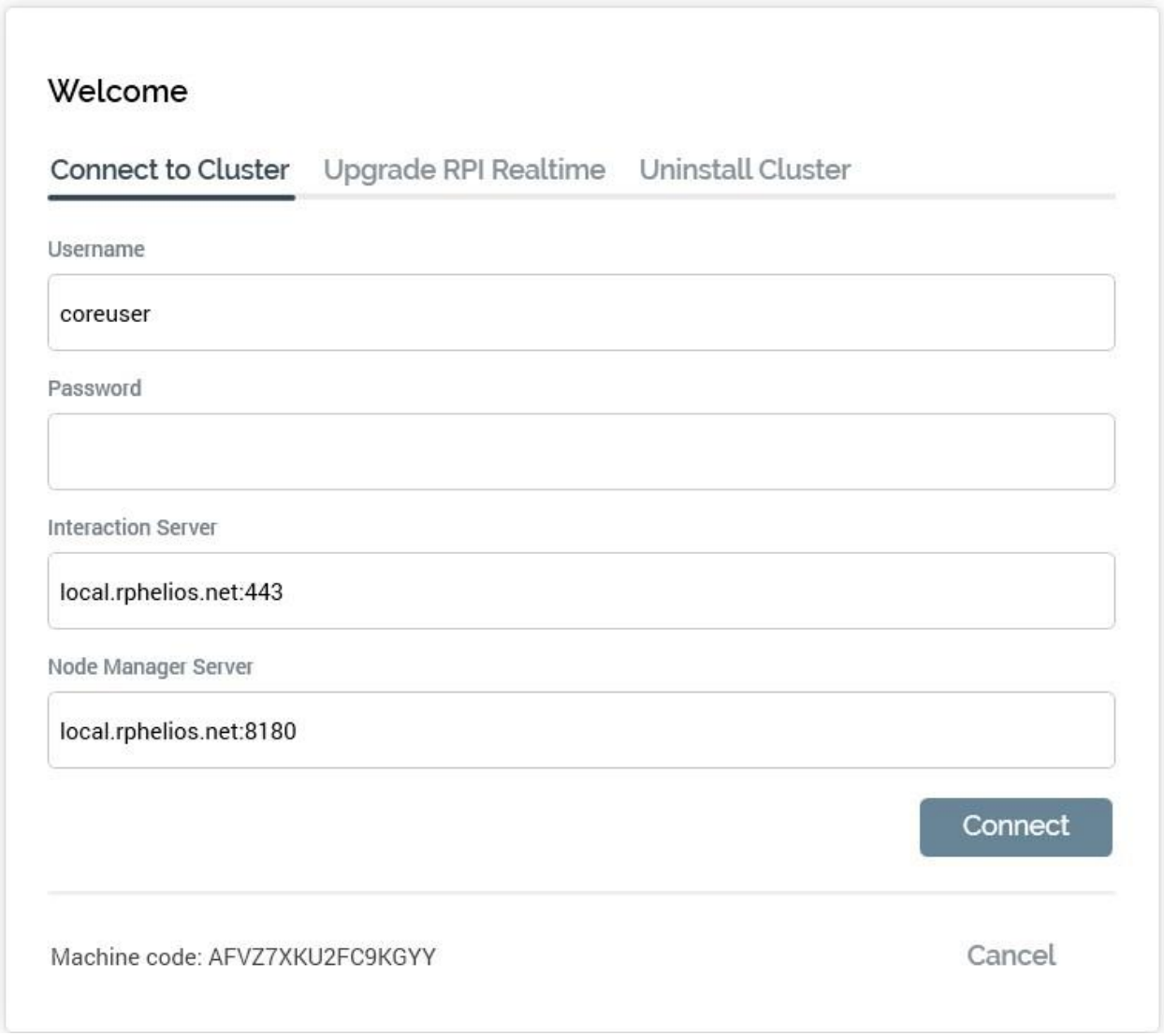
When you do so, if enabled, Windows User Account Control will confirm that you wish to run the application. If you elect to proceed, an initial interface is presented:



The header contains the following elements:

- The name and IP address of the server on which Server Workbench is running.
- The current application version.
- The current machine's license machine code. The license machine code of each physical or virtual machine in the cluster is required when generating an RPI server license. By running Server Workbench on each such machine and then right-clicking the Machine code, the license machine code can be copied to the keyboard and used during the license generation process.

The center of the interface is occupied by a "Welcome" overlay:



From here you can connect to an existing cluster, commence creating a new cluster, upgrade the RPI Realtime files or initiate the uninstallation of the existing cluster.

If you are unable to connect to a cluster, please ensure that your user account is an RPI Cluster Administrator.

Clicking the Cancel button at the bottom of the dialog closes Server Workbench.

Only one Server Workbench instance per user can be connected to a Redpoint Interaction cluster at any given time. When you attempt to log into a cluster for a second time, a message is displayed:

“The username or password is incorrect or you have logged in to this server using the same username on another instance of the Server Workbench application”.

Note that this restriction also applies if logging in multiple times at multiple VMs that are part of the same cluster.

Server Workbench Interface

Server Workbench consists of a header and a tab set.

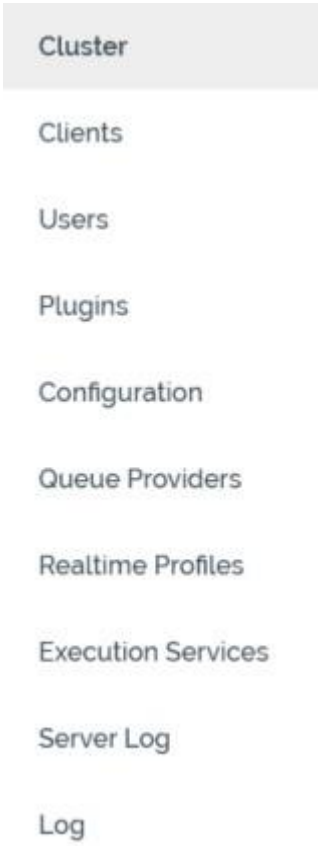
Header

The header contains the following read-only information:

- Redpoint Interaction Server Workbench [version]. The date and time of the version's build is shown in a tooltip on hovering over the version number.
- Running on [RPI server] ([IP address)].
- Logged in as [user name].
- Interaction Server
- Node Manager
- Machine code

Tab Set

The main body of the Server Workbench interface contains a tab set. The following tabs are contained within the tab set; note that not all tabs may be displayed at once.



A summary of each tab follows.

- Cluster: the Cluster tab displays details of the current cluster's nodes. The tab is always shown. It allows you to:
  - ○ Commence upgrade or validation of the core ○ Add the current machine to an existing cluster ○ Install, upgrade and validate node roles ○ Take nodes offline and put nodes online.
- Clients: the clients tab is only displayed when Server Workbench is connected to a cluster. It allows you to create, edit and delete clients within the cluster. It also allows you to assign users to a client. Clients are particularly important when RPI is running in multi-tenancy mode, with a number of discrete client organizations sharing the same set of node services.

- Users: the users tab is only displayed when Server Workbench is connected to a cluster. It allows you to create, edit and delete users within the cluster. It also allows you to assign clients to a user (if required, users can be granted access to multiple clients).
- Plugins: the Plugins tab is only displayed when Server Workbench is connected to a cluster. It allows you to upload new plugin files to the cluster to which you are currently connected, thereby exposing RPI functionality. It also allows you to apply a new license to the cluster.
- Configuration: the Configuration tab is only displayed when Server Workbench is connected to a cluster. It allows you to manage values for a series of configuration settings that apply to all clients within the current cluster.
- Queue Providers: allows you to configure queue providers at the cluster level. Typically, such queue providers will be used to host the listener queue, which supports usage of interaction queue listeners and activities.
- Realtime Profiles: allows you to create, edit and export realtime profiles, which represent the starting point for configuring RPI Realtime.
- Server Log: used to access details of server log messages generated at all clients across the cluster.
- Log: this tab is always displayed. It provides access to all log messages (information, warning and error) generated during the current Server Workbench session.

Getting Help

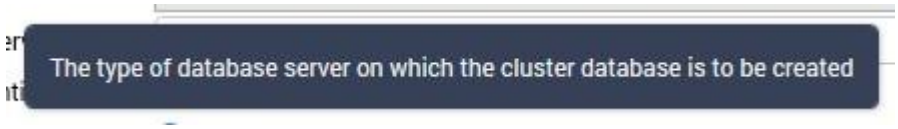
1. **Within Server Workbench**
   When performing a task in the Server Workbench (such as installing or upgrading the core or roles), a Help button is shown to the top right.

   Hovering over the button displays a tooltip.

   When you hover over a property's label, detailed information about its purpose is shown in a tooltip.

2. **Knowledgebase**
   The RPI Knowledgebase is a collection of answers to frequently asked questions, solutions to client issues, and descriptions of best practices. The Knowledgebase is accessed by logging in at **https://support.Redpointglobal.com**. The page provides a login option for individuals who have already registered or a registration option for firsttime users.
3. **Online support**
   In addition to the Knowledgebase, the Redpoint online assistance portal also provides support via forums through which Redpoint personnel and the user community can discuss issues of mutual interest. You can post a question, recommendation, or best practice or view the contributions of other experts to learn more about Redpoint products.
4. **Trouble ticket**
   If you encounter a problem that you can't resolve through use of the online help, the Knowledgebase or online support, you may enter a **Trouble Ticket** to request assistance directly from the Redpoint support staff.

Installation

The RPI server is installed using the Server Workbench application. Before running Server Workbench, please ensure that the account as which you have logged in is a member of the local Administrators group on the machine upon which the cluster core is to be installed.

Pre-Installation Preparation

Prior to installing the RPI Core, use the following checklist to ensure that all pre-requisites have been met.

Pre-Requisite	**Met (Yes/No)?**
A physical or virtual core server.	
For each additional cluster node: a physical or virtual core server.	
When deploying a multi-node environment, a load balancer is required.	
A data warehouse/marketing operational data store. *The data warehouse can reside on one of the cluster node machines; however, this configuration is not recommended.*	
All servers are joined to a domain. *Workgroup membership is neither recommended nor supported.*	
All servers are running Windows Server 2019 (recommended), Windows Server 2016, Windows 2012 R2 or Windows 10	
All servers are configured with an SSL server certificate issued by a Trusted SSL Certificate Authority. *Self-signed certificates are not recommended.* *If running in a clustered environment, a wildcard or SAN certificate are required.*	
A physical or virtual server running an instance of SQL Server 2017 (recommended), 2016, 2014 or 2012, or PostgreSQL v12, upon which the core and client operational databases will be installed. *This server can, but does not have to, be part of the RPI cluster.* *It is recommended that both the cluster node and operational databases machines are on the same domain.*	

The Server Workbench user has local administrator access to the RPI cluster servers.	
If hosting the operational databases in SQL Server, the installing user's domain account has sys admin permissions on the operational database SQL Server instance.	
If installing on non-English servers, the Server Workbench user is a member of one of the following manually-created user groups (with attendant administrative permissions): • Administrators • Redpoint Interaction Administrators	
The domain, or Network Service, account(s) used to run the nodes' Windows services have administrator access to the server(s) upon which the services are to run. *If using Network Service, the account must be added to the Administrators local group separately on each server.*	
Each cluster server has the latest .NET version (currently v4.8) installed. *If installed on a machine where IIS is installed, .NET must be installed after IIS.*	
If a server is to support the Web services and/or Help roles, the following are installed and/or enabled: • Internet Information Services (IIS) v8 or 10.	
If Web services and Help files roles are installed at more than one node and true clustered support for these services is to be made available (e.g., failover in the event of node failure), this is configured separately using an appropriate tool (e.g. IIS load balancing). *Such configuration is beyond the scope of this document.*	

The data warehouse is hosted in one of the following technologies. See The External Provider Supported Versions documentation for versions against which certified. • SQL Server • SQL Server PDW • Netezza • Oracle • Teradata	

• GreenPlum • MySQL • Sybase IQ • AWS Redshift • PostgreSQL • Actian VectorH • Splice Machine • DB2 • Azure SQL Database • Azure Synapse Analytics • Vertica • MariaDB • Azure Database for MySQL • Azure Database for PostgreSQL • Google Big Query • Snowflake • Amazon Aurora (MySQL) • SAP HANA • SybaseIQ • Yellowbrick • MongoDB (NoSQL) • Couchbase (NoSQL) • CosmosDB (NoSQL)	

If auxiliary databases are to be used, they are hosted in one of the data warehouse technologies listed above, or in one of the following: • MongoDB (SQL) • Apache Hive	

• Apache Spark • Salesforce.com • MarkLogic • Apache Cassandra • Azure CosmosDB • Apache Drill/MapR • Amazon Redshift Spectrum • Amazon Athena • Google BigTable • Google Spanner • Presto • Amazon DocumentDB (NoSQL) • Google Datastore (NoSQL)	

If hosting the operational databases in SQL Server, the database user has the following permissions: • Create Database • Create Table • Create Role • Create View …and the following permissions on any object in the operational databases: • Alter • Delete • Execute • Insert • Select • Update	

If hosting the operational databases in PostgreSQL, the database user must exist prior to installation.	
Folders required by the RPI servers have not been created in advance. *Rather, they should be created as part of the installation process; Server Workbench cannot guarantee that folder permissions will be assigned correctly if this is not the case, which may result in errors.*	
A valid RPI License is available, requested through **support@Redpointglobal.com** upon launching Server Workbench for the first time.	
The date/time between the operational database server and the cluster nodes is in sync.	
The operational databases and data warehouse have the same regional settings as the cluster nodes.	

If installing RPI on Windows Server 2012 R2, the following specific pre-requisites are in place: • IIS ○ Web Server Role • .Net Framework Features ○ HTTP Activation ○ Named Pipe Activation ○ TCP Activation *Note that, in some circumstances .NET 4 & IIS are not bound properly; best practice is to follow the instructions for the ASP.NET IIS Registration Tool. Finally, ensure that the 'Friendly name' in the Trusted SSL Certificate can be resolved by DNS (preferably), or by way of an update the clients' local hosts files.*	
A reachable SMTP server that can send emails from the RPI Server to users' email addresses is available. *Required to send out links for password set/reset.*	
The RPI service name is resolvable by all machines where the RPI client will be run and that they can access the RPI server on port 443.	

Database Preparation

Appendix A – Database Preparation describes the steps to be undertaken to ensure that databases are prepared for use by RPI as a data warehouse or auxiliary .

Client Machine Requirements

The following pre-requisites must exist on the machine on which the RPI client application is to run:

- Windows 8/8.1/10
- Microsoft .NET 4.8
- 1Gb RAM available to the RPI client application
- Internet Explorer v7+ (or Microsoft Edge)
- Microsoft Excel (optional – used to view e.g. waterfall reports)

Note that Mac users are supported through the use of a local virtual machine, Citrix, or another remote desktop solution.

Installing the Core

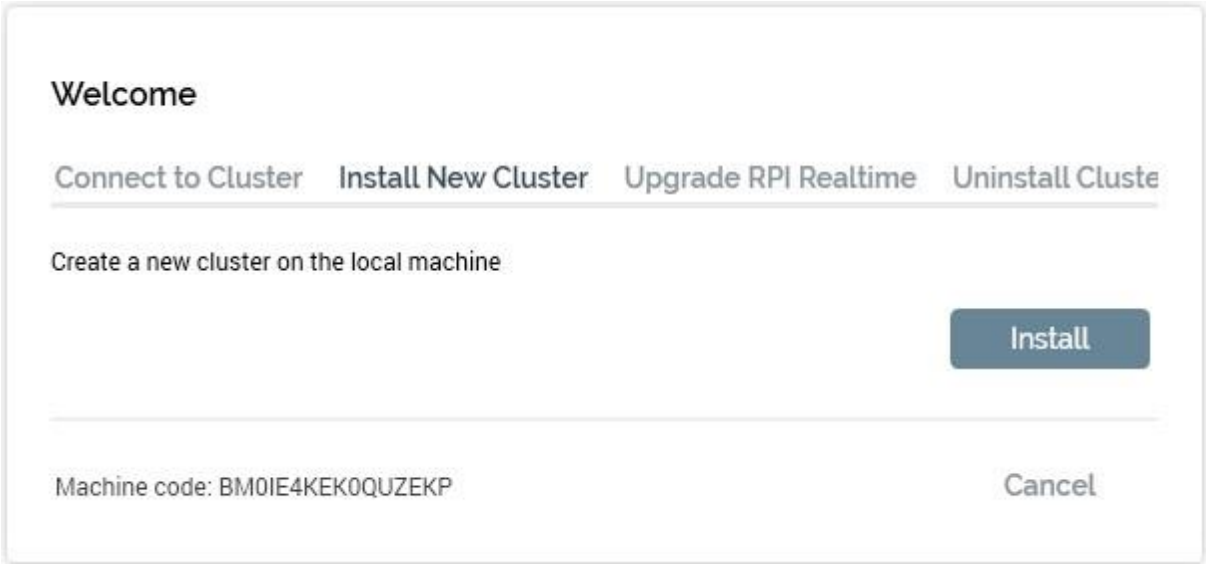
When installing the core, Server Workbench must be run on the machine upon which the initial cluster node is to be installed.

Whether the deployment will involve a single node or multiple nodes, an initial machine must have the core installed and validated and then roles installed. Each additional node in a multiple node deployment will then follow the steps described in the **Cluster** section.

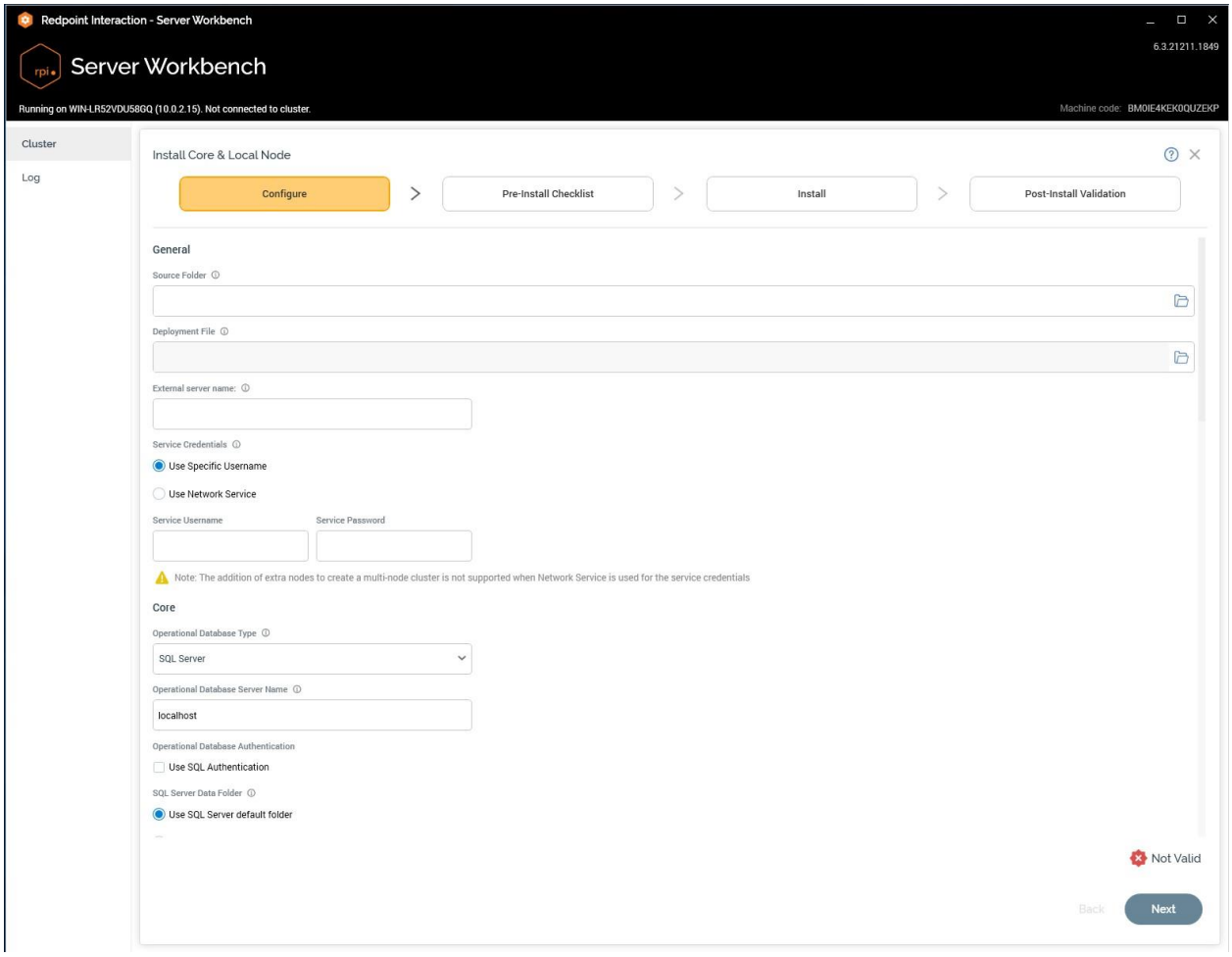
Note that installing the RPI Core also automatically installs the Web Services role; this is required, as Server Workbench itself needs a connection to the RPI web services prior to installing roles.

Creating a Cluster

If the machine upon which you are running Server Workbench does not already have a cluster core installed, you can create a cluster there. You start this process by clicking the Install button, which can be found in the Install New Cluster tab in the Welcome overlay:



The main Server Workbench interface is then displayed, with the Install Core overlay shown over the Cluster tab:



Install Core & Local Node Section

Installation of the core is carried out in a wizard style, using a series of sub-interfaces as follows:

1. Configure
2. Pre-Install Checklist
3. Install
4. Post-Install Validation

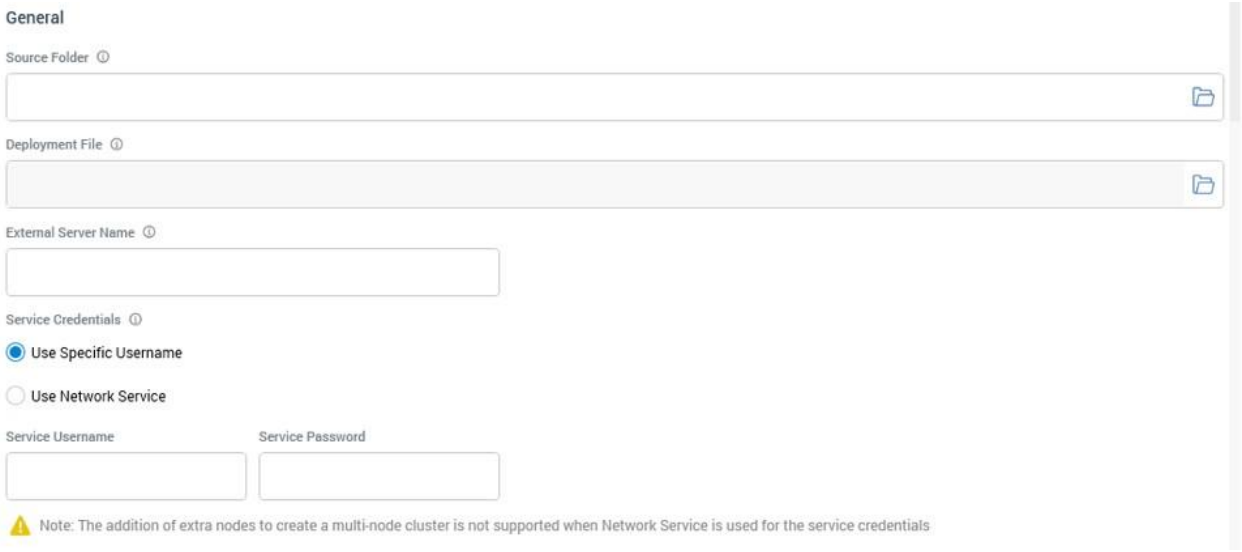
A graphical representation of current progress through the wizard is displayed within the top section:

Configure Sub-interface

The Configure sub-interface allows you define a series of properties required by Server Workbench in advance of installing the core. It consists of the following sections.

General Section

This section allows you to define a basic set of properties relating to the installation:



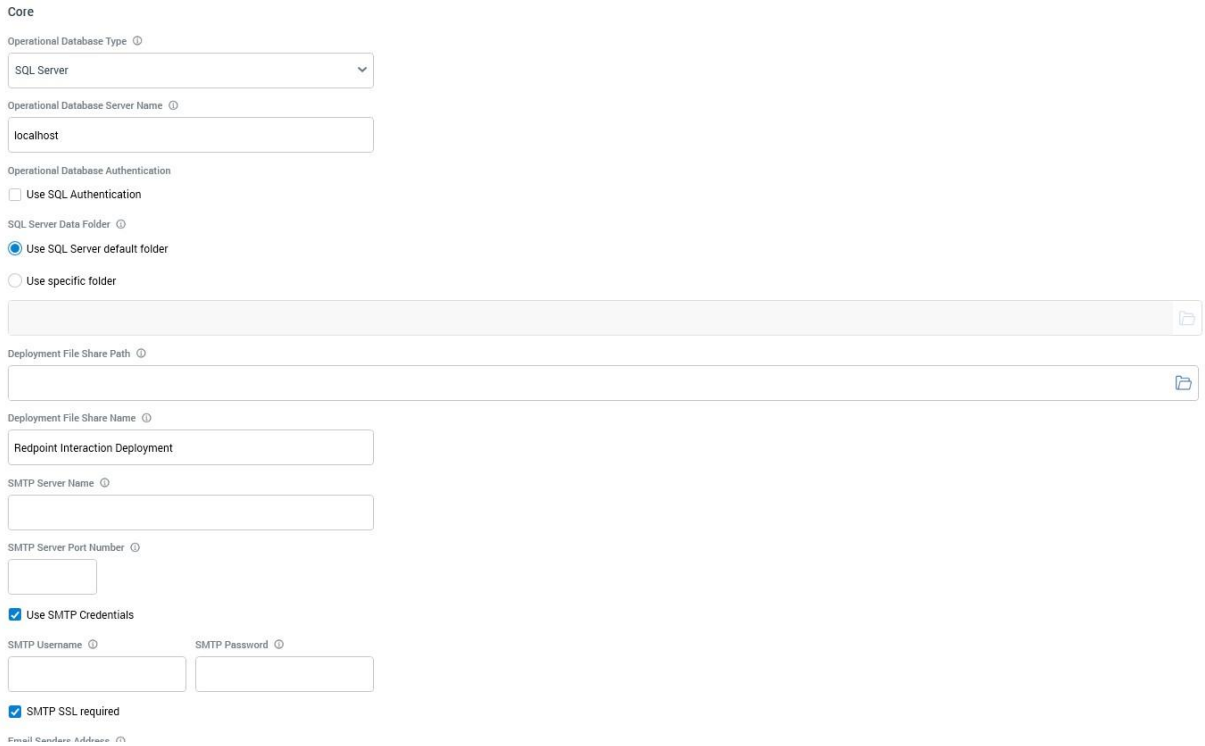
- Source Folder: the Windows file system folder in which the files used to perform the installation are to be found. You can specify this value manually or by browsing for a folder.
- Deployment File: an XML file that defines the tasks to be undertaken during installation. This property defaults to "RPDeployment.xml" if that file can be found within the Source folder.
- External Server Name: the fully-qualified name, including any certificate extension, of the initial node's machine (i.e., the machine upon which Server Workbench is currently running).
- Service Credentials: two radio buttons are available:
  - Use Specific Username: selected by default. When chosen, you can provide a domain Service Username (with sufficient privileges) that will be used to run the initial node's Node manager service. An optional Service Password can also be supplied.
    Note that you cannot use the '.\username' syntax in this context. All user names must be fully declared. Active Directory-style names (e.g. 'username@Redpoint' are supported).
  - Use Network Service: when selected, the initial node's Node manager service will run under the current machine's Network Service account. The account must be a member of the local Administrators group.
    A warning message is displayed if you select this option.

If you intend to create a multi-node cluster, you should not use Network Service as the service credentials.

Core Section

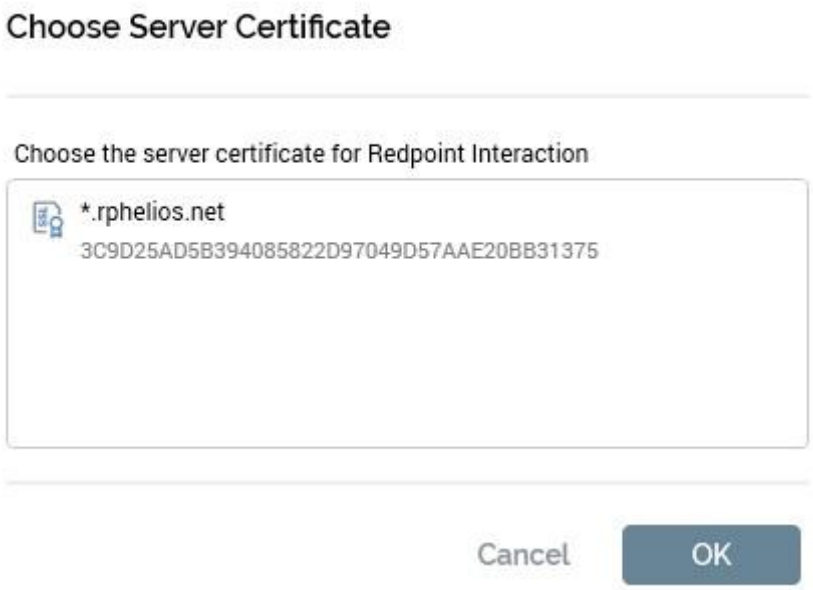
This section allows you to capture fundamental properties of the core.



- Operational Database Type: this dropdown field defaults to SQL Server. You can optionally elect to specify that the operational databases be installed using the Azure SQL Database, Amazon RDS SQL, Google Cloud SQL or PostgreSQL provider.
- Operational Database server name: the name of the server on which the core operational databases are to be installed.
- Database Port Number: only if hosting the operational databases in PostgreSQL.
- Operational Database Authentication: contains a Use SQL Authentication checkbox, which is unchecked by default. When unchecked, connection will be made to the operational databases using Windows authentication. When checked, an SQL Server login username and optional password can be provided instead (the <,>,'," and & characters are not supported at passwords). If using an Azure SQL operational database, leave this property unchecked to use Azure Managed Identity.
- SQL Server Data Folder: allows you to define a folder where the operational databases' data will be persisted, if using SQL Server. Two radio buttons are provided ○ Use SQL Server default folder: selected by default.

- Use specific folder: selection of this option allows you to specify a folder in which operational data will be stored. When selected, a label is shown to confirm that the folder exists on the operational database server.

- Deployment File Share Path: this field is used to capture the physical path of a shared folder that, if it does not exist already, is created during installation of the core. The share is used when adding further elements—other nodes and node roles—to the cluster.
- Deployment File Share Name: accompanying Deployment file share path, this field is set by default to “RPI Deployment”. It allows you to define the name by which the share will be known.
- SMTP Server Name: the name or IP address of a server that will be used to manage operational email communications within the cluster.
- SMTP Port Number
- Use SMTP Credentials: this checkbox allows you specify whether to use credentials when connecting to the SMTP server. If checked, SMTP Username (mandatory) and SMTP Password (optional) fields are also shown. An SMTP SSL required checkbox, unchecked by default, is also displayed; you can check this if secure sockets are to be used in when connecting to the SMTP server.
- Email Sender's Address: allows you to specify the email address from which operational communications from the RPI server will be received.
- License File: you can browse for a valid RPI server license with which to populate this field. Alternatively, you can provide the license’s path manually. Details of the usage of RPI server licenses are provided elsewhere.
- License Machine Code: another read-only representation of the current machine’s unique license machine code value.
- Certificate Hash: you can click Choose certificate hash to display the Choose Server Certificate dialog:

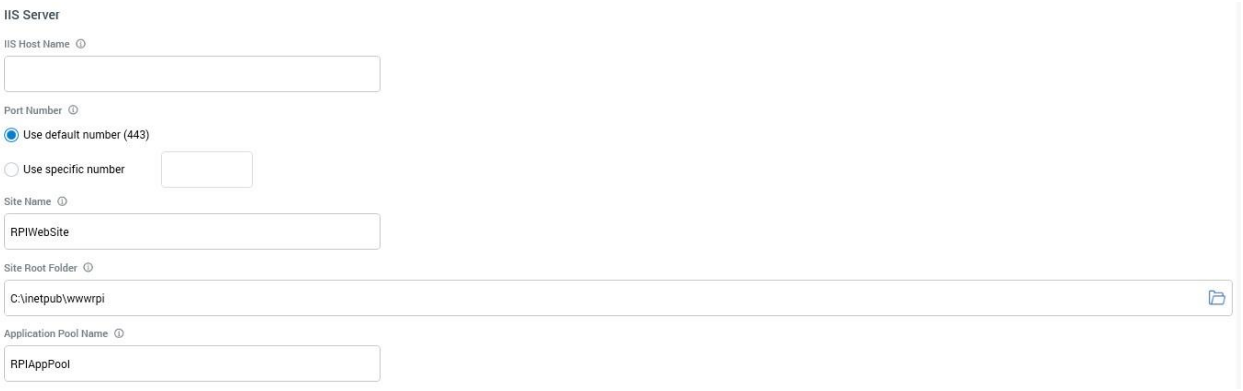


On selection of a certificate, its unique hash value is displayed in the field. Alternatively, you can specify a certificate hash value manually.

- Key Vault Type: this dropdown field allows you to specify that the new RPI Installation will make use of Microsoft Azure KeyVault to securely persist credentials used by the application. A dropdown, it exposes the following values:
  ○ None: the default (don't use KeyVault) ○ Azure: use KeyVault. If selected, the Key vault URI property is displayed.
- Key Vault URI: this property is shown when Key vault type is set to 'Azure'. It allows you to specify the URI of the Azure KeyVault to use to protect RPI application credentials. It is mandatory if shown.
- 'coreuser' Cluster Administrator Password: a read-only password for the default 'coreuser' Cluster Administrator user is generated automatically. **You must take note of this value, as you will need it to log into the new RPI installation as coreuser.** Having logged in, the value can be changed to something more memorable.

IIS Server Section

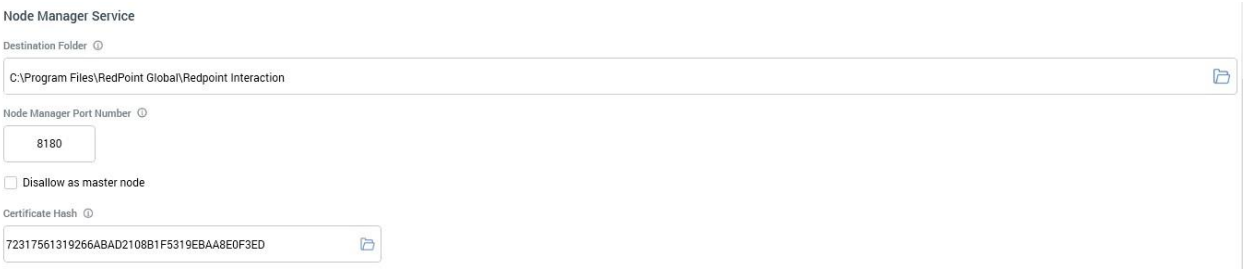
This section is used to specify the Internet Information Server (IIS) to be used by the RPI cluster.



- IIS Host Name: this field should be set to the fully-qualified (including certificate extension) name of the public-facing machine that will host the cluster's web services. This field is used when connecting to the RPI server using the RPI client application. IIS host name is of particular significance when being used within a cluster with the required IIS load balancer; in this case the public name of the load balancer server, rather than the name of any specific web service machine, needs to be specified here.
- Port Number: you can specify the port through which communication will be made with the cluster's web services here. Two radio buttons are provided: ○ Use default number (443): selected by default.
  - ○ Use specific number: selecting this option enables a field in which you can specify a specific port number.
- Site Name: the name of the IIS site.
- Site Root Folder: the IIS site's root folder.
- Application Pool Name: the app pool to be used by the RPI IIS application.

Node Manager Service Section

This section covers properties relating to the cluster's Node Manager service.



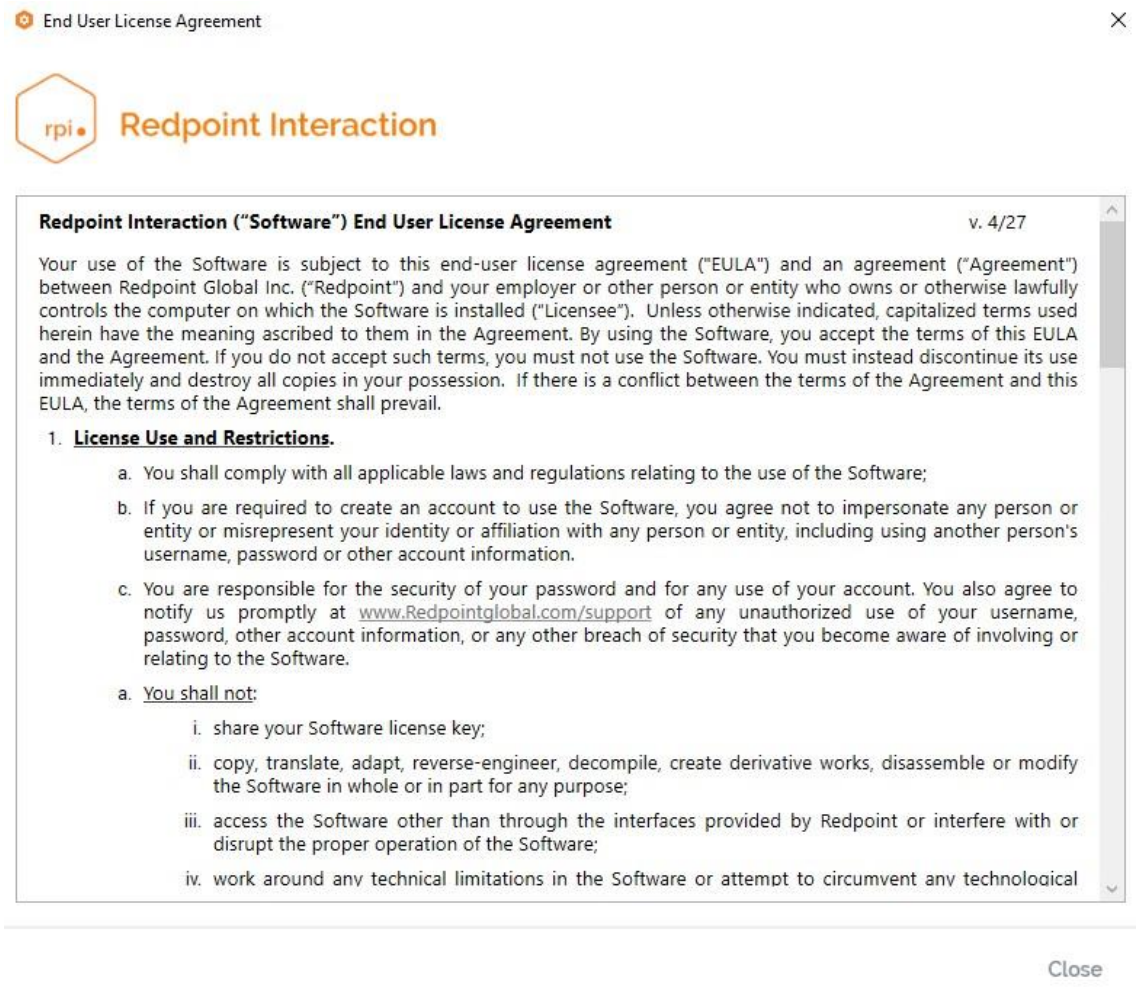
- Destination Folder: this field defaults to "C:\Program Files\Redpoint Global\RPI". It defines the folder in which the initial node's Node manager service will be installed.

- Node Manager Port Number: by default, port 8000 is used to access the Node manager service; however, you can change this value if required.
- Disallow as master node: this checkbox is unchecked by default. When unchecked, the current node can assume master node status. When checked, the current node cannot assume master node status, unless explicitly allowed through use of the context menu option shown when right clicking the node.
- Certificate Hash: as documented above. Set automatically to the value selected in the previous instance of the property.

End User License Agreement Section

The final section covers the RPI End User License Agreement (EULA).

- Check this box to accept the Redpoint Interaction End User License Agreement: this checkbox is unchecked by default. You must accept the EULA before proceeding with the installation.
- View Redpoint Interaction End User License Agreement: clicking this button displays the End User License Agreement in a separate Window.



You can close the Window using the button at the bottom right.

Validation

If you have not correctly completed the mandatory fields within the Configure sub-interface, a validation errors indicator is displayed to the bottom right:

Clicking the indicator displays details of the errors in a dialog.

You cannot progress to the Pre-Install checklist so long as outstanding validation errors remain.

If the configuration is valid, an indicator advises accordingly:

You may now move to the next wizard step.

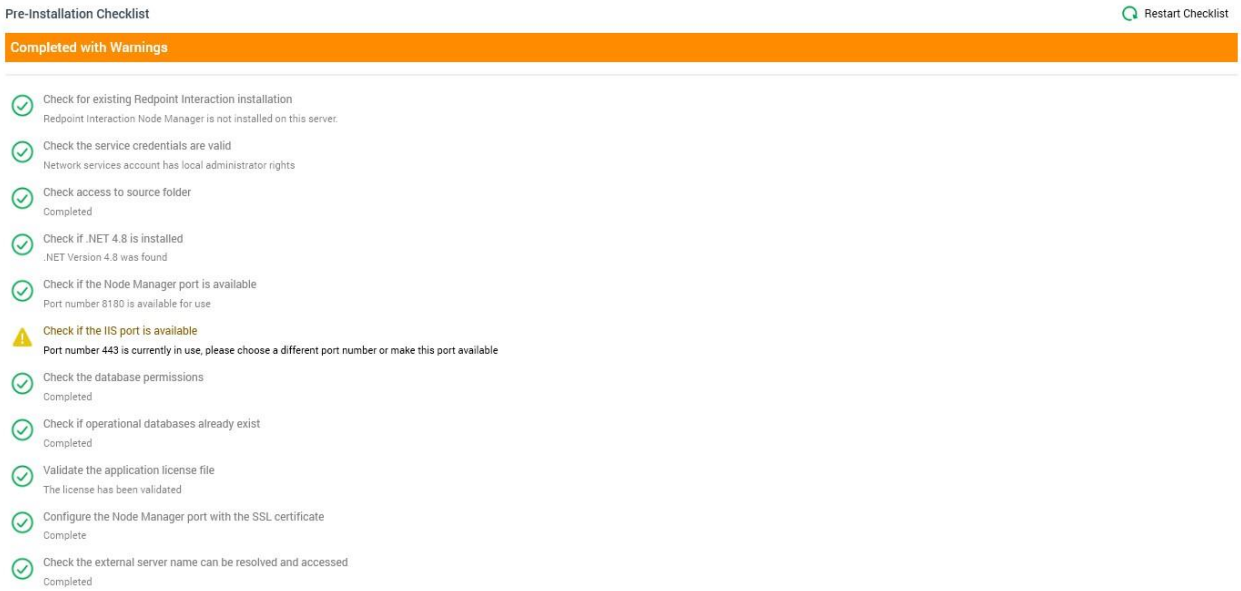
Navigation

You can move forwards through the Install Core wizard by clicking on the Next button at the bottom right. Doing so displays the Pre-Install Checklist sub-interface.

A message ('You must accept the End User License Agreement before installing the core services') if shown if you click Next without accepting the RPI EULA.

Pre-Install Checklist Sub-interface

The purpose of this interface is to allow a series of checks to be performed to ensure that all pre-requisites have been met prior to proceeding with the install.



Toolbar

A toolbar, shown above the Pre-Installation Checklist section, exposes a single button:

- Restart Checklist: clicking this button performs the pre-install validation checks again. In the event of a failed check, you can return to the Configure sub-interface, rectify any configuration issues and re-run the pre-install checks from the Pre-Install Checklist.

Pre-Installation Checklist

A series of checks are performed prior to installation of the core.

In the event of a failure, you can return to the Configure interface to resolve the issue in question before returning to the Pre-Install Checklist.

Some checks may result in the display of a warning. In the event of a warning, you are still at liberty to progress with the installation; however, you need to take into account any accompanying messages.

During execution of the pre-install checks, a status bar is shown at the top of the checklist.

On their completion of the checks, the progress bar is replaced by an overall status summary (Completed, Failed or Completed with Warnings).

In addition, a message is displayed at the top of the Server Workbench on completion of the checklist.

Navigation

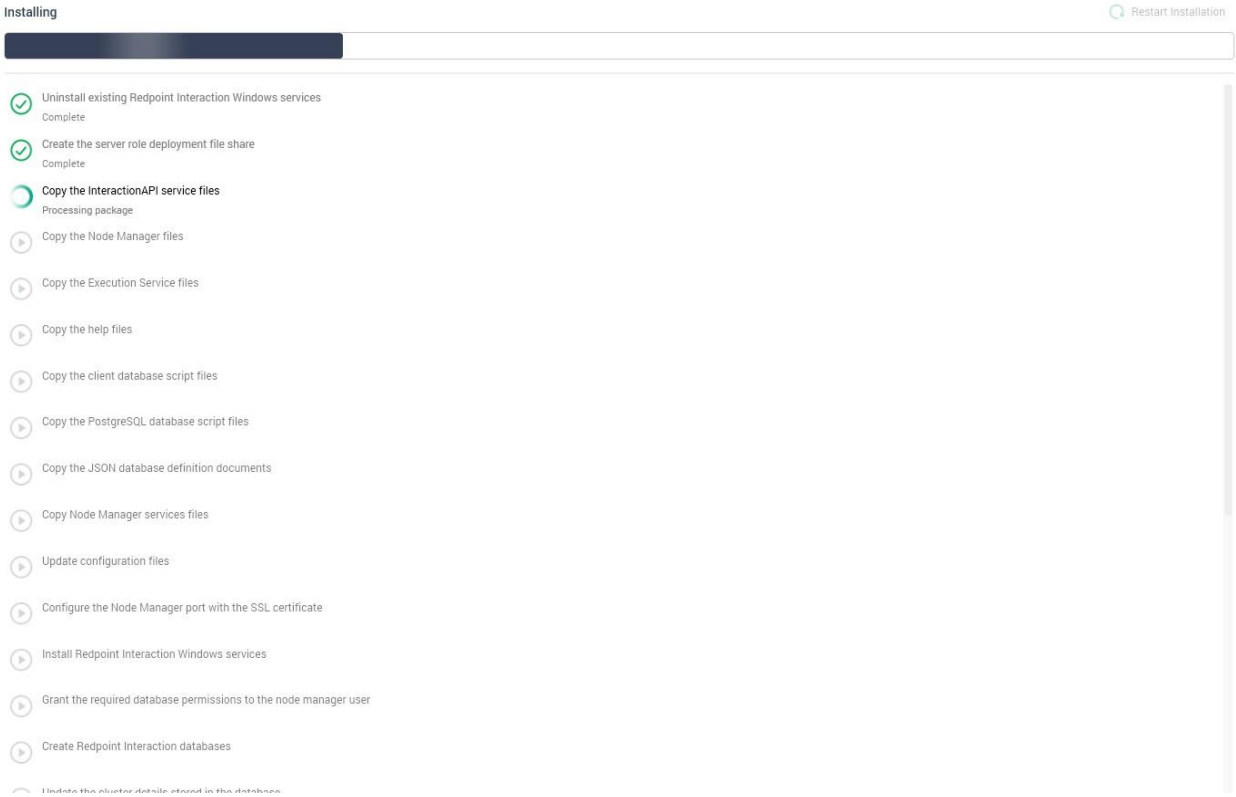
On completion of the checklist, you can continue your progress through the Install Core wizard by clicking on the Next button at the bottom right. Access to the Install sub-interface is controlled by an "Are You Sure?" dialog.

If you attempt to access the next interface prior to completion of the checklist, you will be advised to wait until all steps are complete.

You can return to the Configure sub-interface using the Back button.

Install Sub-interface

This interface is used to provide visibility of the tasks executed during installation of the core.



Toolbar

A single button is shown in the toolbar:

- Restart Installation: this button is available when the installation's status is one of "Completed with Warnings" or "Has a Dependent Job with Errors". Restart Installation allows you to repeat the core installation steps from the beginning.

Installation Steps

Details of status information provided at each installation step and overall progress are provided in the Core Pre-Install Checklist documentation.

Note that installing a new core automatically installs the Web Services role at the cluster's initial node.

Navigation

On completion of the installation, you can continue your progress through the Install Core wizard by clicking on the Next button at the bottom right.

If you attempt to access the next interface prior to completion of the installation, you will be advised to wait until all steps are complete.

You can return to the Pre-Install Checklist sub-interface using the Back button.

Post-Install Validation

A series of checks are performed following installation of the core to verify that the exercise was completed successfully.

Clicking the Restart Validation button allows to you repeat the post-install validation checklist Coreuser

On initial cluster commissioning, a default user, with the following properties, is created:

- Username: “coreuser”
- Full Name: “System administrator”
- Email: “notset@default.com”
- Password: “.Admin123”

Navigation

On completion of the Post-Install checklist, the Next button, displayed at the bottom right, is replaced by a Finish button. Clicking Finish closes the Install Core overlay and displays the Server Workbench tab set, which now includes all tabs. The Cluster tab is shown by default and contains a single node upon which the Node manager role is installed.

If you click Finish prior to completion of the checklist, you will be advised to wait until all steps are complete.

You can return to the Install sub-interface using the Back button.

Closing Prior to Completion

If you close the Install Core overlay prior to completion of the wizard, the main Server Workbench tab set, containing on the Cluster and Log tabs, is displayed. A message is shown at the Cluster tab, advising you that ‘You have not yet connected to an RPI Cluster. Click ‘Connect’ to do so, or to start installing a new Cluster Core’. A button accompanies the message; clicking the Connect / Install Cluster Core button redisplays the Connect dialog.

If the core had been successfully installed prior to the Install Core overlay’s closing, you can click Connect to connect to the new cluster; if not, you can re-commence the installation process by clicking Install Core.

Installing Roles

When installing roles, Server Workbench must be run on the machine where the roles are to be installed.

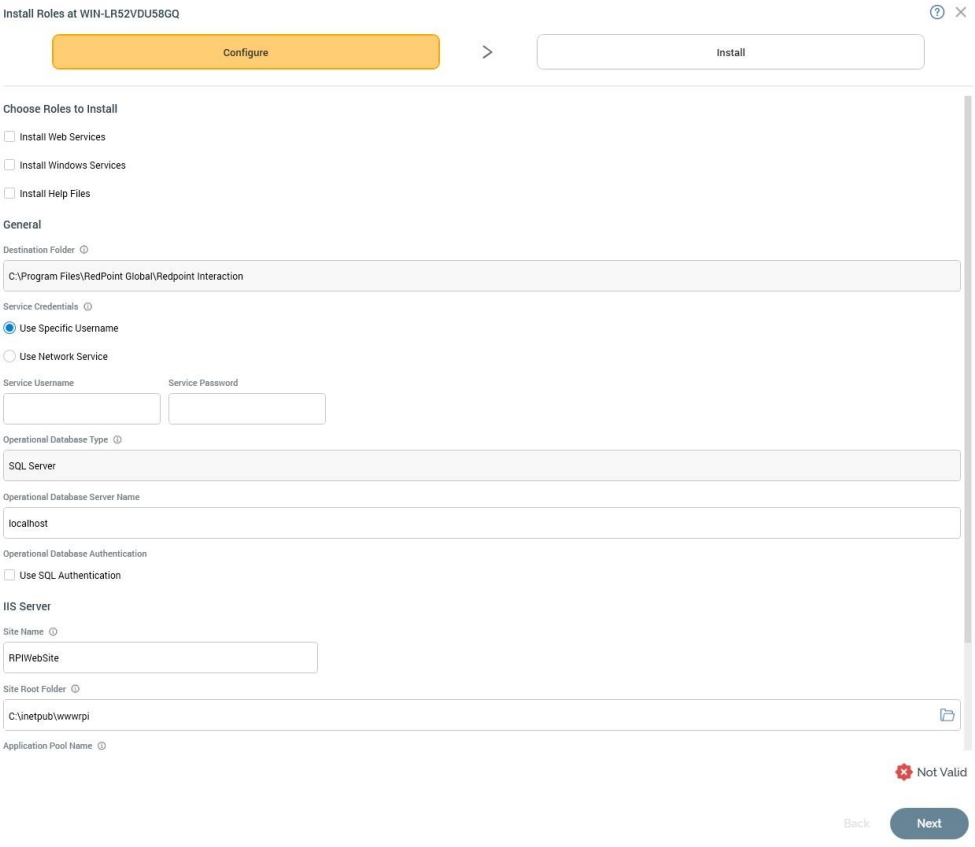
Every role within an RPI server cluster supports the Node manager role. In addition, each role can support one or more of the following roles:

- Web services
- Windows services
- Help files

The Web services role is automatically installed at a new cluster core's initial node.

Having created an initial node (by installing the core), or having added and installed a local node, you can start the process of choosing and installing addition roles by clicking the Install Roles context menu button, displayed when you hover over the node.

When you do so, the Install Roles overlay is shown:



Install Roles at [Server Name] Section

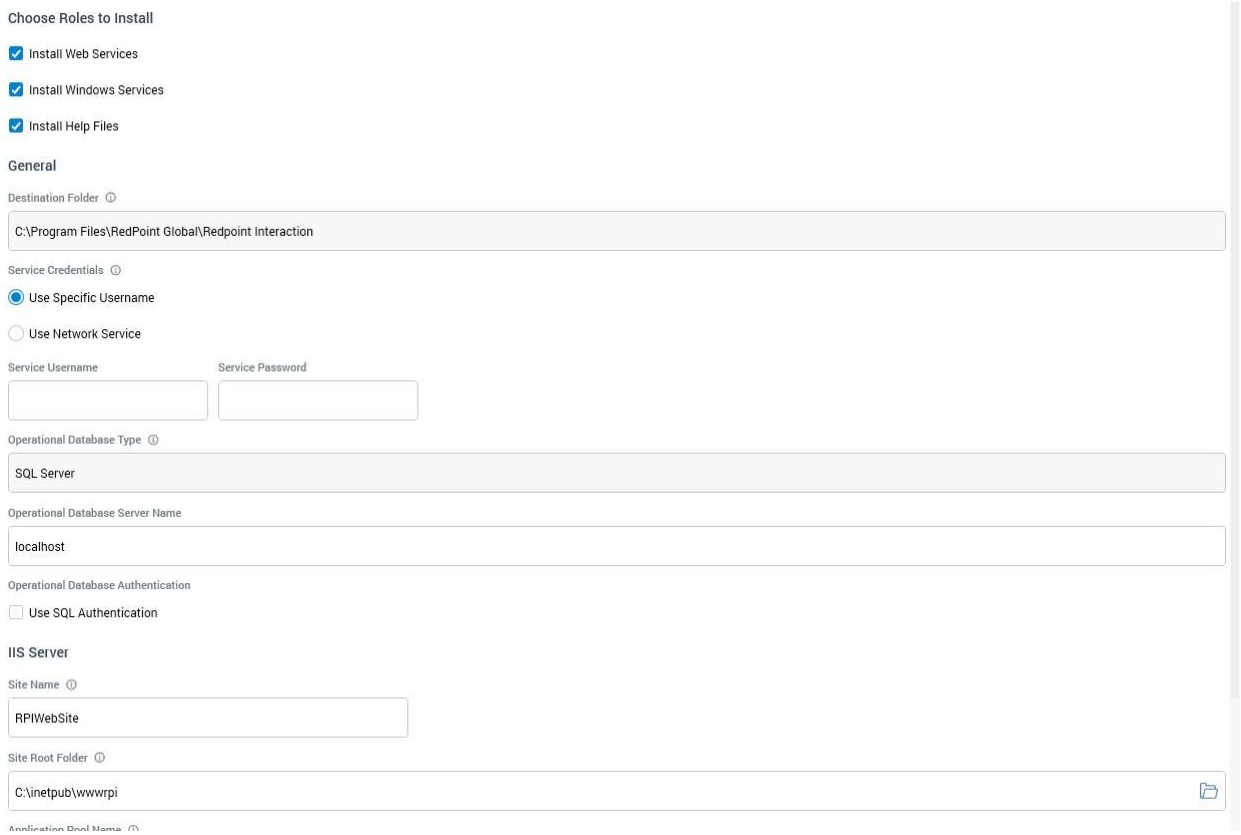
Installation of node roles is carried out in a wizard style, using two sub-interfaces:

- Configure
- Install

A graphical representation of current progress through the wizard is displayed within the top section:

Configure Sub-interface

The Configure sub-interface allows you define a series of properties required by Server Workbench in advance of installing node roles:



It consists of Choose Roles to Install, General and IIS Server sections.

Choose Roles to Install

This section allows you to choose which roles you wish to install at the node. A series of checkboxes, all unchecked, are displayed, and you can check the roles to be installed.

- Install web services
- Install Windows services
- Install help files

Note that, if installing roles at the initial cluster node, the web services role will already have been installed. If you attempt to install a service that has already been installed, you will receive an error.

General

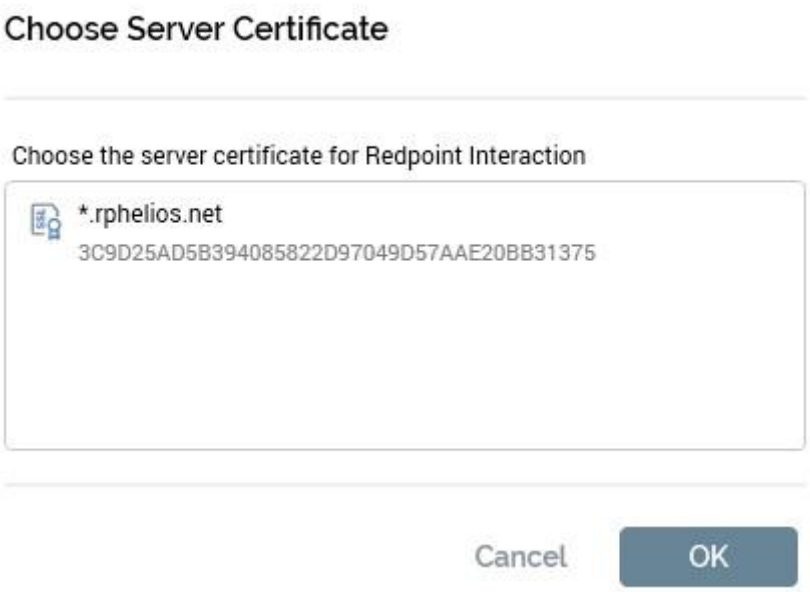
This section displays a series of properties required prior to commencing role installation:

- Destination folder: this read-only field is set to the same path specified when installing the Node Manager service on the node in question.
- Service credentials: two radio buttons are available:
    - Use specific username: selected by default. When chosen, you can provide a domain username (with sufficient privileges), that will be used to run the Windows services role. An optional Password can also be supplied.
    - Use Network Service: when selected, Windows services will run under the current machine's Network Service account. The account must be a member of the local Administrators group.
- Operational database type: read-only; defined when installing the core.
- Ops. database server name: the name of the server on which core operational databases are installed.
- Use SQL authentication: when unchecked, operational database connection will be made using Windows authentication. When checked, a login username and optional password can be provided instead.

IIS Server Section

The fields within this section are only displayed when Install web services and/or Install help files is/are checked:

- Site name: defaults to 'RPIWebSite'
- Site root folder: defaults to 'C:\inetpub\wwwrpi'
- Application pool name: defaults to 'RPIAppPool'
- Certificate hash: you can click Choose certificate hash to display the Choose Server Certificate dialog:



On selection of a certificate, its unique hash value is displayed in the field. Alternatively, you can specify a certificate hash value manually.

Validation

If you have not correctly completed the mandatory fields within the Configure sub-interface, a validation errors indicator is displayed to the bottom right:

Clicking the indicator displays details of the errors in a dialog.

You cannot progress to the Install interface when outstanding validation errors remain.

If the configuration is valid, this indicator is shown:

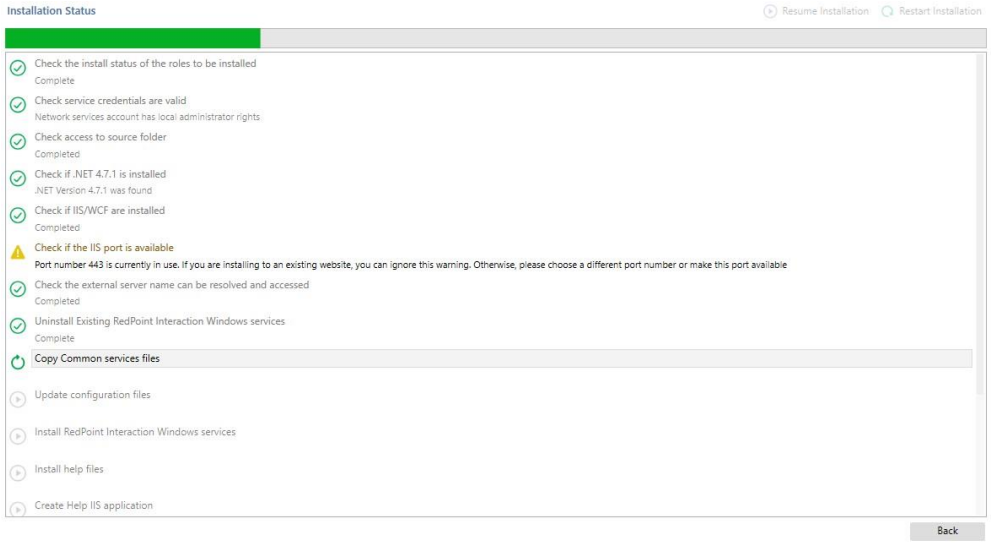
You may now move to the next wizard step.

Navigation

You can move forwards through the Install Roles wizard by clicking on the Next button at the bottom right. Doing so displays the Install sub-interface.

Install Sub-interface

This interface is used to provide visibility of the tasks executed during installation of node roles.



Toolbar

Two buttons are shown at the toolbar:

- Resume Installation: this button is available when the installation's status is Has Been Paused on Warnings. It allows you to carry on the installation at the point where the warning occurred.
- Restart Installation: this button is available when the status is one of Completed with Warnings or Has a Dependent Job with Errors. It allows you to repeat the installation steps from the beginning.

Installation Steps

A series of steps are performed to install the selected roles. The exact steps undertaken depend on the roles being installed.

Navigation

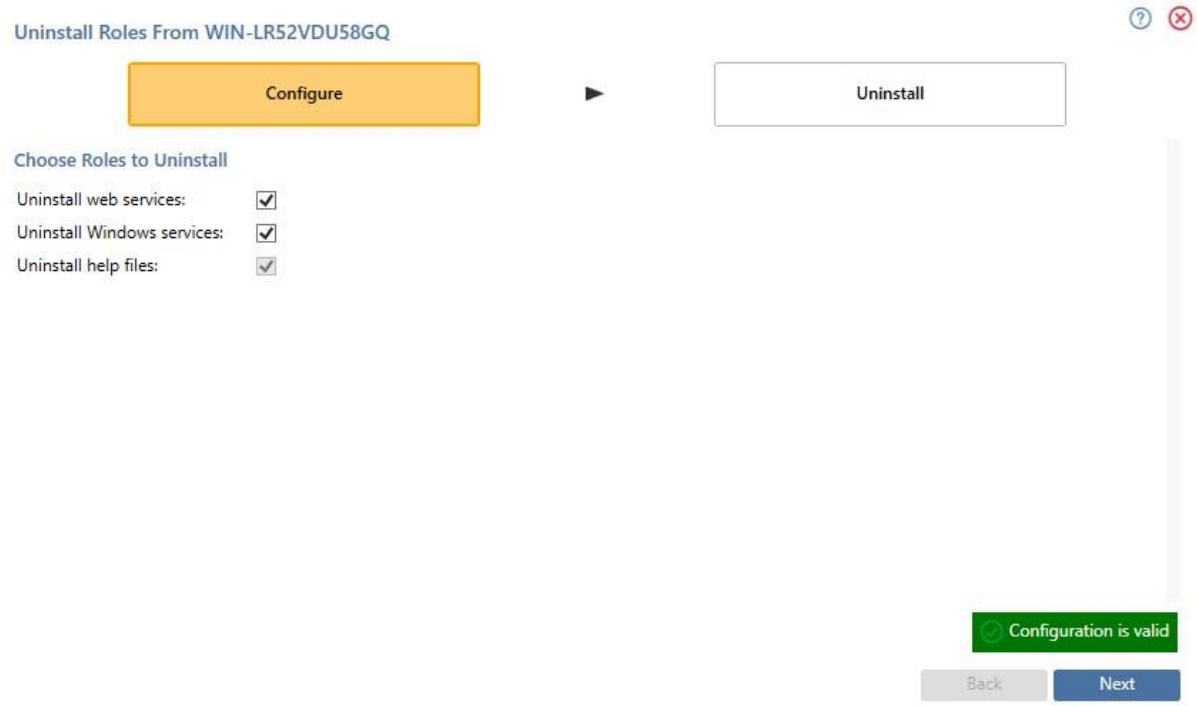
On completing the installation, the Next button, displayed at the bottom right, is replaced by a Finish button. Clicking Finish closes the Install Roles overlay and displays the Server Workbench tab set, which includes all tabs. The node at which the role(s) was/were installed is shown as Online, and its new roles are displayed therein.

If you click Finish prior to completing the installation, you will be advised to wait until all steps are complete.

You can return to the Configure sub-interface using the Back button.

Uninstalling Roles

You can invoke the uninstallation of one or more roles at a node using the Uninstall roles context menu option, displayed on clicking a node within the Cluster tab. When you do so, the Uninstall Roles overlay is displayed.



Uninstall Roles at [Server Name] Section

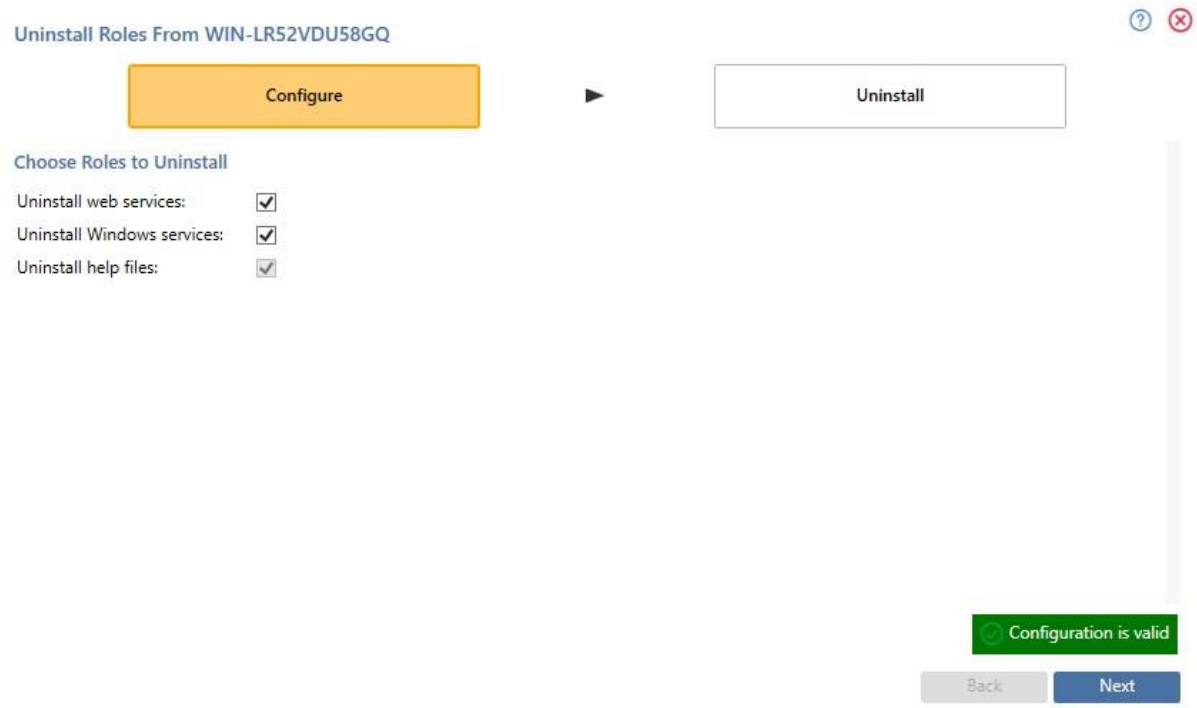
Uninstallation of node roles is carried out in a wizard style, using two sub-interfaces:

- Configure
- Uninstall

A graphical representation of current progress through the wizard is displayed within the top section:

Configure Sub-interface

The Configure sub-interface allows you to select the role(s) you wish to uninstall.



It contains a single section – Choose Roles to Uninstall.

Choose Roles to Install

This section allows you to choose which roles you wish to uninstall. Up to three checkboxes are shown (depending on the roles currently installed), and all are unchecked by default:

- Uninstall web services: checking this option automatically checks Uninstall help files. **Note that you are advised not to uninstall the Web services role at a single-node cluster**.
- Uninstall Windows services
- Uninstall help files: this option cannot be unchecked if Uninstall web services is checked.

It is mandatory to check at least one checkbox before proceeding.

Validation

If you do not select at least one role within the Configure sub-interface, a validation errors indicator is displayed to the bottom right:

Clicking the indicator displays details of the errors in a dialog. You cannot progress so long as outstanding validation errors remain.

If the configuration is valid, this indicator is shown:

You may now move to the next wizard step.

Navigation

You can move forwards through the Uninstall Roles wizard by clicking on the Next button at the bottom right. Doing so displays the Uninstall sub-interface.

Uninstall Sub-interface

This interface is used to provide visibility of the tasks executed during uninstallation of node roles.

Toolbar

A single button is shown at the toolbar:

- Restart Uninstall: this button is available when the status is one of Completed with Warnings or Has a Dependent Job with Errors. It allows you to repeat the installation steps from the beginning.

Uninstallation Steps

The exact steps taking during role uninstallation depend on the roles that are being uninstalled.

Navigation

On completing the uninstallation, the Next button, displayed at the bottom right, is replaced by a Finish button. Clicking Finish closes the Uninstall Roles overlay and displays the Server Workbench tab set, which includes all tabs.

If you click Finish prior to completing the installation, you will be advised to wait until all steps are complete.

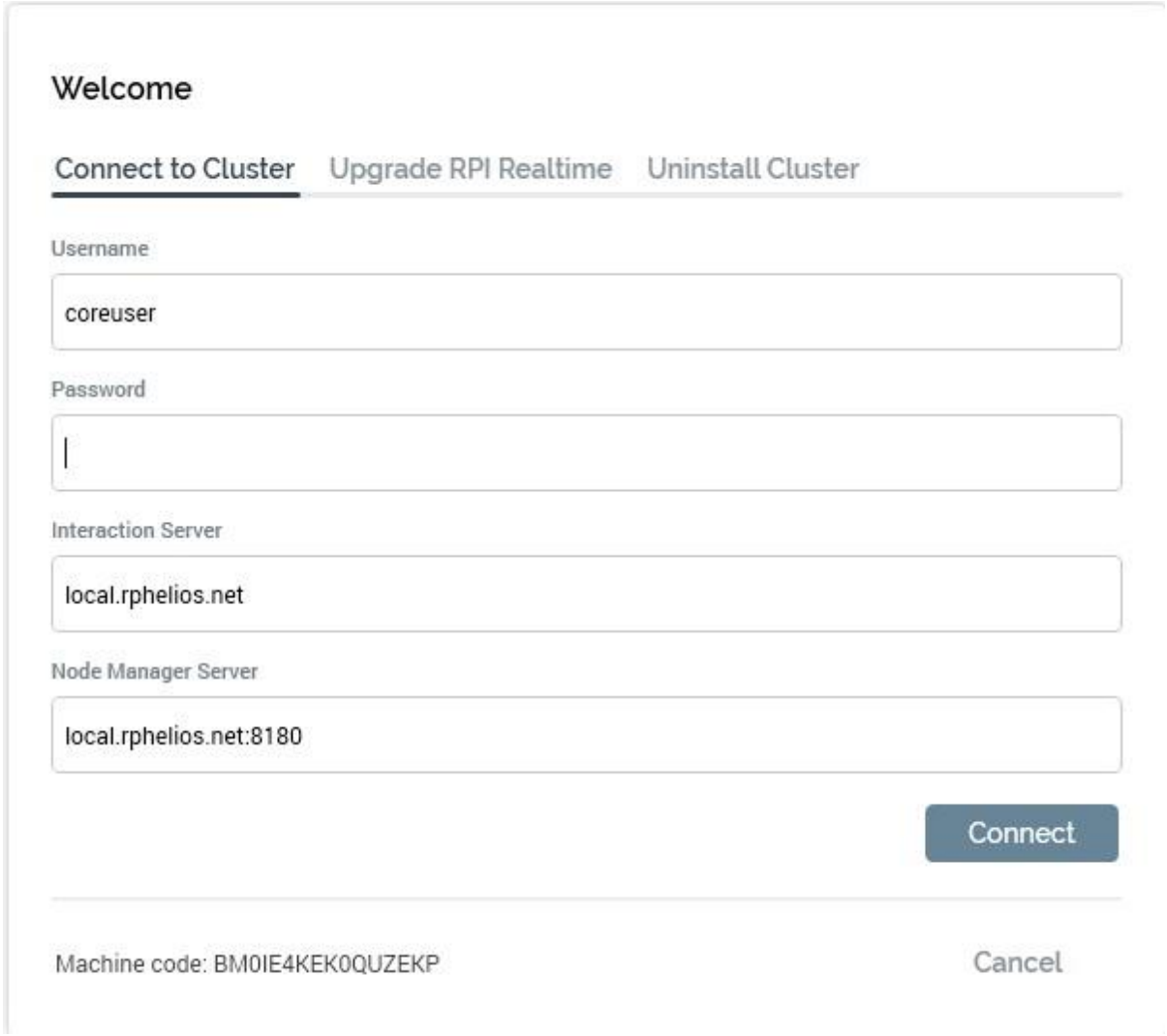
You can return to the Configure sub-interface using the Back button.

Cluster

You must install the core before adding additional cluster nodes. Refer to **Installing the Core** for instructions on deploying the initial cluster node. For each additional node in the cluster, you will add the local node (while logged into the node) and install roles (refer to **Installing Roles**).

Connecting to a Cluster

The Connect to Cluster tab, displayed within the Server Workbench Welcome overlay, allows you to connect to an existing RPI cluster on the current, or another, machine.



Note that, if connecting to a remote machine, the specified port must have opened there.

The following are shown:

- Username: this field is only shown when the Full access radio button is selected. It is mandatory when displayed. You must enter the username of an RPI cluster administrator to connect to Server Workbench. For more information on cluster administrators, please see the Users section.
- Password: this field is only shown when the Full access radio button is selected. It is mandatory when displayed. You must enter the password for the supplied Username.
- Interaction Server: this field is only shown when the Full access radio button is selected. It is mandatory when displayed. You must enter the Interaction service URI of the RPI server to which you wish to connect.
- Node Manager Server: this field is always displayed and is mandatory. You must enter the Node Manager service URI of the RPI server to which you wish to connect.
- Connect: having completed the required fields, click this button to connect to your required RPI cluster.

Upon connecting successfully in Full access mode, the Cluster tab, within the main Server Workbench interface, is displayed. All tabs are available.

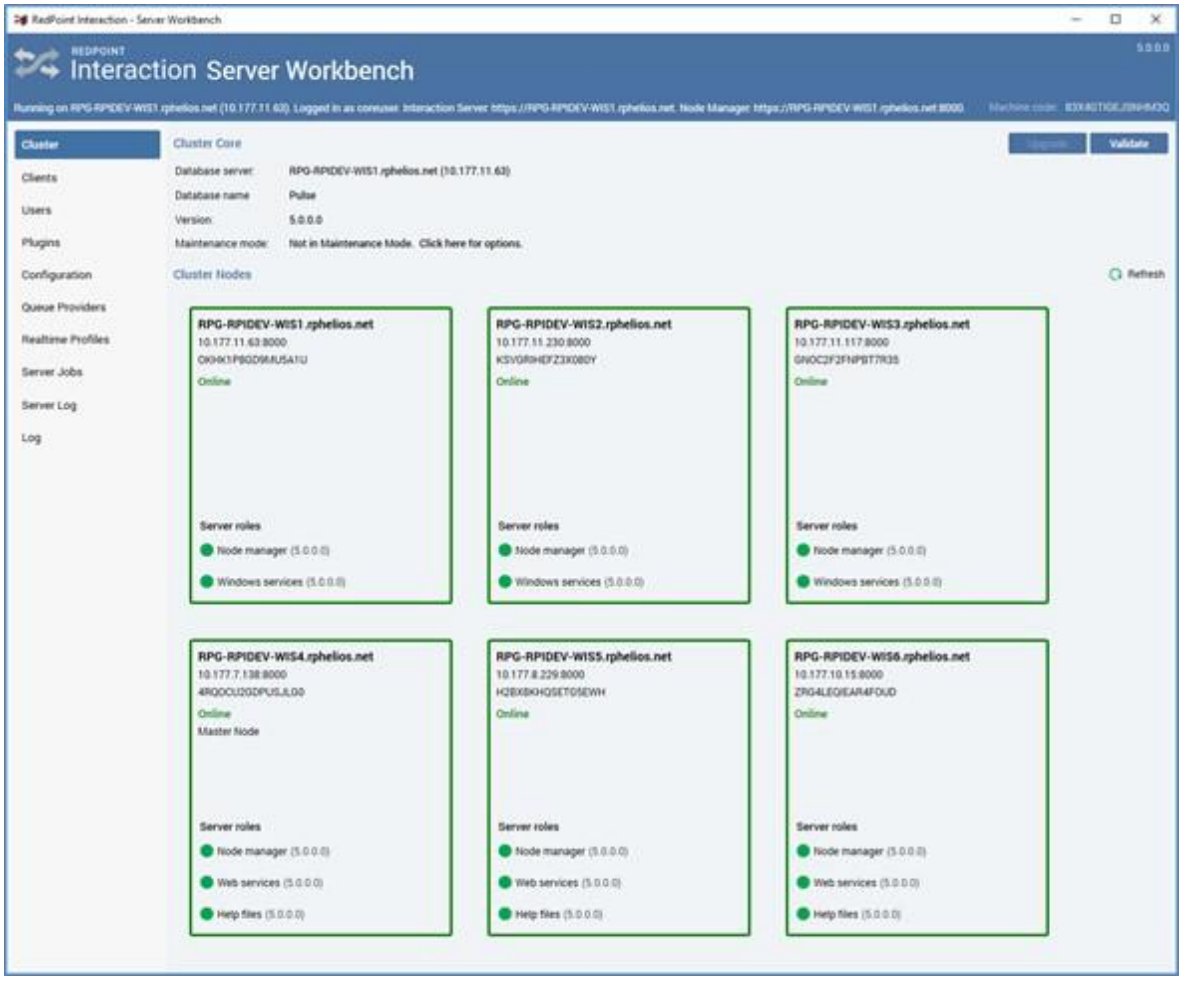
If connecting in Upgrade only mode, only the Cluster and Log tabs are shown. Please upgrade to the latest RPI version to access all available Server Workbench functionality.

If the details you entered are not recognized, or the Node manager service is not running on the machine to which you were trying to connect, an error dialog is shown.

A Failed to connect message is then shown at the Connect dialog.

Cluster Tab

The Cluster tab provides information relating to the current RPI cluster to which Server Workbench is connected. Details of its core and nodes are displayed:



The Cluster page of the Server Workbench. In this example it is shown connected to a cluster containing six nodes. After an initial Cluster installation, only one node will be visible.

Cluster Core

The section contains a toolbar and core properties.

Toolbar

Two buttons are exposed within the Cluster Core toolbar:

- Upgrade Core: this button is enabled when the version of the current Server Workbench instance is greater than the core to which it is connected. Clicking it displays the Upgrade Core overlay over the Cluster tab.
- Validate Core: clicking this button displays the Validate Core overlay over the Cluster tab.

Core Properties

The following read-only properties are shown:

- Database server: name and IP address
- Database name: set to “Pulse”
- Version

If the version of the currently-executing Server Workbench instances is higher than that of the core to which it is connected, an information label advises that “This Cluster Core can be upgraded”.

Cluster Nodes

This section contains a toolbar and cluster nodes list.

Toolbar

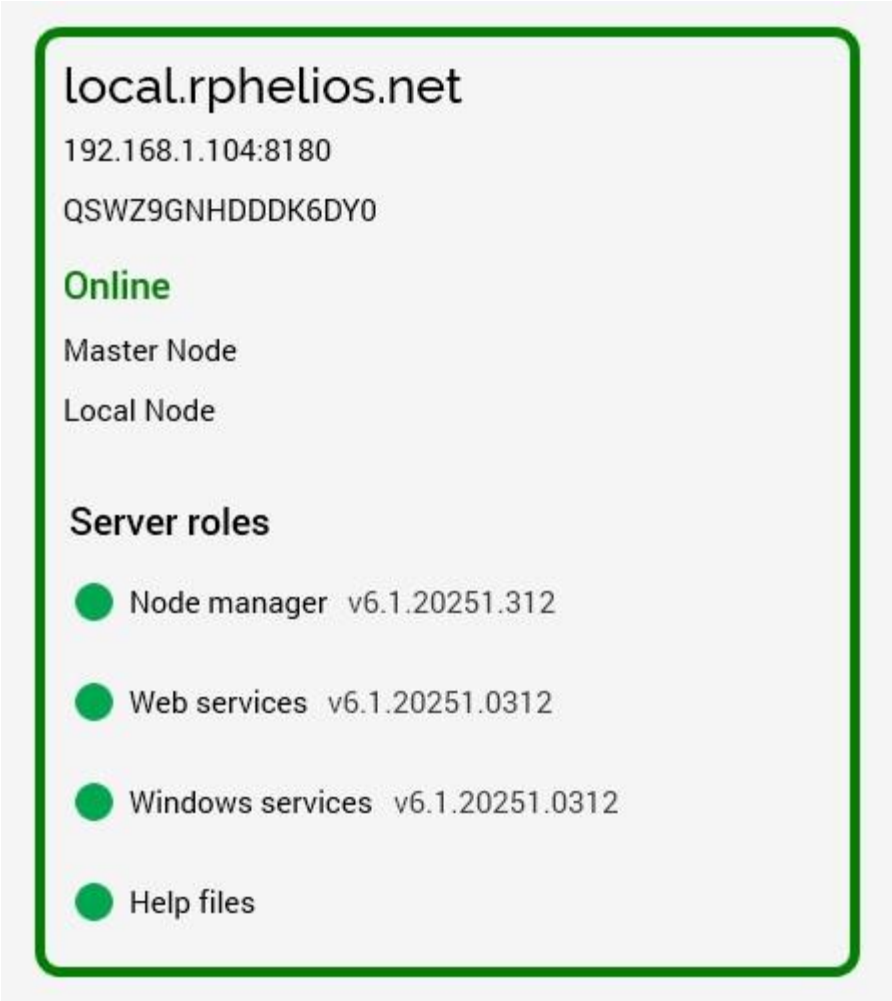
Two buttons are available at the Cluster Nodes toolbar:

- Refresh: clicking this button refreshes the cluster nodes list with the nodes’ most recent status information.
- Install & Add Local Node: this button is only displayed if:
    - Server Workbench is currently running on the same machine as the cluster core to which it is connected.
    - A node belonging to the cluster to which Server Workbench is connected is not currently installed on the machine upon which Server Workbench is running.

  Clicking this button allows you to add the local machine as a node within the cluster to which Server Workbench is connected. This is carried out in the Install Local Node overlay, which is displayed over the Cluster tab.

Nodes List

All nodes in the current cluster are listed. For each node, the following properties are displayed:



- Upgrade node message: if the versions of the role(s) installed at the node are lower than that of the core to which it belongs, a warning message (“Please upgrade this node”) is displayed at the top of the node.
- Server name: the name of the physical or virtual machine on which the node is installed
- Server IP address:port
- Machine code: the unique code of the physical or virtual machine on which the node is installed. Machine code is used when generating RPI server licenses. You can right-click the node’s machine code to copy its value to the clipboard.
- Online status: advice as to whether the node is online or offline (or in error) is presented via a halo and colored icon as well as being shown verbally.
- Master Node: displayed at the node within the cluster that is the master node. It is the master node’s responsibility to assign server jobs requested via the RPI client application to an online

node on which the Windows services role is installed. If the current master node is taken offline, another node in the cluster immediately assumes this responsibility.

- Master Node Role Disabled: displayed at the node when the master node role is disabled. Please see Disallow as Master Node, below, for further information.
- Local Node: displayed at the node representing the current machine on which Server Workbench is running.
- Server roles: nodes can support up to four roles, each of which defines the type of work that will be carried out by the node:
    - Node manager: all nodes support this role. ○ Web services: support for this role is optional. ○ Windows services: support for this role is optional. ○ Help files: support for this role is optional.

  For information on the tasks undertaken by each role, see **Roles**.

  The following are displayed at each server role:

    - Status icon. The role may be “Online”, “Offline” or “In error”. If “In error”, hovering over the role displays a tooltip containing the error details. ○ Role name ○ Version

  The following buttons are shown when hovering over a role:

    - Put this role offline: this button is available when the role is online. Clicking it puts the role offline. ○ Put this role online: this button is available when the role is offline. Clicking it puts the role online.

When hovering over a node, a context menu is displayed. The following buttons can appear in the menu:

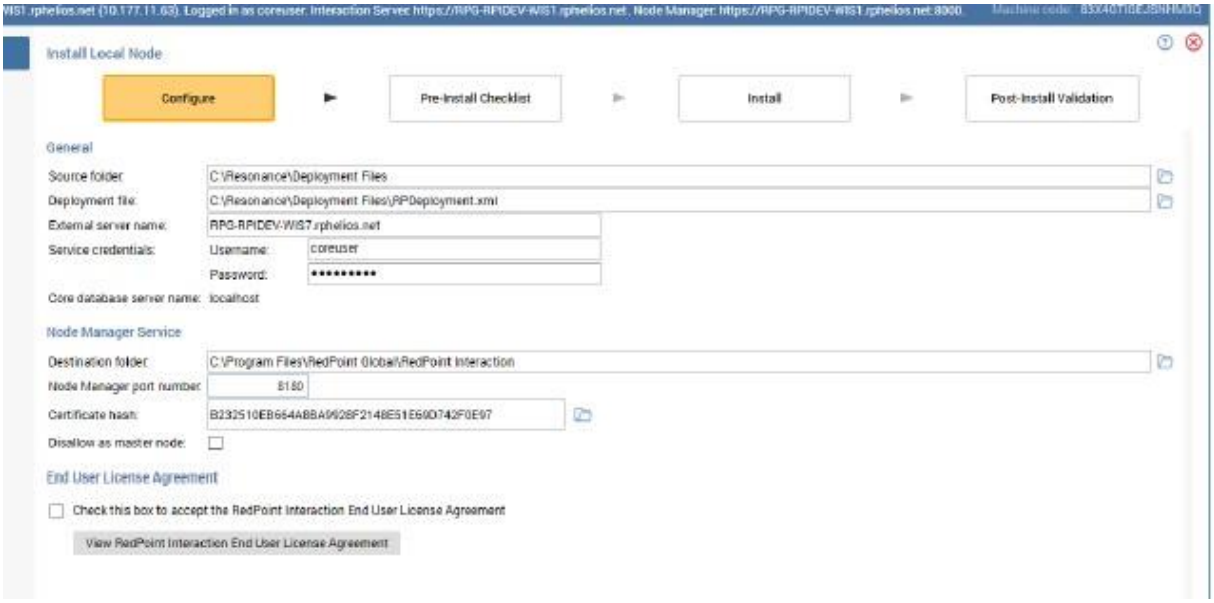
- Go offline: this button is available when the node is online. Clicking it takes the node and all associated roles offline.
- Go online: this button is available when the node is offline. Clicking it puts the node and all associated roles online.
- Install Roles: this button is available when one or more roles are not yet installed at the node. Clicking it displays the Install Roles overlay over the Cluster tab.
- Upgrade Roles: this button is available when the node is offline and the node role(s) versions are lower than the core version. Clicking it displays the Upgrade Roles overlay over the Cluster tab.
- Validate Roles: this button is always available. Clicking it displays the Validate Node overlay.
- Remove: this button is available when the node is offline. Clicking it removes the node from the cluster. Its invocation is protected by an “Are You Sure?” dialog. Note that you cannot remove a node if it is the only node in the cluster. Note also that, if you remove a node, you must manually uninstall its Node Manager service. You can do this using the following command:
  `sc delete [ServiceName]`
  Having done so, manually remove the ‘Program Files\RedPoint Global\RedPoint Interaction\Node Manager’ folder.

- Disable node: this option is available in a multi-node cluster. It allows you to disable a node within the cluster (other than the node to which you are currently connected). When a node is disabled, it assumes a Disabled state, and is displayed with a black border. You can invoke Enable at a Disabled node to re-enable it.
  When a node is disabled, no new jobs will be sent to it and existing work will be stopped so it can be picked up by another node. There may be short delay for the node to reactivate if the Node Manager service was started before the node was enabled as it polls the database looking for a status change.
- Disallow as Master Node: this option is available when the node is able to assume master node status. Invocation displays the text 'Master Role Node Disabled' at the node (note that it is necessary to manually refresh Server Workbench to remove the 'Master Node' indicator at the node, if displayed.
- Allow as Master Node: this option is available when the node's ability to assume master node status is disabled. Invocation enables the node to assume master node status ('Master Node' is displayed after a few seconds if the node assumes master node responsibility).

Installing and Adding a Local Node

If you want to add the current physical or virtual machine to a cluster, the core of which is installed on another machine on the same Windows domain, ensure that appropriate inbound ports are open on the core machine. You can start the process by clicking the Install & Add Local Node button displayed at the Cluster Nodes toolbar in the Cluster tab.

When you do so, the Install Local Node overlay is shown:



Install Local Node

Installation of node roles is carried out in a wizard style, using a series of sub-interfaces:

- Configure
- Pre-Install Checklist
- Install
- Post-Install Checklist

A graphical representation of current progress through the wizard is displayed within the top section.

Configure Sub-interface

The Configure sub-interface allows you define a series of properties required by Server Workbench in order to add the local machine to the cluster.

It consists of General and Node Manager Service sections.

General Section

This section allows you to define a basic set of properties relating to the action you are about to undertake:

- Source folder: the Windows file system folder in which the files used to perform the installation are to be found. You can specify this value manually or by browsing for a folder.
- Deployment file: an XML file that defines the tasks to be undertaken during installation. This property defaults to “RPDeployment.xml”, if that file can be found within the Source folder.
- External server name: of the local machine you are trying to connect to the cluster
- Service credentials: provide a domain username (with sufficient privileges), that will be used to run the initial node’s Node manager service. An optional Password can also be supplied.

- Core database server name: set to the name of the server on which the core operational databases are installed.

Node Manager Service

- Destination folder: this field defaults to “C:\Program Files\Redpoint Global\RPI”. It defines the folder in which the initial node’s Node manager service will be installed.
- Node manager port number: by default, port 8000 is used to access the Node manager service; however, you can change this value if required.
- Certificate hash: select the server certificate for this node to use.
- Disallow as master node: this checkbox is unchecked by default. When unchecked, the current node can assume master node status. When checked, the current node cannot assume master node status, unless explicitly allowed through use of the context menu option shown when right clicking the node.

Validation

If you have not correctly completed the mandatory fields within the Configure sub-interface, a validation errors indicator is displayed to the bottom right. Clicking the indicator displays details of the errors in a dialog. You cannot progress to the Pre-Install checklist when outstanding validation errors remain.

If the configuration is valid, an indicator advises accordingly. You may now move to the next wizard step.

Navigation

You can move forward through the Install Local Node wizard by clicking on the Next button at the bottom right. Doing so displays the Pre-Install Checklist sub-interface.

Pre-Install Checklist Sub-interface

The purpose of this interface is to allow a series of checks to be performed to ensure that all pre-requisites have been met prior to proceeding with the install.

Toolbar

A toolbar, shown above the Pre-Upgrade Checklist section, exposes a single button:

- Restart Checklist: clicking this button performs the pre-installation validation checks again. In the event of a failed check, you can return to the Configure sub-interface, rectify any configuration issues therein and, on return to the Pre-Install Checklist, re-run the pre-upgrade checks.

Pre-Installation Checklist

Details of status information provided at each installation step, and overall progress, are provided in the Core Pre-Install Checklist documentation.

The following checks are performed:

- Check for existing RPI installation
- Check the service credentials are valid
- Check access to source folder
- Check if .NET 4.8 is installed
- Check if the Node Manager port is available
- Check if any required software dependencies are available
- Validate the application license file
- Configure the Node Manager port with the SSL certificate

Navigation

On completion of the checklist, you can continue your progress through the Install Local Node wizard by clicking on the Next button at the bottom right. Access to the Install sub-interface is controlled by an “Are You Sure?” dialog.

If you attempt to access the next interface prior to completion of the checklist, you will be advised to wait until all steps are complete.

You can return to the Configure sub-interface using the Back button.

Install Sub-interface

This interface is used to provide visibility of the tasks executed during installation of the local node.

Toolbar

A single button is shown in the toolbar:

- Restart Installation: this button is available when the installation's status is one of "Completed with Warnings" or "Has a Dependent Job with Errors". It allows you to repeat the Install Local Node steps from the beginning.

Installation Steps

The steps required to perform the install are undertaken in sequence.

Navigation

On completion of the installation, you can continue your progress through the Install Local Node wizard by clicking on the Next button at the bottom right.

If you attempt to access the next interface prior to completion of the installation, you will be advised to wait until all steps are complete. Note that you can return to the Pre-Install Checklist sub-interface using the Back button.

Post-Install Validation Sub-interface

A series of checks are performed following installation of a local node, to verify that the exercise was completed successfully.

Toolbar

Clicking the Restart Validation button allows to you repeat the post-install validation checklist

Post-Install Checklist

A series of steps are undertaken to validate that the install was performed correctly.

Navigation

On completion of the Post-Install checklist, the Next button, displayed at the bottom right, is replaced by a Finish button. Clicking Finish closes the Install Local Node overlay and displays the Server Workbench tab set, which includes all tabs. The Cluster tab is shown by default, and the cluster's nodes, including the new local node, are displayed therein. You can now install additional roles as required at the new node.

If you click Finish prior to completion of the checklist, you will be advised to wait until all steps are complete.

You can return to the Install sub-interface using the Back button.

Upgrade

The RPI server is upgraded using the Server Workbench application. Before running Server Workbench, please ensure that the account as which you have logged in has full access to the deployment share folder that was created when the core was installed.

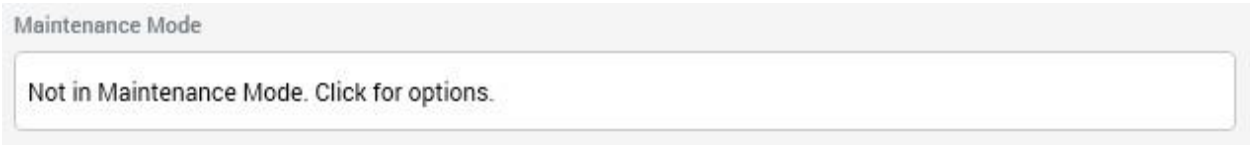
Note that, whenever an RPI upgrade is performed, all relevant components (e.g. Realtime, Callback service, etc.) should be upgraded at the same time.

Maintenance Mode

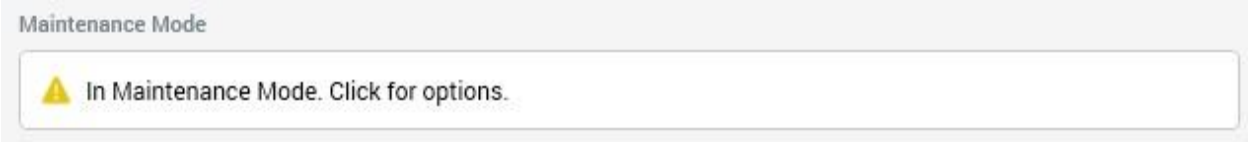
It is possible to put the cluster into 'maintenance mode' in advance of performing scheduled maintenance activities – typically applying a new version upgrade. When in maintenance mode, users are unable to log into the system. When the cluster enters maintenance mode, any logged-in users are logged out of the system automatically.

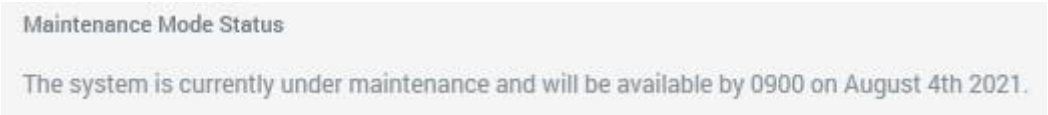
A button at the Cluster Core section of the Cluster tab facilitates display of the Maintenance Mode dialog, which allows you to put the cluster in and take the cluster out of maintenance mode. The verbiage shown at the button depends on whether the cluster is currently in maintenance mode.

If the cluster is not in maintenance mode, the following information message is shown:



If the cluster is in maintenance mode, the following are shown:



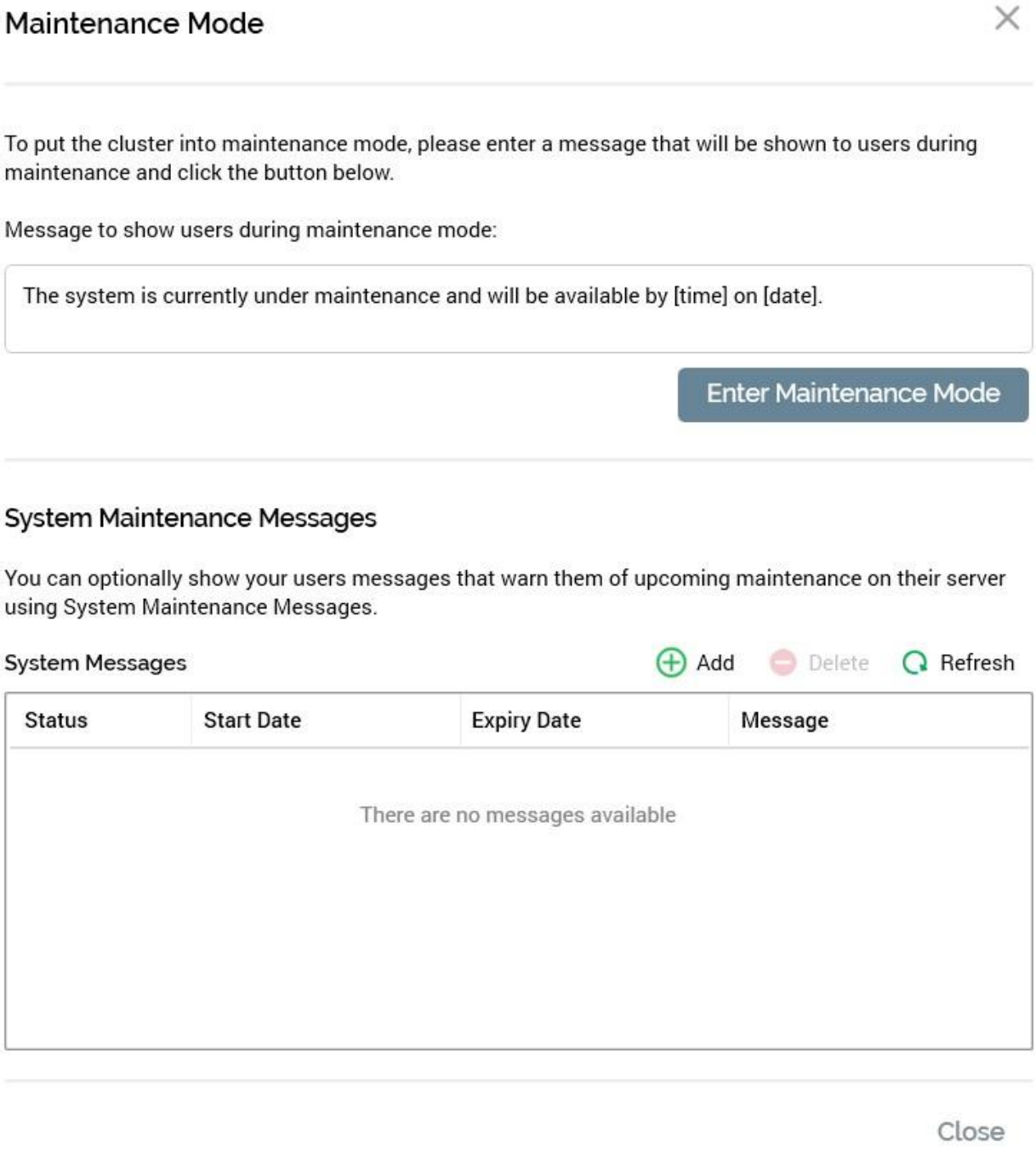


Note display of the maintenance mode message in this case.

Clicking the button in either case displays the Maintenance Mode dialog.

Maintenance Mode Dialog

The Maintenance Mode dialog allows you to put the cluster into, and take the cluster out of, maintenance mode.



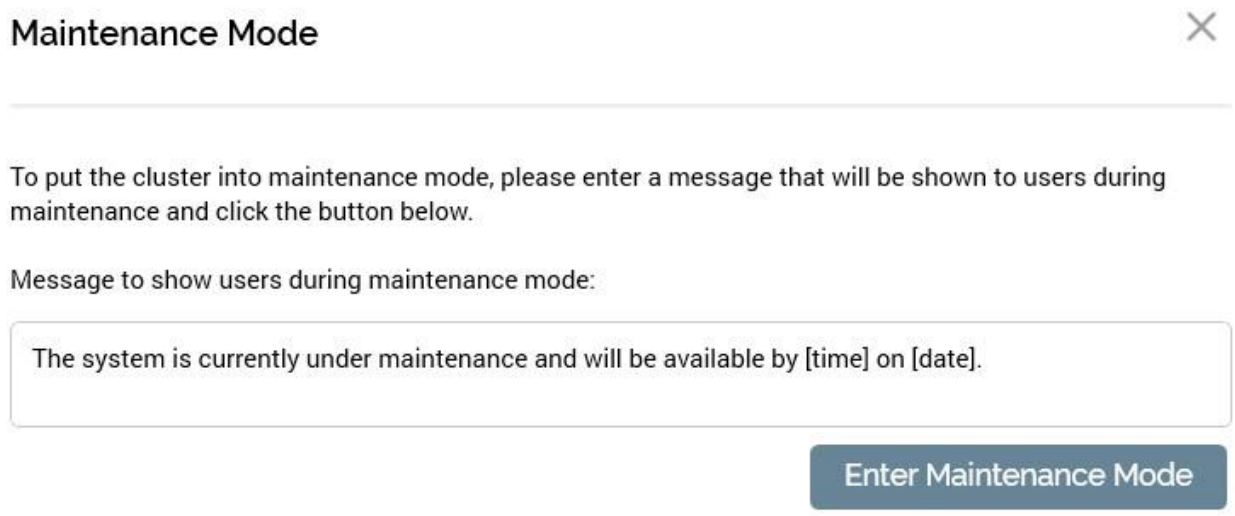
It also allows you to define a message to be shown to users who attempt to log in when the cluster is in maintenance mode, and additional system maintenance messages that primarily provide warning of scheduled maintenance.

A Close button is shown at the top right of the dialog. Clicking it removes the Maintenance Mode dialog from display.

The dialog consists of Maintenance Mode and System Maintenance Messages sections, both described in more detail below.

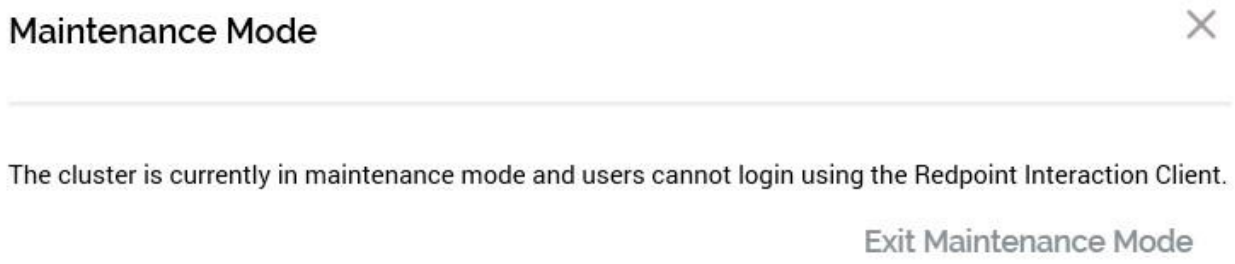
Maintenance Mode Section

When not in maintenance mode the top section of the Maintenance Mode dialog contains the following:



- Maintenance mode message: you can enter a message that will be shown to users who attempt to log in when the cluster is in maintenance mode. The text field contains a default message. A maintenance mode message must be supplied.
- Enter Maintenance Mode: this button allows you to put the cluster into maintenance mode. When you click it, an 'Are You Sure?' dialog is shown. If you elect to proceed, the cluster is put into maintenance mode. The dialog is removed from display, and the Cluster tab is refreshed. Any users who are logged in at the beginning of maintenance mode are logged out automatically.

When in maintenance mode, the following is displayed:



- Maintenance mode message: note that this property is not displayed when in maintenance mode.

- Exit Maintenance Mode: this button allows you to take the cluster out of maintenance mode. When you click it, an 'Are You Sure?' dialog is shown. If you elect to proceed, the cluster is taken out of maintenance mode. The dialog is removed from display, and the Cluster tab is refreshed. Users may once again log into the cluster using the RPI client.

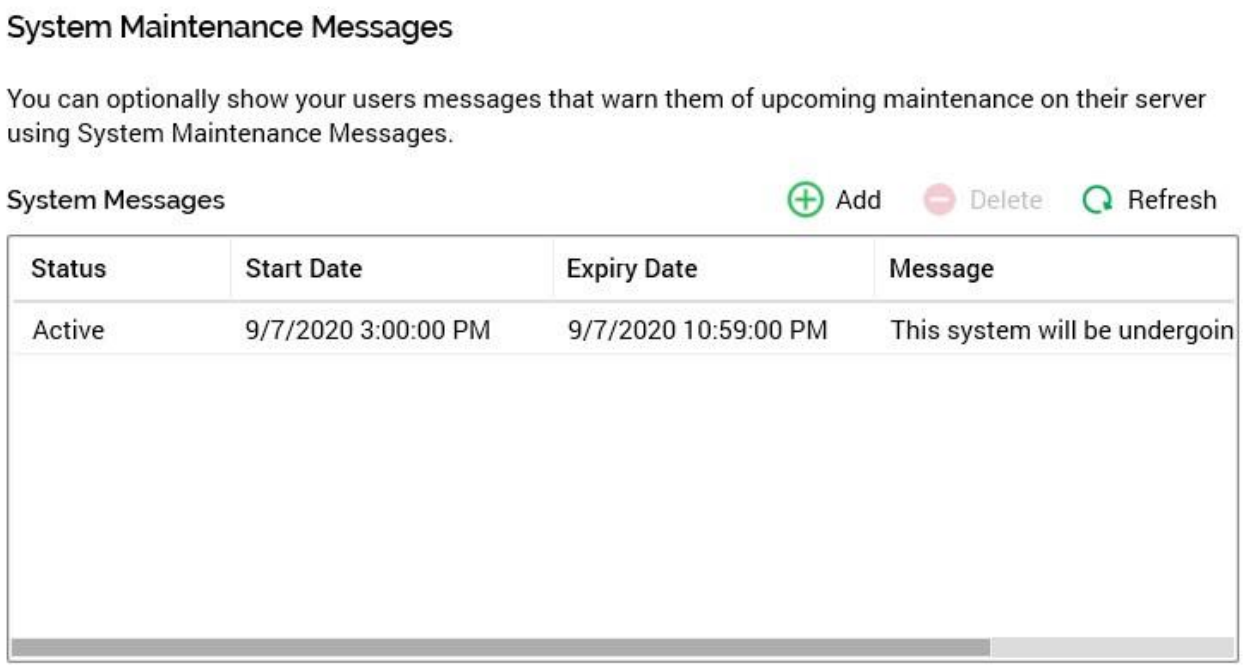
System Maintenance Messages Section

System maintenance messages can be used to warn users in advance of impending system downtime. They can also be used to convey other salient information to users of the application.

Active system maintenance messages are displayed as pulse notifications when a user logs in. If more than one system maintenance message has been configured, each is shown as a separate pulse notification.

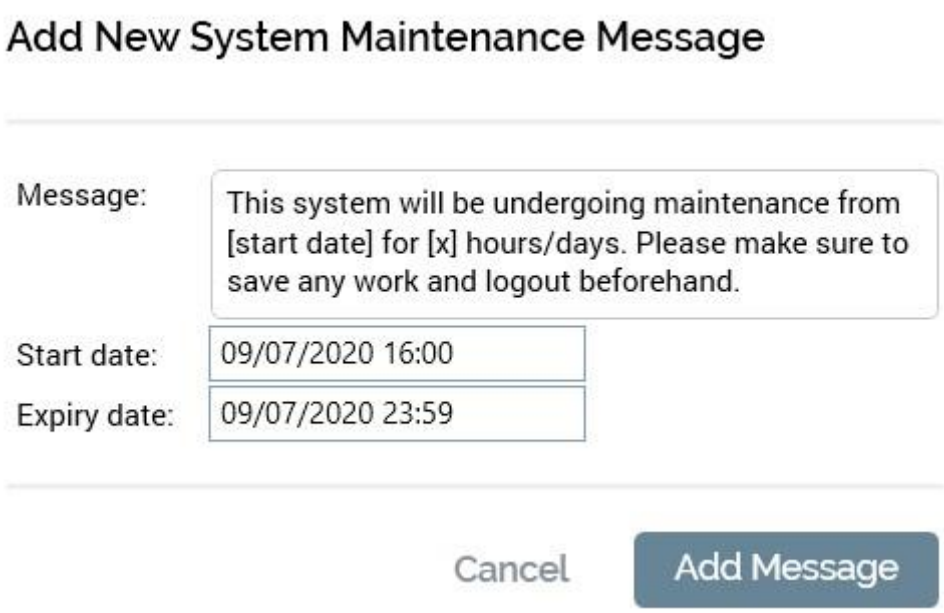
If an active system maintenance message is added when a user is logged in, a pulse notification is similarly shown. This also occurs when a pending system maintenance message becomes active.

The bottom section of the Maintenance Mode dialog contains the following:



The System Messages section contains a toolbar and a grid.

- Toolbar: the following buttons are available at the System Messages toolbar: ○ Add: clicking this button displays the Add New System Maintenance Message dialog, which contains the following:



  - Message: a mandatory field that defaults to 'This system will be undergoing maintenance from [start date] for [x] hours/days. Please make sure to save any work and logout beforehand'.
  - Start date: defaults to today, and the most recent hour. Its provision is mandatory.
  - Expiry date: defaults to today, 23:59. Its provision is mandatory. Expiry date must be after Start date and must be greater than the current date and time.
  - Add Message: clicking this button close the dialog and adds the message to the grid. An advisory message is shown at the top of the Server Workbench interface.
  - Cancel: clicking this button removes the dialog from display.

  - Delete: invocation of this option is protected by and 'Are You Sure?' dialog. Clicking the button deletes the message from the grid. An advisory message is shown at the top of the Server Workbench interface. ○ Refresh: refreshed the list of system maintenance messages.

- Grid: when empty, a message ('There are no messages available') is shown. The System Messages grid is read-only, and contains the following columns:
  - Status: one of Pending, Active or Expired.
  - Start Date ○ Expiry Date ○ Message

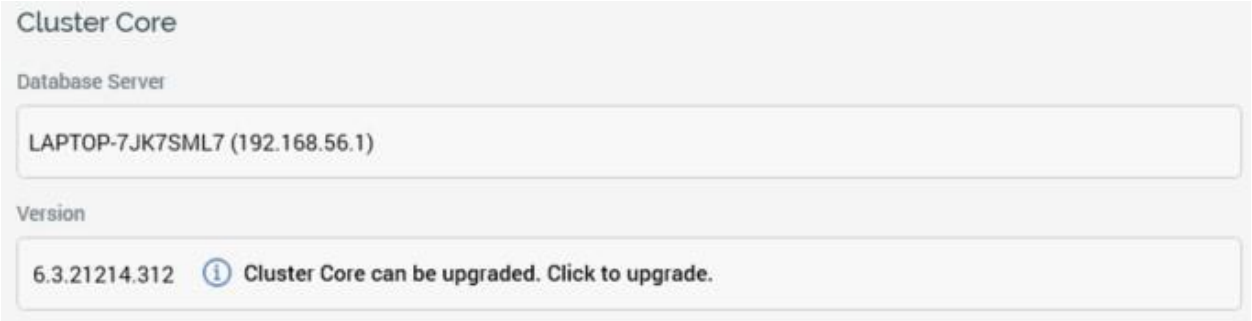
Messages are ordered in descending order by Start Date.

Performing an Upgrade

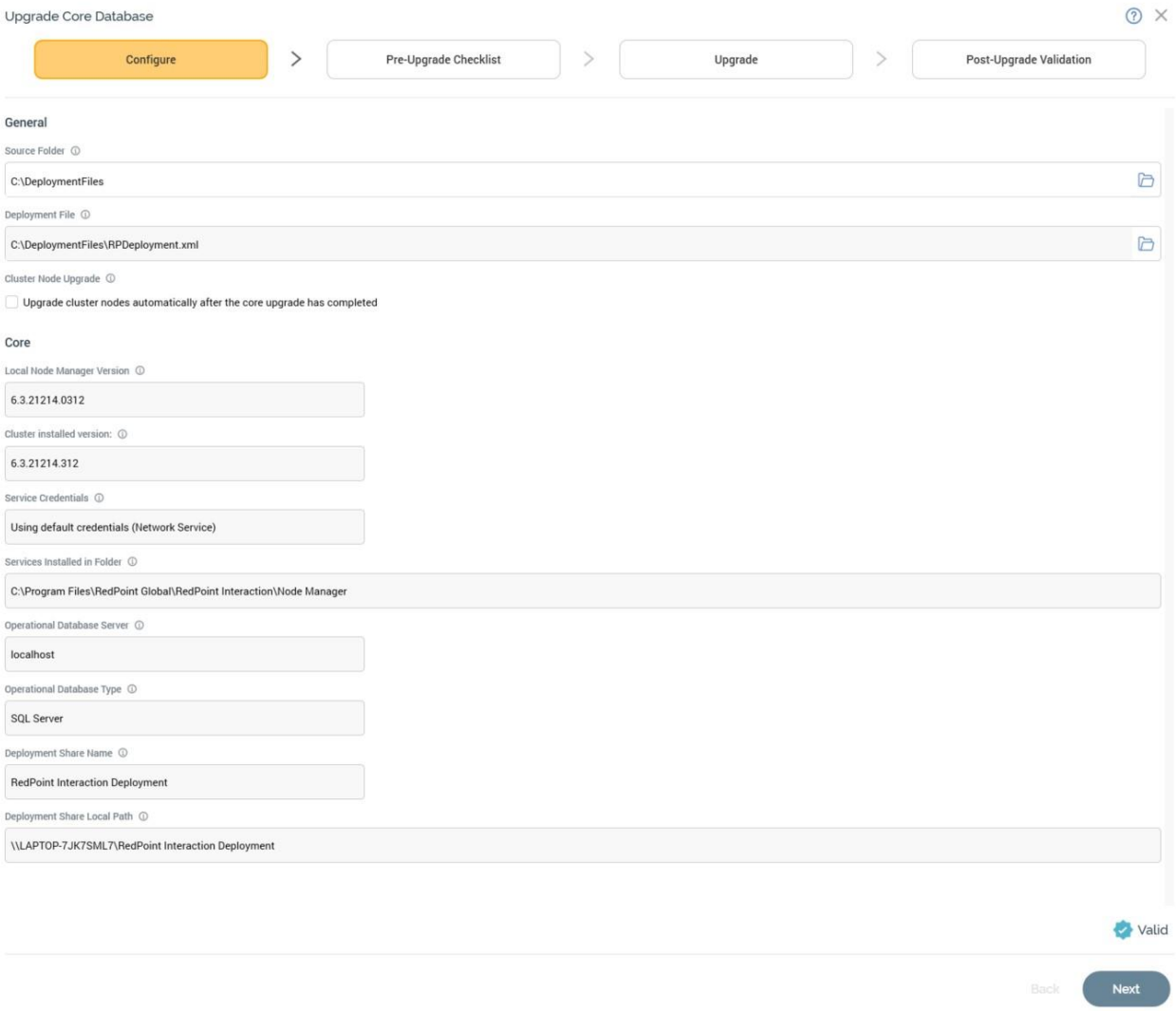
Upgrading the Core

The process of upgrading an RPI server consists of two steps: upgrading the core and upgrading each node within the core. This section describes the steps undertaken in performing the former task.

When the version of the Server Workbench you are currently running is greater than the version of the core to which you have connected, an indicator is provided advising that you are now at liberty to upgrade the core:



In addition, the Upgrade Core button, shown within the toolbar to the top right of the Cluster tab, is enabled. You can start the process of upgrading the core by clicking this button. Doing so displays the Upgrade Core Database overlay:



Note that you can only upgrade to the latest version of RPI when running RPI v6.0 or greater. If you attempt to upgrade from an older version, a message is displayed advising you to upgrade to v6.0 first.

Upgrade Core Database

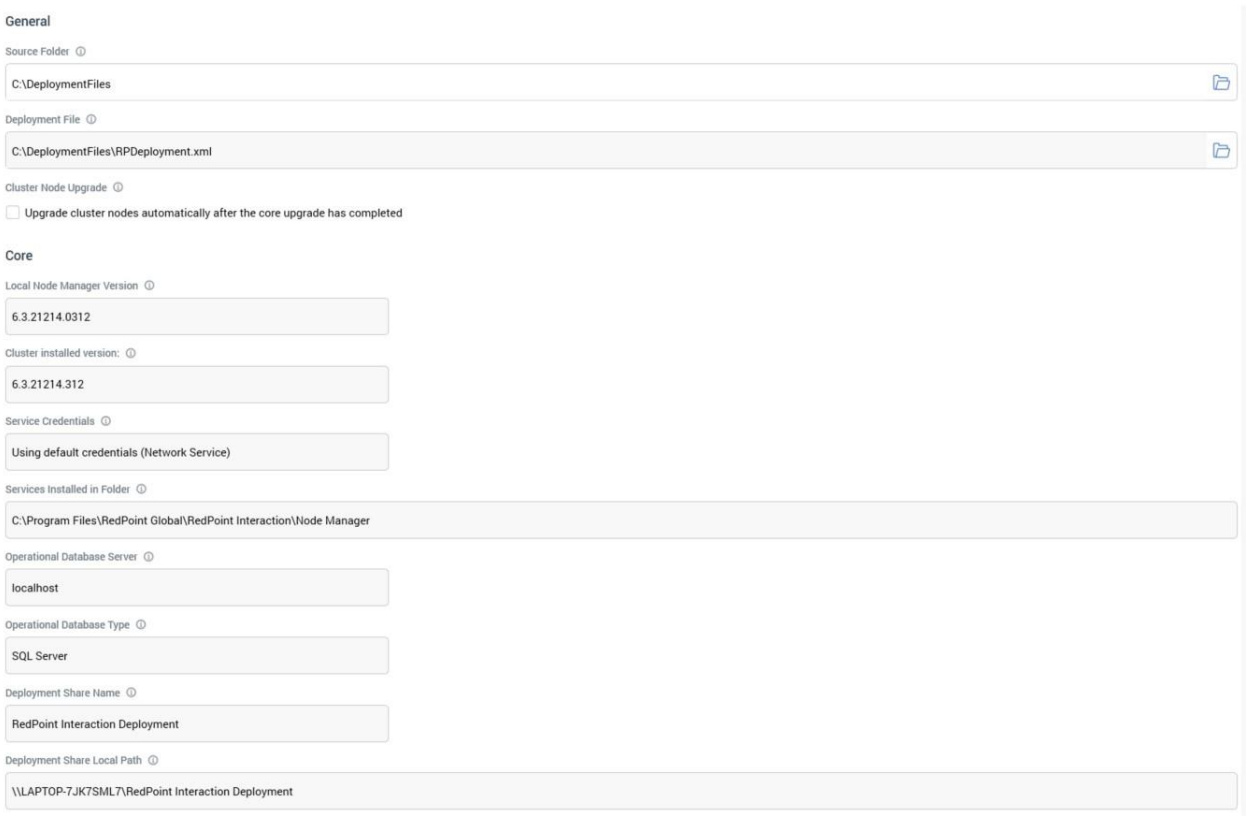
Upgrade of the core is carried out in a wizard style, using a series of sub-interfaces as follows:

- Configure
- Pre-Upgrade Checklist
- Upgrade
- Post-Upgrade Validation

A graphical representation of current progress through the wizard is displayed within the top section:

Configure Sub-interface

The Configure sub-interface allows you define a series of properties required by Server Workbench in advance of upgrading the core.



It contains General and Core sections.

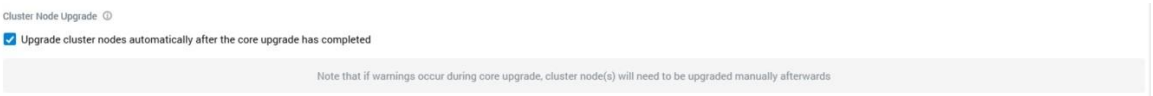
General

This section allows you to define a basic set of properties relating to the installation:

- Source Folder: the Windows file system folder in which the files used to perform the upgrade are to be found. You can specify this value manually or by browsing for a folder.
- Deployment File: an XML file that defines the tasks to be undertaken during upgrade. This property defaults to “RPDeployment.xml”, if that file can be found within the Source folder.
- Cluster Node Upgrade: containing an Upgrade cluster nodes automatically after the core upgrade has completed checkbox, which is unchecked by default. If left unchecked, once the core upgrade is complete, you will need to upgrade roles at all roles in the cluster manually.

Please note: if you choose to upgrade roles manually having first upgraded the core, please do not close Server Workbench post-core upgrade. If you do so, you will not be able to reconnect, and will need to complete the upgrade manually.

If checked, Server Workbench upgrades the roles at all cluster nodes automatically. The property is accompanied by a note:



Core

This section displays a series of read-only properties of the current cluster:

- Local Node Manager version
- Cluster Installed Version
- Service Credentials
- Services Installed in Folder
- Operational Database Server
- Operational Database Type
- Deployment Share Name
- Deployment Share Local Path

Validation

If you have not correctly completed the mandatory fields within the Configure sub-interface, a validation errors indicator is displayed to the bottom right:

Clicking the indicator displays details of the errors in a dialog.

You cannot progress to the Pre-Upgrade checklist when outstanding validation errors remain.

If the configuration is valid, the following indicator is shown:

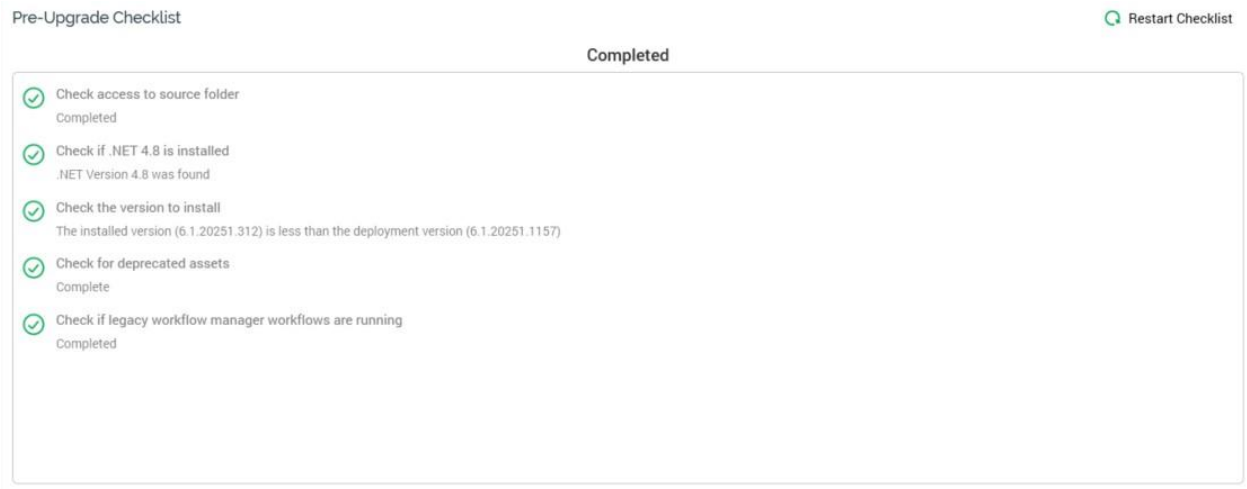
You may now move to the next wizard step.

Navigation

You can move forward through the Upgrade Core wizard by clicking on the Next button at the bottom right. Doing so displays the Pre-Upgrade Checklist sub-interface.

Pre-Upgrade Checklist Sub-interface

The purpose of this interface is to allow a series of checks to be performed to ensure that all pre-requisites have been met prior to proceeding with the upgrade.



Toolbar

A toolbar, shown above the Pre-Upgrade Checklist section, exposes a single button:

- Restart Checklist: clicking this button performs the pre-upgrade validation checks again. In the event of a failed check, you can return to the Configure sub-interface, rectify any configuration issues therein and, on return to the Pre-Upgrade Checklist, re-run the preupgrade checks.

Pre-Upgrade Checklist

A series of pre-upgrade checks are carried out to ensure that all necessary pre-requisites are in place for the upgrade to occur.

Navigation

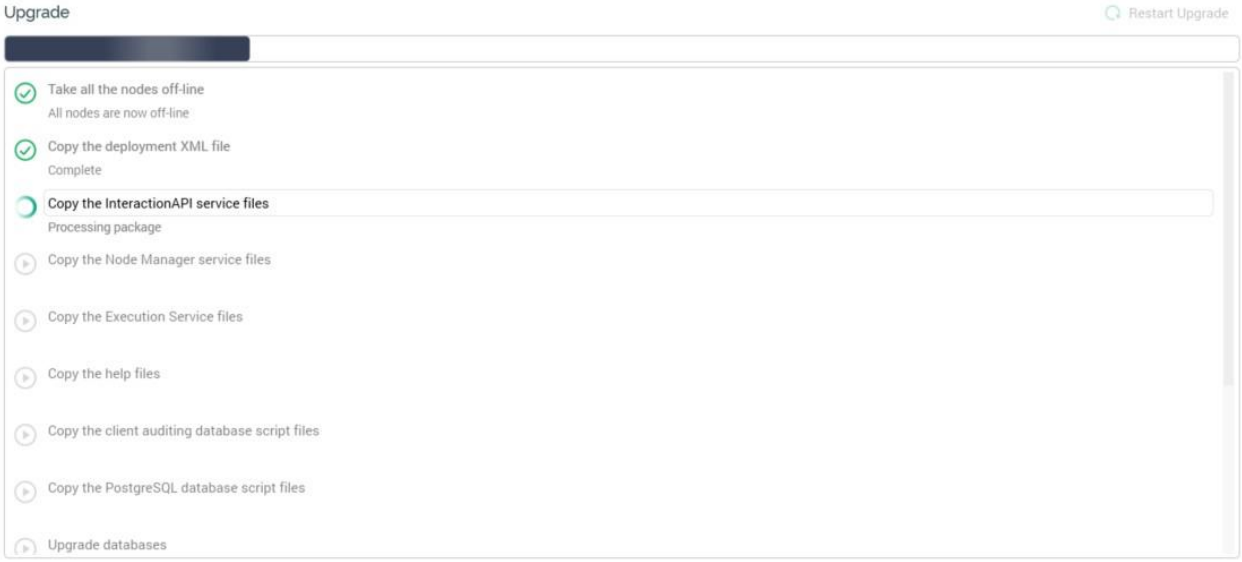
On completion of the checklist, you can continue your progress through the Upgrade Core wizard by clicking on the Next button at the bottom right. Access to the Upgrade sub-interface is controlled by an “Are You Sure?” dialog.

If you attempt to access the next interface prior to completion of the checklist, you will be advised to wait until all steps are complete.

You can return to the Configure sub-interface using the Back button.

Upgrade Sub-interface

This interface is used to provide visibility of the tasks executed during Core upgrade.



Toolbar

A single button is shown in the toolbar:

- Restart Upgrade: this button is available when the upgrade's status is one of Completed with Warnings or Has a Dependent Job with Errors. It allows you to repeat the core upgrade steps from the beginning.

Upgrade Steps

A series of steps to upgrade the RPI core services and databases are undertaken.

Navigation

On completion of the upgrade, you can continue your progress through the Upgrade Core wizard by clicking on the Next button at the bottom right.

If you attempt to access the next interface prior to completion of the installation, you will be advised to wait until all steps are complete.

You can return to the Pre-Upgrade Checklist sub-interface using the Back button.

Post-Upgrade Validation Sub-interface

A series of checks are performed following upgrade of the core to verify that the exercise was completed successfully.

Toolbar

Clicking the Restart Validation button allows to you repeat the post-upgrade validation checklist.

Post-Upgrade Checklist

A series of checks are performed to make sure the upgrade was successful.

Navigation

On completion of the Post-Upgrade checklist, the Next button, displayed at the bottom right, is replaced by a Finish button.

If you elected to upgrade node roles manually, clicking Finish closes the Upgrade Core overlay and displays the Server Workbench tab set, which includes all tabs. The Cluster tab is shown by default, and the cluster's nodes are displayed. All nodes are offline; you must now upgrade each one in turn.

If you elected to upgrade node roles automatically, on clicking Finish a dialog confirms that you wish to proceed with the node upgrades. The roles upgrade usually takes a few minutes, after which you can reconnect to RPI using Server Workbench.

If you click "Finish" prior to completion of the checklist, you will be advised to wait until all steps are complete.

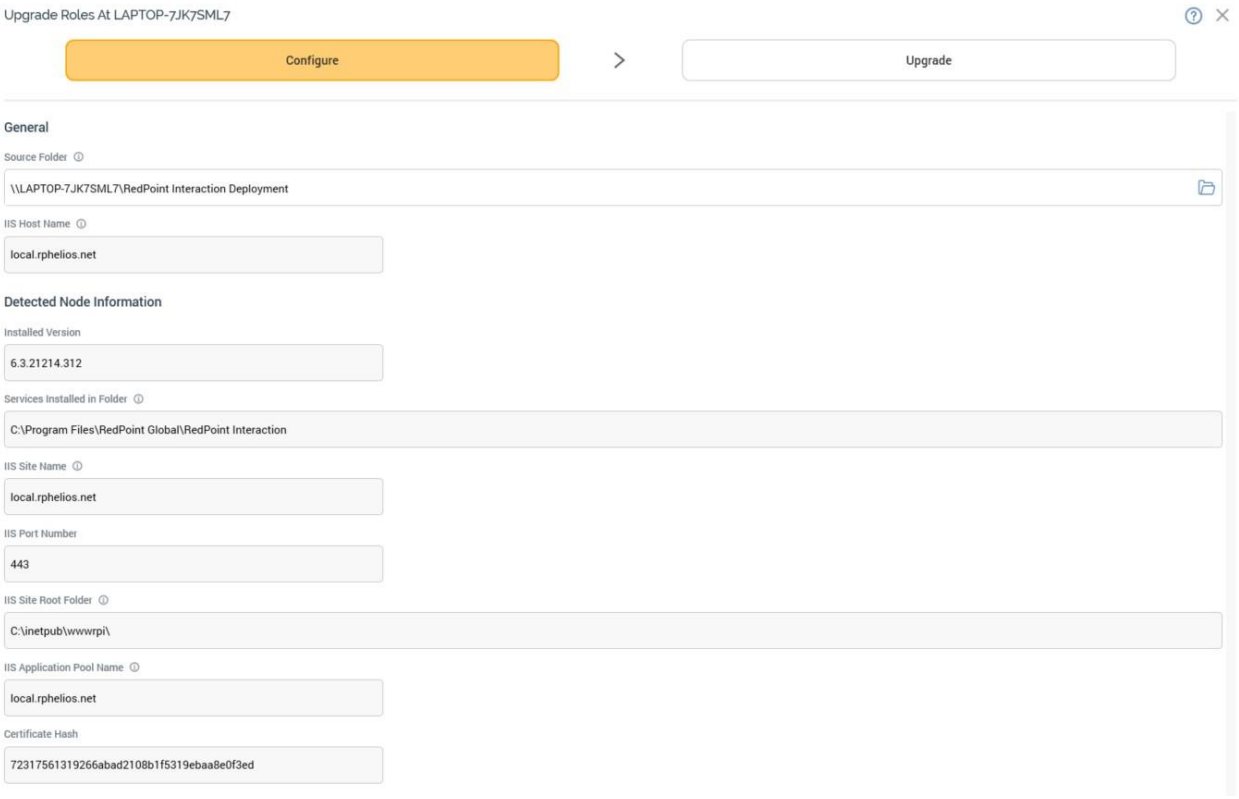
You can return to the Upgrade sub-interface using the Back button.

Upgrading Roles

You must upgrade roles at a node in an RPI cluster when the versions thereof are lower than the version number of the core in which they are located.

You can start the process of upgrading node roles by clicking the Upgrade Roles context menu button, displayed when you hover over the node.

When you do so, the Upgrade Roles overlay is shown:



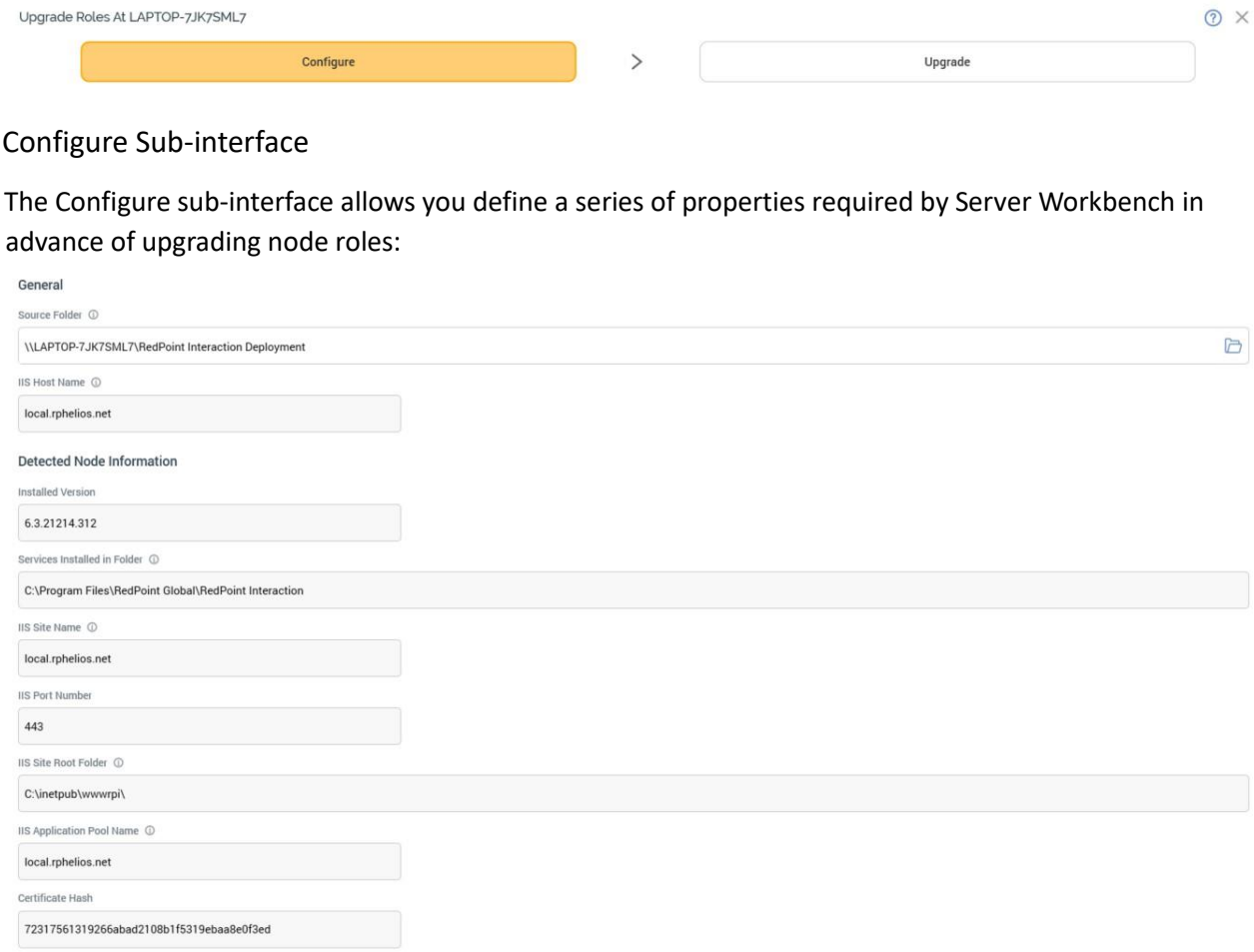
The overlay contains the following:

Header

Installation of node roles is carried out in a wizard style, using two sub-interfaces:

- Configure
- Upgrade

A graphical representation of current progress through the wizard is displayed within the header:



Configure Sub-interface

The Configure sub-interface allows you define a series of properties required by Server Workbench in advance of upgrading node roles:

It consists of General and Detected Node Information sections.

General Section

This section displays a series of non-role specific properties required prior to commencing role installation:

- Source Folder: defaults to the share created at core install, where the current set of deployment files are to be found.
- IIS Host Name: this needs to be set to the external server name of the current machine, or, if web services are running using IIS load balancing, as the public-facing load balancer server.

Detected Node Information

The read-only fields within this section are determined automatically by Server Workbench and displayed for informational purposes:

- Installed Version

This property is only shown if the Windows services role is installed:

- Services Installed in Folder

These properties are only displayed if the Web services and/or Help files role(s) is/are installed:

- IIS Site Name
- IIS Port Number
- IIS Site Root Folder
- IIS Application Pool Name
- Certificate Hash

Validation

If you have not correctly completed the mandatory fields within the Configure sub-interface, a validation errors indicator is displayed to the bottom right. Clicking the indicator displays details of the errors in a dialog. You cannot progress to the Upgrade interface when outstanding validation errors remain.

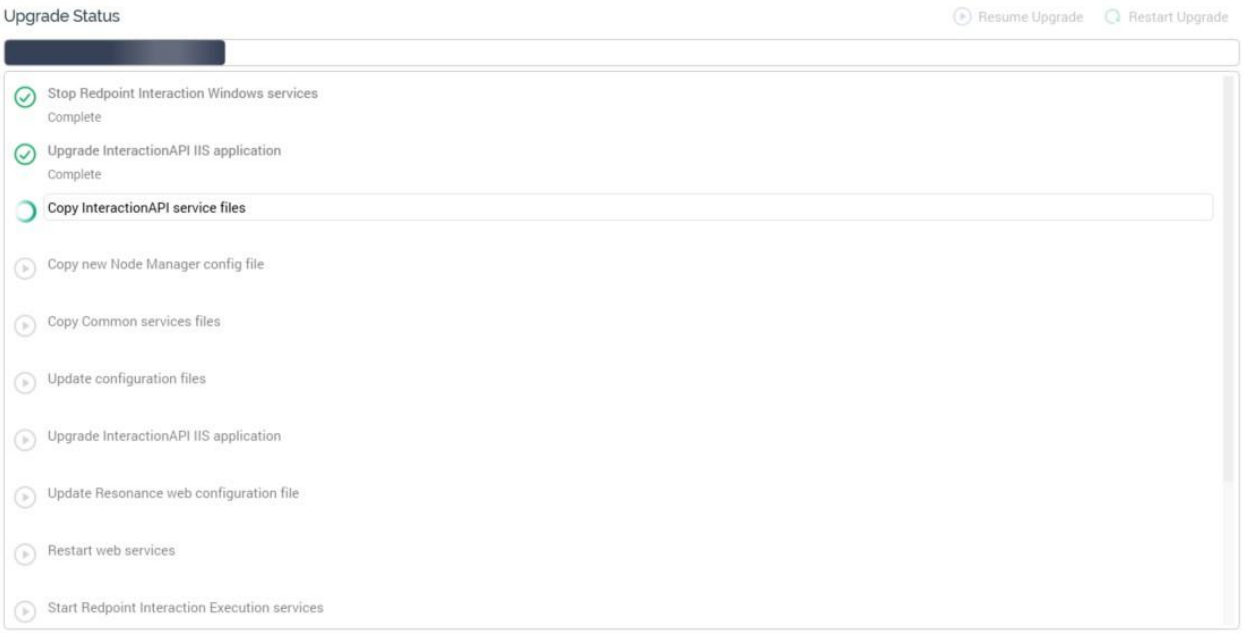
If the configuration is valid, an indicator advises accordingly. You may now move to the next wizard step.

Navigation

You can move forwards through the Upgrade Roles wizard by clicking on the Next button at the bottom right. Doing so displays the Upgrade sub-interface.

Upgrade Sub-interface

This interface is used to provide visibility of the tasks executed during the upgrade of node roles.



Toolbar

Two buttons are shown at the toolbar:

- Resume Upgrade: this button is available when the upgrade's status is Has Been Paused on Warnings. It allows you to carry on the upgrade at the point where the warning occurred.
- Restart Upgrade: this button is available when the status is one of Completed with Warnings or Has a Dependent Job with Errors. It allows you to repeat the upgrade steps from the beginning.

Upgrade Steps

A series of steps to upgrade node roles are performed.

Navigation

On completing the upgrade, the Next button, displayed at the bottom right, is replaced by a Finish button. Clicking Finish closes the Upgrade Roles overlay and displays the Server Workbench tab set, which includes all tabs. The node at which the role(s) was/were upgraded is shown as Online.

If you click Finish prior to completing the upgrade, you will be advised to wait until all steps are complete.

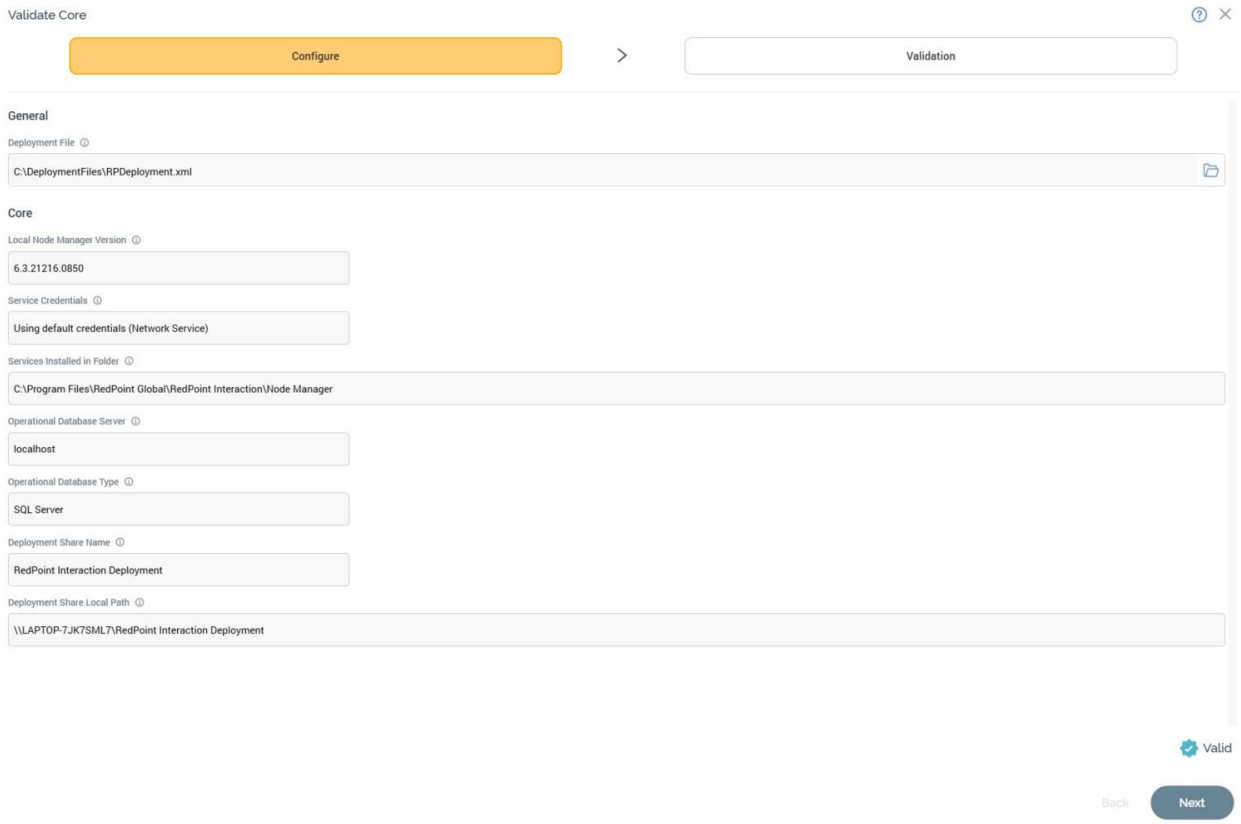
You can return to the Configure sub-interface using the Back button.

Validation

Single Stack

Validating the Core

You may wish to check that everything is installed and running correctly in a given RPI core. You can do so by clicking the Validate Core button, shown within the toolbar to the top right of the Cluster tab. Doing so displays the Validate Core overlay:



Validate Core

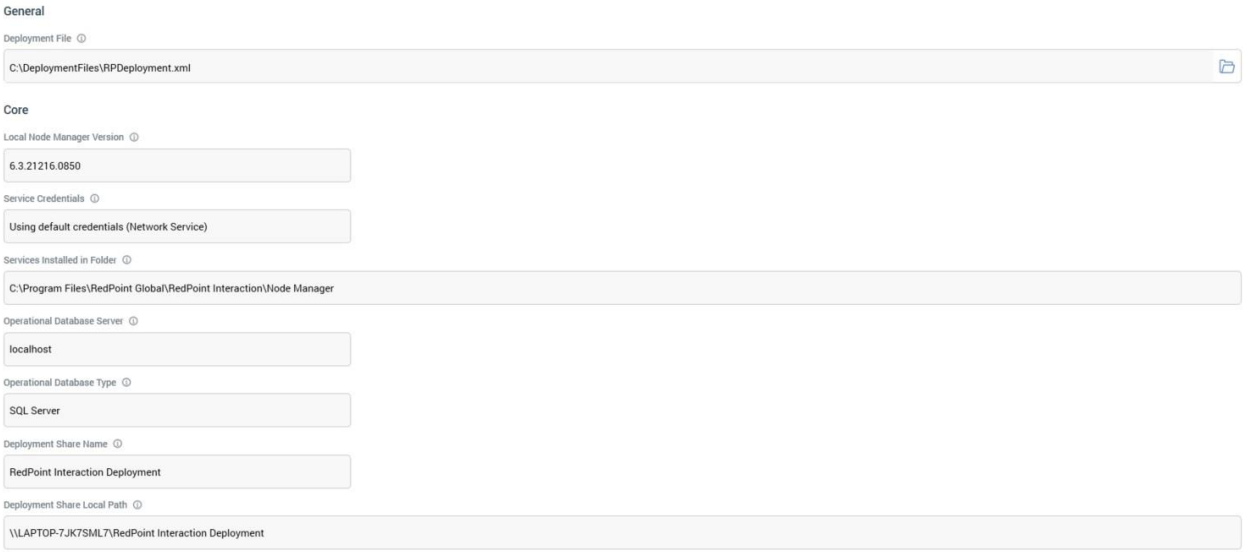
Validation of the core is carried out in a wizard style, using two sub-interfaces as follows:

- Configure
- Validation

A graphical representation of current progress through the wizard is displayed within the top section:

Configure Sub-interface

The Configure sub-interface allows you define a single property required by Server Workbench in advance of upgrading the core.



General Section

This section allows you to define a single property:

- Deployment File: an XML file that defines the validation steps to be performed. It should be set to “RPDeployment.xml”, which can be found within the Source folder.

Core Section

This section displays a series of read-only properties of the current cluster:

- Local Node Manager version
- Service Credentials
- Services Installed in Folder
- Operational Database Server
- Operational Database Type
- Deployment Share Name
- Deployment Share Local Path

Validation

If you have not correctly completed the mandatory fields within the Configure sub-interface, a validation errors indicator is displayed to the bottom right. Clicking the indicator displays details of the errors in a dialog. You cannot progress to the Validation sub-interface when outstanding validation errors remain.

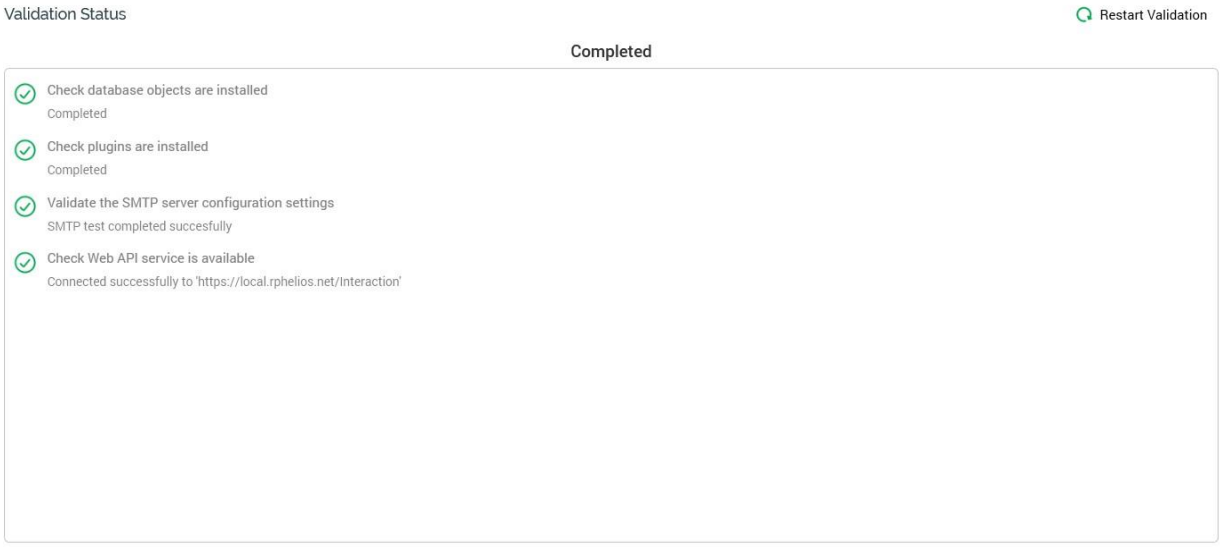
If the configuration is valid, an indicator advises accordingly. You may now move to the next wizard step.

Navigation

You can move forward through the Validate Core wizard by clicking on the Next button at the bottom right. Doing so displays the Validation sub-interface.

Validation Sub-interface

The purpose of this interface is to display the results of a series of checks performed to ensure that the current RPI cluster is valid.



Toolbar

Clicking the Restart Validation button performs the validation checks again.

Validation Status

A series of checks are performed to ensure the current cluster is valid.

Navigation

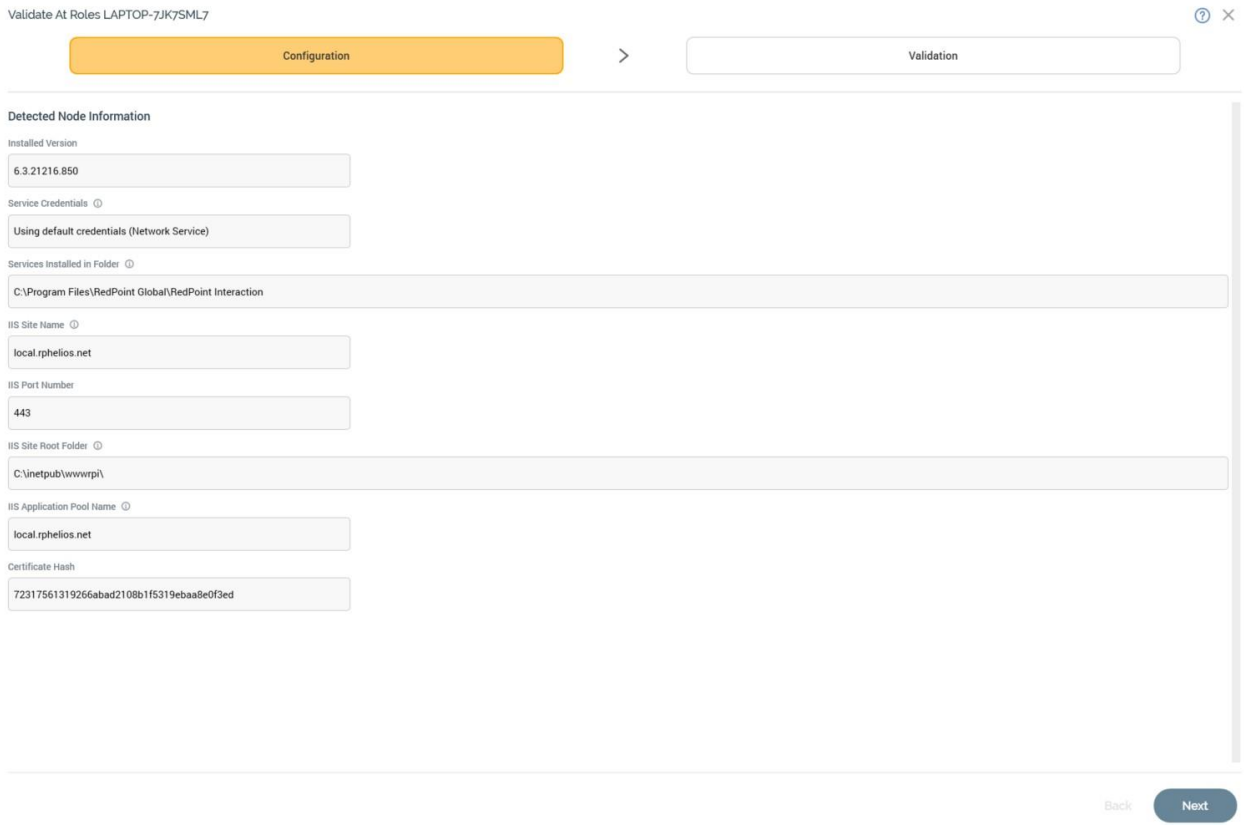
On completing validation, the Next button displayed at the bottom right is replaced by a Finish button. Clicking Finish closes the Validate Core overlay and displays the Server Workbench tab set, which includes all tabs. The Cluster tab is shown by default.

If you click Finish prior to completion of the checklist, you will be advised to wait until all steps are complete.

You can return to the Configure sub-interface using the Back button.

Validating Roles

You may wish to check that everything is installed and running correctly in a given RPI node. You can do so by clicking the Validate Roles context menu button, displayed when hovering over a node. Doing so displays the Validate Roles overlay:



The overlay contains the following:

Header

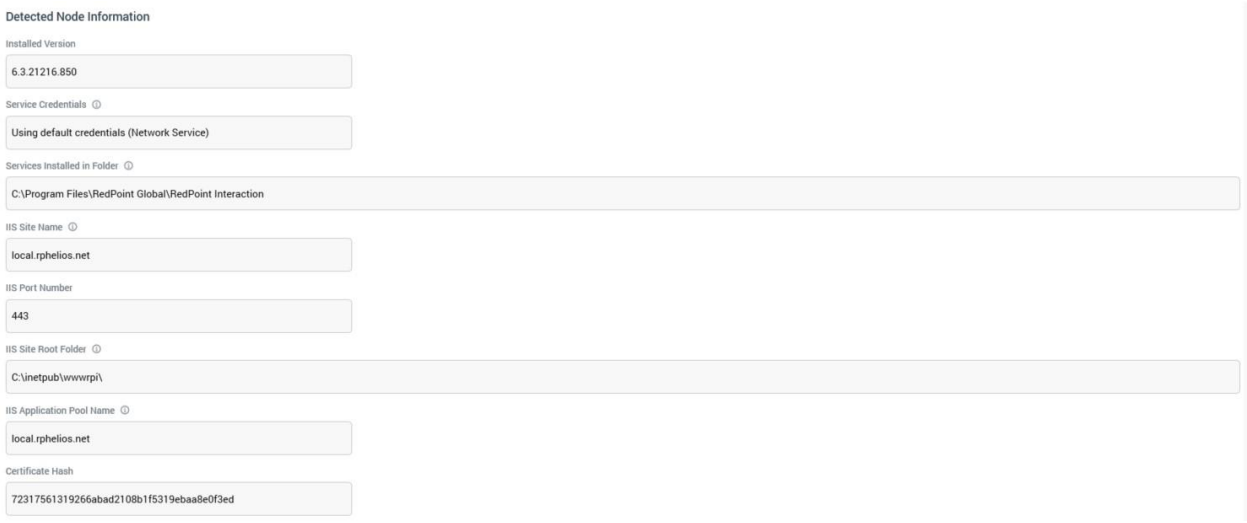
Validation of node roles is carried out in a wizard style, using two sub-interfaces:

- Configuration
- Validation

A graphical representation of current progress through the wizard is displayed within the top section:

Configuration Sub-interface

The Configuration sub-interface allows you to view a read-only summary of details of the roles installed at the current node, prior to performing the validation.



It consists of a single Detected Node Information section.

Detected Node Information

The read-only fields within this section are determined automatically by Server Workbench and displayed for informational purposes. The following properties are displayed:

- Installed Version
- Service Credentials

This property is only shown if the Windows services role Is installed:

- Services Installed in Folder

These properties are only displayed if the Web services and/or Help files roles is/are installed. Note that the Network Service account must be a member of the local Administrators group for IIS settings to be discovered.

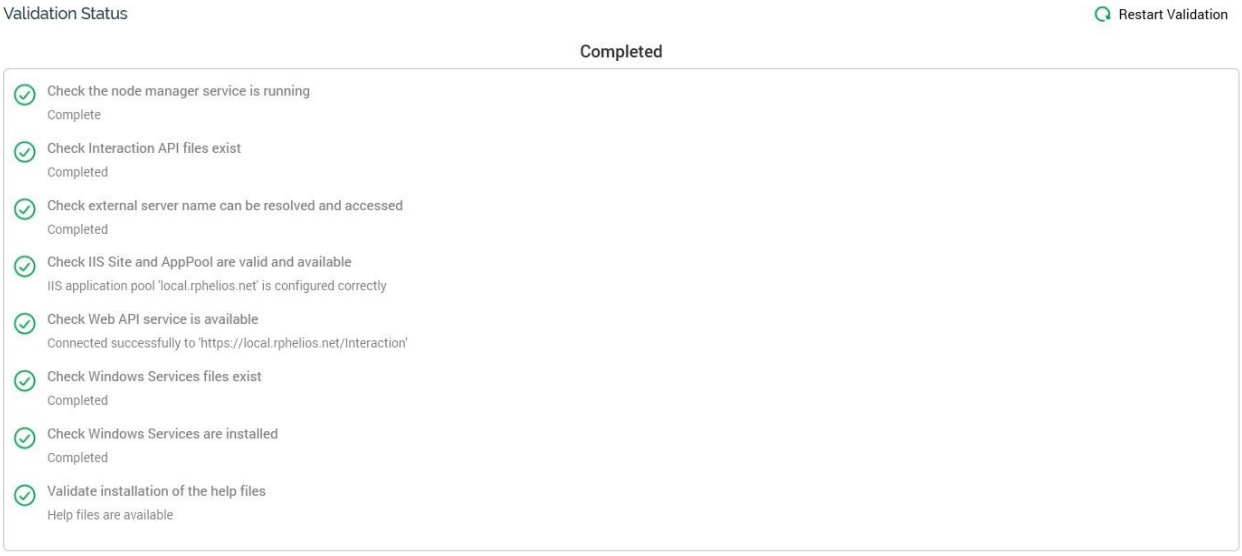
- IIS Site Name
- IIS Port Number
- IIS Site Root Folder
- IIS Application Pool Name
- Certificate Hash

Navigation

You can move forwards through the Validate Roles wizard by clicking on the Next button at the bottom right. Doing so displays the Validation sub-interface.

Validation Sub-interface

This interface is used to provide visibility of the checks performed during validation of node roles.



Toolbar

A single button is shown at the toolbar:

- Restart Validation: clicking this button allows to you repeat the validation checklist.

Validation Status

Details of status information provided at each validation step, and overall progress, are provided in the Core Pre-Install Checklist documentation.

Navigation

On completing validation, the Next button, displayed at the bottom right, is replaced by a Finish button. Clicking Finish closes the Upgrade Roles overlay and displays the Server Workbench tab set, which includes all tabs.

If you click Finish prior to completing validation, you will be advised to wait until all steps are complete.

You can return to the Configure sub-interface using the Back button.

Cluster

To validate the whole cluster, you must perform a Single Stack validation—including validating the core and validating roles—before validating additional cluster nodes. Refer to **Validating the Core** for instructions on validating the initial cluster node. For each additional node in the cluster, you will validate the local node (while logged into the node) and validate roles (refer to **Validating Roles**).

You are, of course, at liberty to simply validate only those elements of the cluster that require your attention.

Working with Roles

Taking a Node Offline	When a node is taken offline, all roles at the node are taken offline.
Putting a Node Online	• When a node is put online, all roles at node are put online. • When a node is offline, putting one of its roles online puts the node as a whole online. • If all of a node's roles are put offline, the node itself becomes offline.
Taking a Node Manager Role Offline	When a Node manager role is taken offline and it is the master node, another node in the cluster assumes the Master node status.
Putting a Node Manager Role Online	When a Node manager role is put online, it is available to assume the status of master node.

Taking a Windows Services Role Offline	• When a Windows services role is taken offline, the role is no longer able to have server jobs assigned to it. Any server jobs owned by the node in question with an InProgressAtHost status remain as such. • At RPI, any Playing workflows controlled by the node are Paused. Playing and Waiting for Trigger activities within workflows are Paused. • An attempt to Play a Paused workflow owned by the node causes it to enter a Resume Play Requested state; invoking Stop in the same scenario causes the workflow to enter a Stop Requested state. • If a scheduled or recurring trigger fires while its owning node is offline, it enters a Play Requested state. If a scheduled wait for event fires in the same context, it enters a Paused state.
Putting a Windows Services Role Online	When a Windows services role is put online, it is once again available to accept server jobs. Any InProgressAtHost jobs re-commence execution. At RPI, any automatically-Paused workflows and the activities they contain are resumed.

Taking a Web Services Role Offline	• Load balancing of web service roles is typically handled by IIS and is outside the scope of this documentation. • When all of a cluster's web services roles are taken offline, users will no longer be able to log into RPI, and users logged into the RPI client application will be unable to communicate with the server.
Putting a Web Services Role Online	At least one web services role must be online in order for users to log into RPI.
Taking a Help Files Role Offline	• Load balancing of help files roles is typically handled by IIS and is outside the scope of this documentation. • When all of a cluster's help files roles are taken offline, users will no longer be able to view RPI help.
Putting a Help Files Role Online	At least one help files role must be online in order for users to view RPI help.

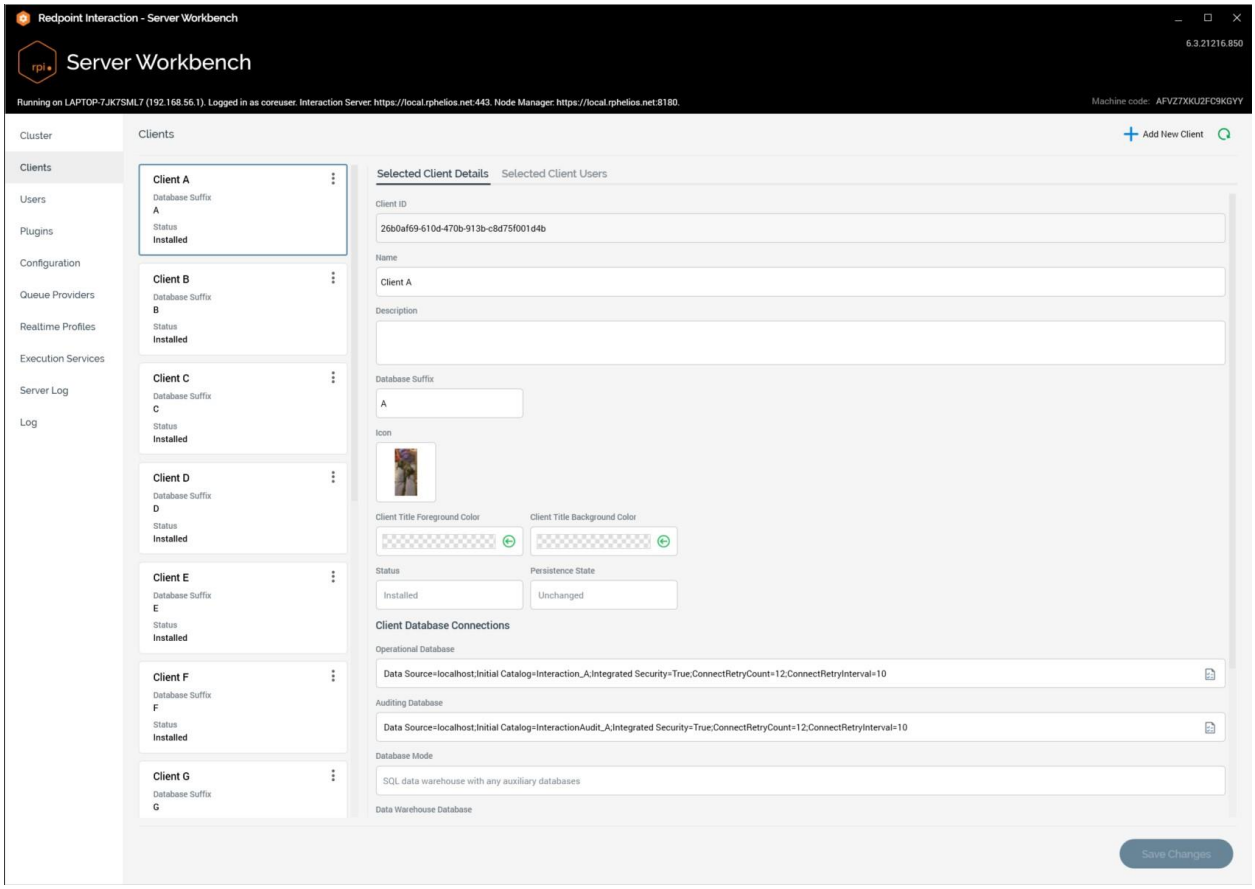
Clients

In the context of Server Workbench, a client represents an organization using RPI within a cluster. Note that the term "client" is not to be confused with the RPI client application, which end users use to manage their customer or prospect communications.

To be used, an RPI cluster must contain one client. Optionally, multiple clients can be created. Each client's RPI installation is entirely separate and wholly isolated from any others within the cluster.

Clients can be associated with many users. A given user may also have access to multiple clients.

Clients are managed within the Server Workbench Clients tab.



The tab consists of the following:

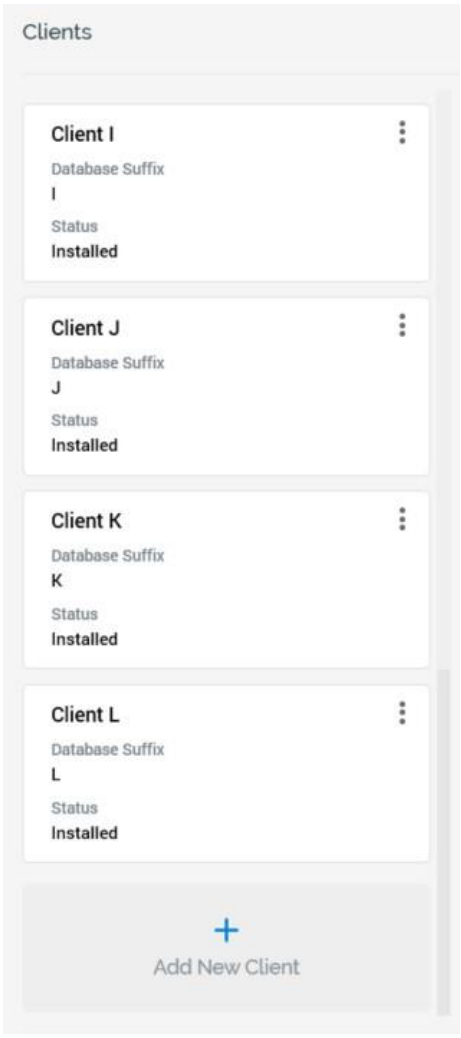
Toolbar

The Client tab toolbar exposes the following options:

- Add New Client: clicking this button adds a new client to the bottom of the Clients list, and shows its details in the right-hand tabset. Its name is ‘New Client’, and Description is ‘This is a new client’.’
- Refresh: clicking this button loads the latest client information from the core operational database. Note that an “Are You Sure?” dialog is shown if unsaved client or user changes are present.

Clients List

A list of existing clients is displayed to the left, at the bottom of which is provided an Add New Client button (behavior of which is as the Add New Client toolbar button (see above):



When no clients have yet been set up in the cluster, the list displays an advisory message ("Add, save and install a new client, then choose the users to associate with it").

The following appears at a client within the list:

- Client name
- Database Suffix

- Status: one of ‘Installed’ or ‘Not Installed’.
- Actions menu: exposing the following:
    - Remove: selecting this option sets the selected client’s status to Removed (unless its status was Added, in which case it is removed). You can no longer select a Removed client. Any Removed clients are deleted permanently on invocation of Save, when an ‘Are You Sure?’ dialog is shown. Note that deletion of an Installed client does not remove any of its operational databases.
- Install Client: this button is available when the currently-selected client’s Status is Not Installed, and Persistence state is Unchanged. Clicking the button displays the Install Client overlay (documented separately).

Clients are ordered alphabetically by name.

Selected Client Tabset

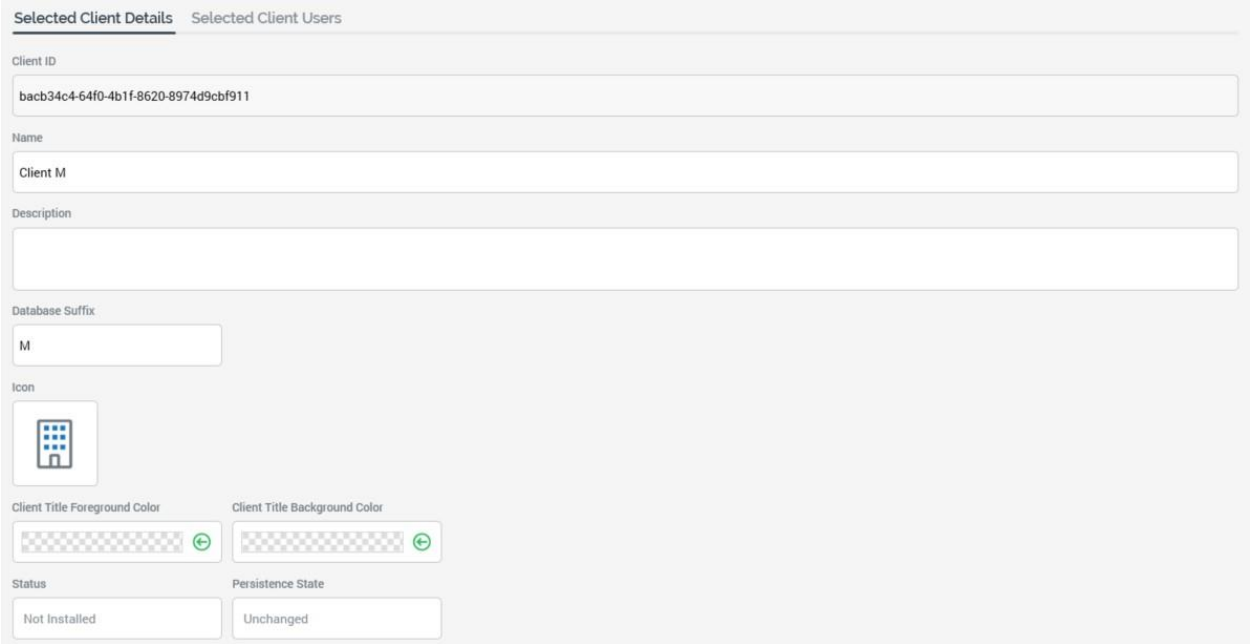
Two tabs are shown:

- Selected Client Details
- Selected Client Users

Each is covered separately.

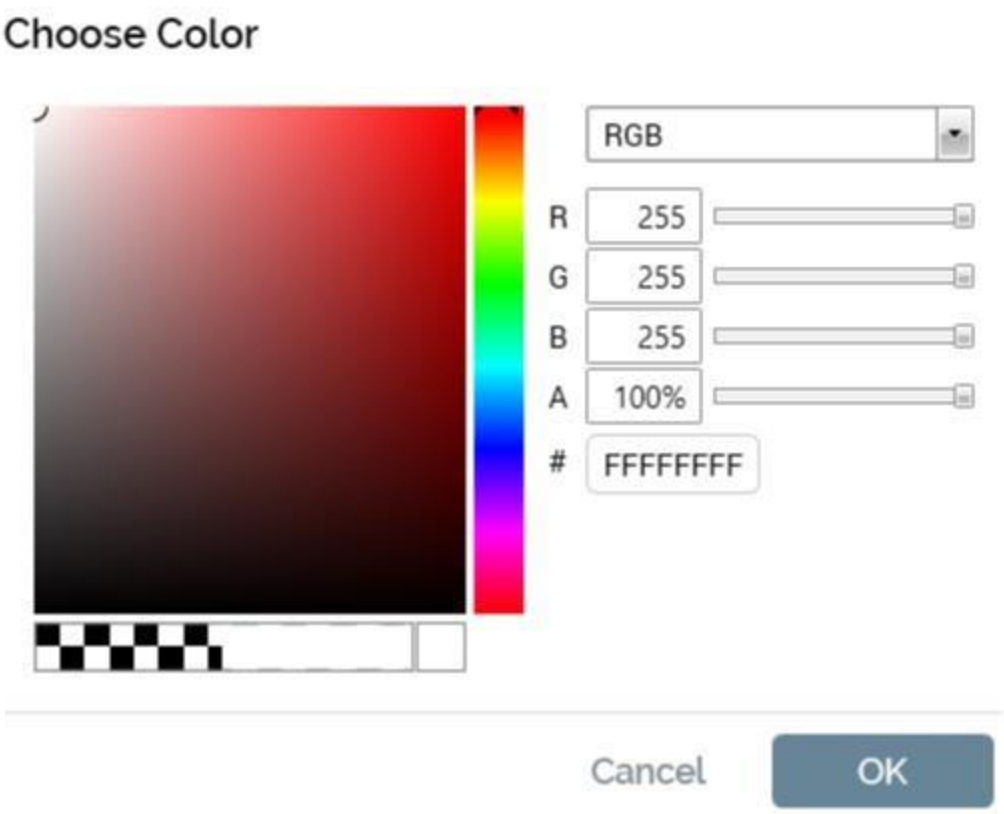
Selected Client Details Tab

This tab is used to manage the main details of the client. When no client is selected in the lefthand Clients list, a message is displayed (“Please select a client from the list to view details here”). The following is shown when a client is selected:



The tab contains the following:

- Client ID: a read-only GUID. Client ID is displayed following the saving of a new client. You can select the value and copy it to the clipboard if required.
- Name: provision of a Name for the client is mandatory. The value provided can be a maximum length of 100 characters and must be unique amongst the clients in the cluster.
- Description: provision of a Description for the client is optional. The value provided can be a maximum length of 1000 characters.
- Database Suffix: provision of a Database suffix is mandatory. The value provided can be a maximum length of 30 characters. Only database-suitable characters are allowed, and the value must be unique within the cluster.
- Icon: a default icon is provided. You can click the accompanying Change button to display a Windows file system dialog, in which you can select an image to serve as the client's new icon.
- Client colors: it is possible to customize the colors used to display the client's name at the top of the RPI client application. Foreground and Background colors can be defined. You can set colors using the Choose Color dialog:



Clicking OK changes the color; clicking Cancel removes the dialog from display without changing the color.

- Revert: this button allows you to revert the client colors back to their defaults.
- Status: one of:
  - Not Installed
  - Installed
- Persistence State: one of:
  - Unchanged ○

  Added ○ Edited ○

  Deleted

Client Database Connections Section

This section contains the following:

- Connection strings: these are only displayed following a client's installation (wherein its operational databases are created). The following properties are shown:
    - ○ Operational Database connection string ○
    Auditing Database connection string ○
    Database Mode: read-only; one of:
        - SQL data warehouse with any auxiliary databases
        - NoSQL databases only
        - NoSQL databases with SQL auxiliary databases ○ Data Warehouse Database connection string ○ Read-only Data Warehouse Database: to improve performance, you can optionally provide an alternative data warehouse string that will be used to perform read-only queries in the following contexts:
            - Selection rule count
            - Analysis Panel refresh
            - Refresh attribute values task
            - Attribute value catalog task
            - Count Results widget
            - Chart widget
            - Data Viewer ○ Data Warehouse Platform (displayed as a read-only string).

    The database connection string fields are writeable (allowing you to manually adjust the connection strings) and are all mandatory.

    If you wish to take advantage of .NET's automatic connection retries feature when configuring a SQL Server Operational Database connection string, please see the following article:

    https://msdn.microsoft.com/enus/library/system.data.sqlclient.sqlconnectionstringbuilder.connectretrycount(v=vs.11 0).aspx

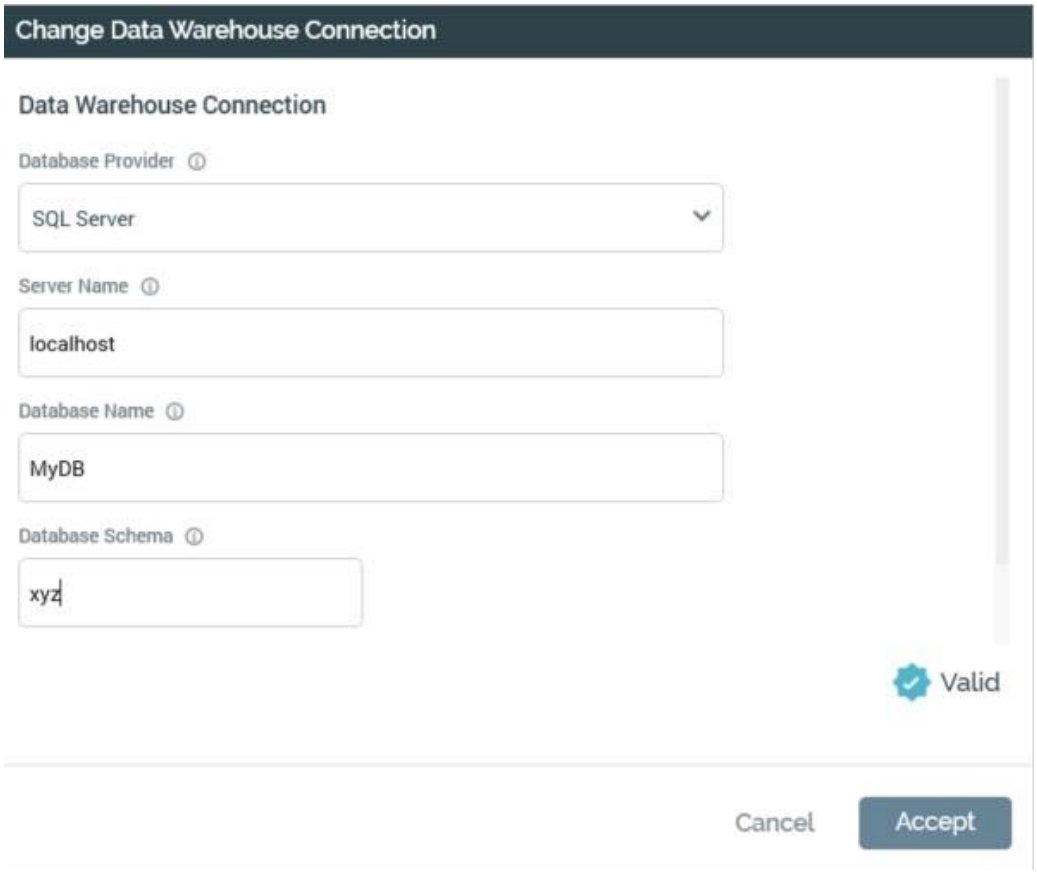
    Note that all SQL Server connection strings automatically include the ConnectRetryCount and ConnectRetryInterval parameters, e.g.:

    ```
    Data Source=localhost;Initial Catalog=Interaction_D;Integrated
    Security=True;ConnectRetryCount=12;ConnectRetryInterval=10
    ```

    Operational and Auditing Database connection strings are accompanied by a single Test the Operational and Auditing database connections button. Clicking the button checks that the supplied credentials are valid. Results are displayed at the top of the Server Workbench interface.

    Data warehouse connection string is accompanied by the following buttons:

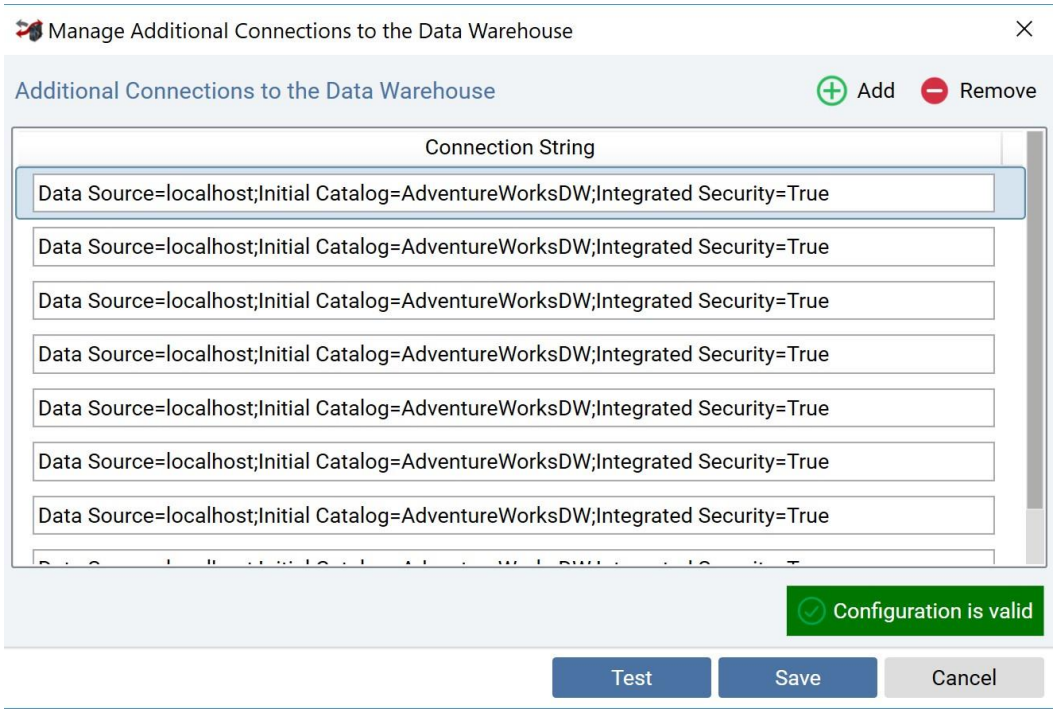
    - Change the Data Warehouse Connection: clicking this button displays the Change Data Warehouse Connection dialog, in which you can specify a new data warehouse's details:



The properties required are contingent on the selection of Database provider. You can click Accept to update the data warehouse connection string. You can also click Cancel to remove th dialog from display without making changes.

- Manage additional connections to the data warehouse: clicking this button displays the Manage Additional Connections to the Data Warehouse dialog, which allows you to define multiple data warehouse connection strings for the current client. When executing database queries during RPI operational usage, if multiple connection strings have been provided, they are picked up for use in a round robin capacity.

  The dialog contains the following:

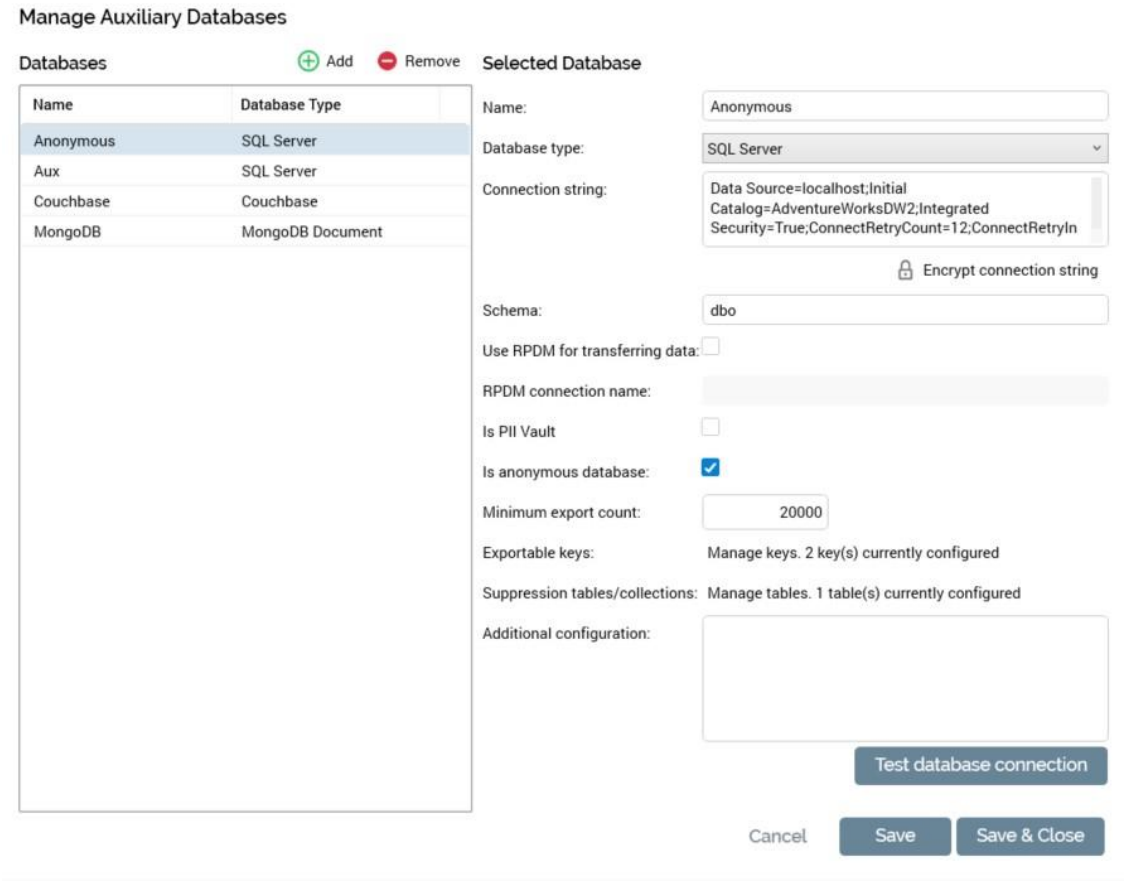


- Toolbar: exposing the following:
  - Add new Connection: clicking this button adds a new row to the list. Its default text is 'New connection string' (an integer is appended if required to ensure uniqueness).
  - Remove selected Connection: clicking this button removes the currentlyselected connection string, without displaying an 'Are You Sure?' dialog.
- List: no additional data warehouse connection strings are provided by default. Provision of additional connection strings is optional. For each row in the list, a Connection String property must be provided.
- Validation status indicator
- Footer: exposing the following buttons:
  - Test: this button is enabled when one or more connections are present in the list. At invocation, the Last Test Result is displayed below the list. If all connection strings are OK, 'All connections are valid' is shown. If one or more connection strings are not OK, the message 'One or more connections are invalid. Hover for further information' is displayed. On hovering over the message, failure details are displayed in a tooltip.
  - Save: clicking this button persists any changes made within the dialog and closes it. An advisory message is displayed at the top of the Client tab. Note that you can save irrespective of as to whether all provided connection strings have been configured correctly.
  - Cancel: clicking this button abandons any changes made in the dialog and closes it.

- Encrypt: clicking this button encrypts the supplied connection string value.
- Test the data warehouse connection: clicking this button checks that the supplied data warehouse credentials are valid. Results are displayed at the top of the Server Workbench interface.

- Auxiliary databases: you can optionally define one or more auxiliary databases to augment the data warehouse, and to serve as an additional source of segmentation data.

The property is accompanied by a Manage Auxiliary Databases button, which, when clicked, displays the Manage Auxiliary Databases overlay:



The overlay contains the following:

○ Databases section: displayed to the left, this consists of a toolbar and a grid.

- Toolbar: exposing the following options:

- Add: invocation of this option adds a new auxiliary database to the grid. Its name is ‘New database’ (an incrementable integer can be added to ensure uniqueness).
- Remove: invocation of this option removes the selected auxiliary database from the grid without display of an ‘Are You Sure?’ dialog.

Grid: containing the following read-only columns:

- Name
- Database Type ○ Selected Database section: displayed to the right, and containing the following:

 Name: a mandatory property that can be a maximum of 100 characters in length. Name must be unique.

 Database Type: a dropdown that exposes the following values:

- Actian VectorH
- Amazon Athena
- Amazon Aurora
- Amazon Redshift Spectrum
- Apache Cassandra
- Apache Drill (also use for MapR)
- AWS Redshift
- Azure Cosmos SQL
- Azure MySQL
- Azure PostgreSQL
- Azure SQL Server
- Azure Synapse Analytics
- Couchbase
- DB2
- Dynamo DB
- Google Big Query
- Google Big Table
- Google Datastore
- Greenplum
- Apache Hive
- MariaDB
- MarkLogic
- MongoDB SQL
- MongoDB Document (also used for CosmosDB, Amazon DocumentDB)
- MySQL
- Netezza
- Oracle
- PostgreSQL
- Presto
- SalesforceDB
- SAP HANA

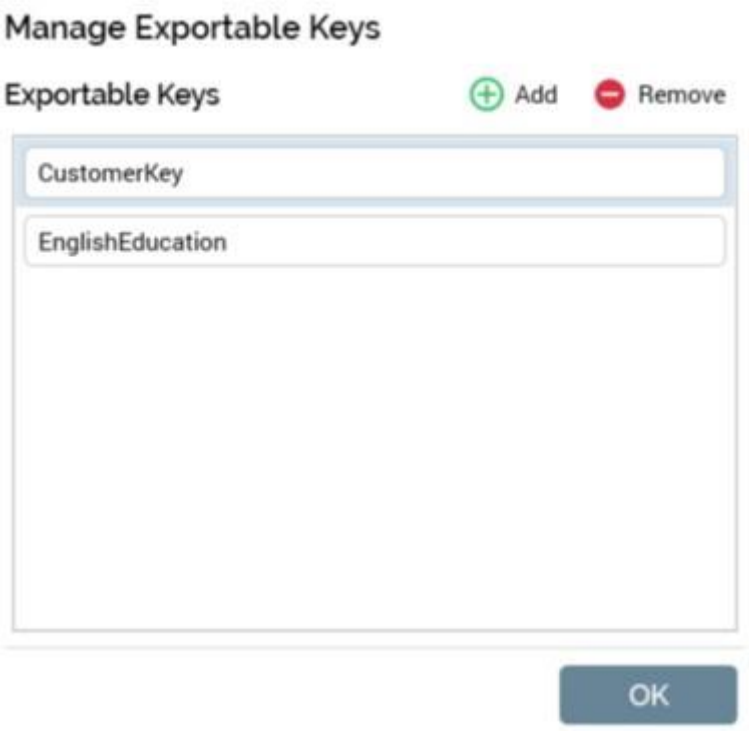
- Snowflake
- SparkSQL
- SpliceMachine
- SQL Server (the default)
- SQL Server PDW
- Sybase IQ
- Teradata v15
- Teradata v16
- Vertica

- Connection String: this mandatory property can be a maximum of 1000 characters in length.
- Encrypt connection string: clicking this button encrypts the current connection string.
- Schema: this mandatory property can be a maximum of 100 characters in length.
- Use RPDM for transferring data: this property is shown for SQL auxiliary databases only. A checkbox, it is unchecked by default. When checked, the RPDM connection name property is enabled. The property allows you to make use of RPDM when transferring keys between databases, which may realize performance improvements in this context. Note that, to use RPDM to transfer keys between databases, the relevant client system configuration settings must be configured correctly. These include:
  - DataManagementCredentials
  - DataManagementDataConnection
  - DataManagementServerName
  - DataManagementVersionMajor
- RPDM connection name: this property is enabled when Use RPDM... is checked. It is used to specify the name of the RPDM connection to be used when transferring keys between databases. It is mandatory when enabled.
- IS PII Vault: this checkbox is unchecked by default. When checked, the auxiliary database will be used to host personally identifiable information (PII) only. When using data from the database as RPI attributes, all of the same will be marked as PII, and will be subject to restrictions as documented in the RPI Reference Guide.
- Is anonymous database: this checkbox is unchecked by default. When checked, the Is attribute export enabled checkbox is not shown, and the Minimum export count, Exportable keys and Suppression tables properties are displayed. Anonymous auxiliary databases are primarily intended for user when integrating with the LiveRamp data management platform as part of Redpoint's Digital Acquisition Platform.
- Minimum export count: this property is displayed when Is anonymous database is checked. It is used to define the minimum number of records that can be exported in a single file from an anonymous auxiliary database. A mandatory integer property, it defaults to the value 25, and accepts minimum and maximum values of 0 and 9,999,999, respectively.

-

Exportable keys: this button is shown when Is anonymous database is checked. The button's text reads 'Manage keys. [n] key(s) currently configured'. Clicking the button displays the Manage Exportable Keys dialog.

- Manage Exportable Keys dialog: this dialog is used to manage an anonymous auxiliary database's exportable keys, which define the specific data items that can be exported from an anonymous auxiliary database. The dialog contains a toolbar and an Exportable Keys list.

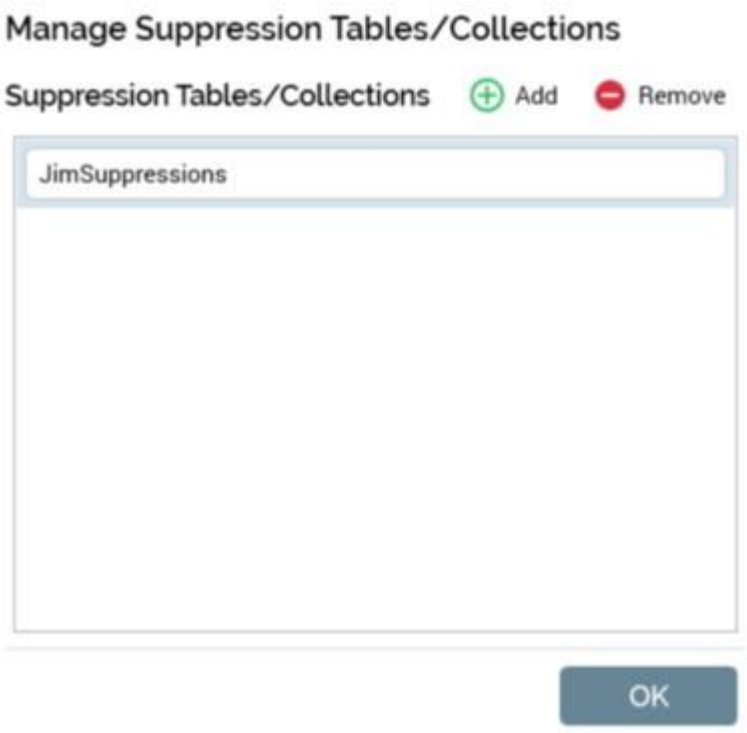


  - Toolbar: exposing the following:
    - Add new Exportable Key: clicking this button adds a new exportable key to the list. Its default name is 'Key'. Uniqueness is ensured through the appending of an incrementable integer.
    - Remove selected Exportable Key: this button is enabled when an exportable key has been selected. Clicking it removes the exportable key from the list. An 'Are You Sure?' dialog is not shown.
  - List: all of the anonymous auxiliary database's exportable keys are listed. A single key ('identitylink') is included by default. Provision of export keys is optional. If a key is provided, it must be supplied with a value, and all keys' values must be unique. The maximum supported value length is 100 characters.

-

  Suppression tables/collections: this button is displayed when Is anonymous database is checked. The button's text reads 'Manage tables. [n] key(s) currently configured'. Clicking the button displays the Manage Suppression Tables/Collections dialog.

- Manage Suppression Tables/Collections FD: this dialog is used to manage an anonymous auxiliary database's suppression tables (SQL) or collections (NoSQL). Such tables/collections can be used only for the application of suppressions when executing queries against an anonymous auxiliary database. The dialog contains a toolbar and a Suppression Tables/Collections list.



  - Toolbar: exposing the following:
    - Add new Suppression Table/Collection: clicking this button adds a new entry to the list. Its default name is 'Key'. Uniqueness is ensured through the appending of an incrementable integer.
    - Remove selected Suppression Table/Collection: this button is enabled when a list entry has been selected. Clicking it removes the entry from the list. An 'Are You Sure?' dialog is not shown.
  - List: all of the anonymous auxiliary database's suppression tables/collections are listed. Provision of the same is optional. If a suppression table/collection is provided, it must be supplied with a value, and all values must be unique. The maximum supported value length is 100 characters.

-

Is attribute export enabled: this checkbox is unchecked by default. When unchecked, auxiliary attributes cannot be used at the client in the following contexts:

- Export Template Designer: ○ Attributes
- Offer Designer:
  ○ Data Extract offer:

- Additional Export Attributes
  - Configuration.Audience Definitions:
    ○ Offer history attributes ○ Unable to select auxiliary Transactional resolution level

- Additional configuration: this property is currently applicable to Cassandra auxiliary databases only. It is blank by default. You can provide JSON configuration to be used by a Cassandra database as per the following example:

```
{
 "ConsistencyLevel": "One",
 "LoadBalancingPolicy": "TokenAwarePolicy",
 "ChildLoadBalancingPolicy": "DCAwareRoundRobinPolicy",
 "LocalDC": ""
}
```

Details of the elements within the JSON follow:

- ConsistencyLevel: values appropriate to this element are available at **https://docs.datastax.com/en/drivers/csharpdse/2.1/html/T_Dse_ConsistencyLevel.htm**.
- LoadBalancingPolicy: the following values are supported:
  ○ RoundRobinPolicy ○ DCAwareRoundRobinPolicy ○ TokenAwarePolicy
- ChildLoadBalancingPolicy: if LoadBalancingPolicy is set to TokenAwarePolicy, the following ChildLoadBalancingPolicy values are supported:
  ○ RoundRobinPolicy ○ DCAwareRoundRobinPolicy
- LocalDC: optionally used for DCAwareRoundRobinPolicy

- Test database connection: on clicking this button, a message advises that the connection tested OK or not OK, as appropriate.

The following buttons are displayed at the overlay's footer:

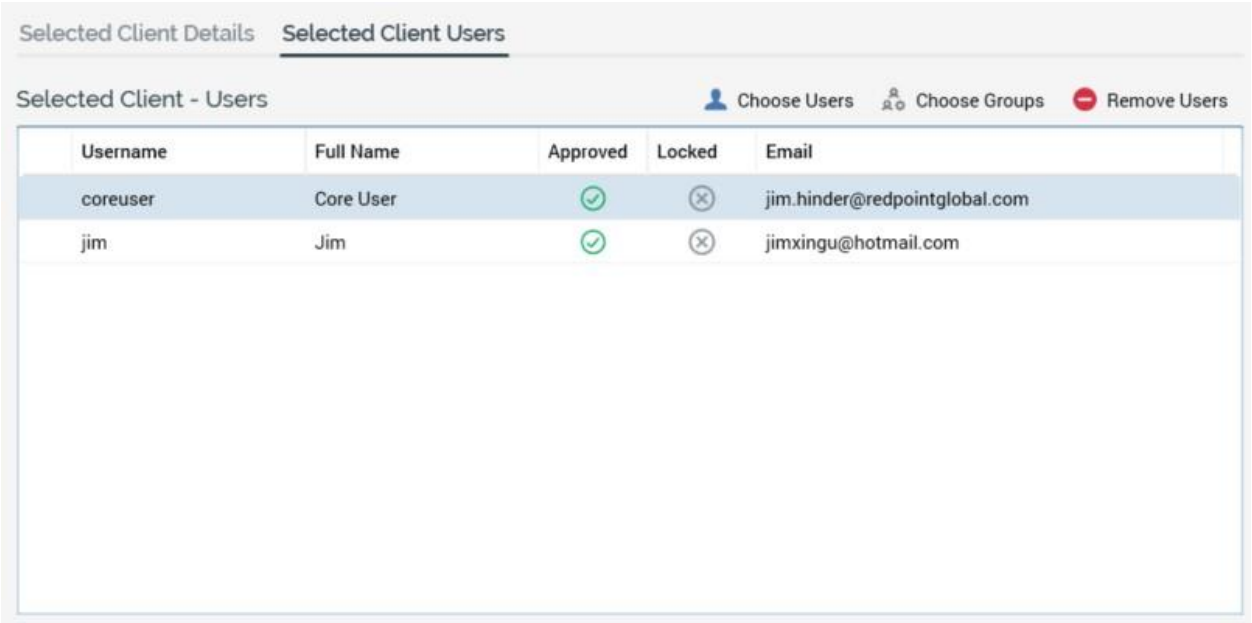
- Cancel: if changes have been made to auxiliary databases, invocation of this option is protected by an Are You Sure? dialog. Invocation removes the overlay from display without saving any changes.

- Save: saves changes to auxiliary databases; does not remove the overlay from display. Following a successful save, an advisory message is shown. ○ Save & Close: saves changes to auxiliary databases and closes the overlay. Following a successful save, an advisory message is shown.

Note that you cannot configure two Cassandra databases, which point to two different Cassandra key spaces, at the same RPI client.

Selected Client Users Tab

The users tab is used to manage users associated with the client. It contains a toolbar and a list of users:



A toolbar is shown above the list, and exposes the following buttons:

- Choose Users: clicking this button displays the “Assign Users to [Client]” dialog (documented separately). The dialog allows you to manage the assignment of users to the currently-selected client.
- Choose Groups: clicking this button displays the “Assign Groups to [User] in [Client]” dialog (documented separately). The dialog allows you to manage the assignment of user groups to the currently-selected user in the context of the currently-selected client. The button is available when a user is selected, and the current client’s Persistence state is Unchanged.
- Remove Users: this button is available when one or more user(s) is/are selected. Clicking it removes the selected user(s) from the Users list.

The Users list itself displays a read-only list of those users associated with the currentlyselected client. Users are displayed in alphabetical order by username. The following user properties are shown:

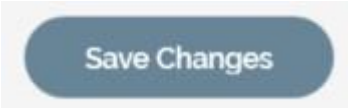
- Status icon
- Username
- Full name
- Approved
- Locked

- Email Address

Clicking on a user in the list displays the Assign Groups dialog (documented separately).

Footer

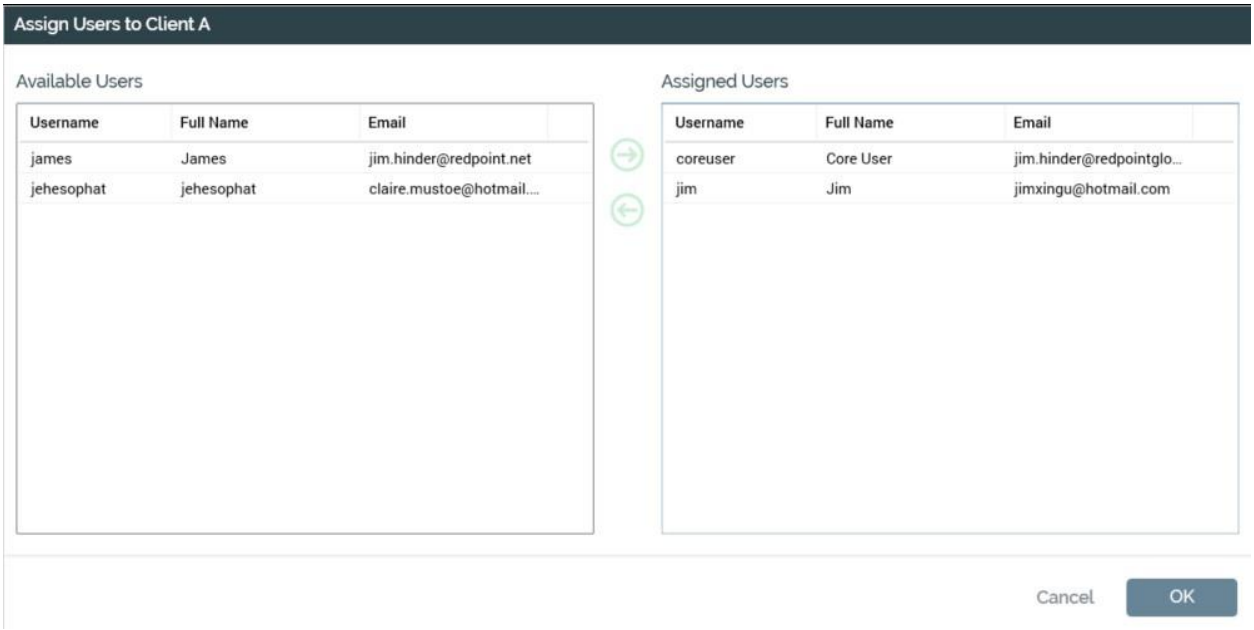
A single button is displayed below the Selected Client section:



The button is available when outstanding changes exist in the Clients or Users tabs. Clicking Save persists any client and user changes to the core operational database. Any displayed status icons are removed.

Assign Users

The Assign Users to [Client] dialog, displayed on invocation of Choose Users at the Users list in the Client tab, allows you to manage the assignment of users to the currently-selected client.



On invocation, the dialog presents two lists of users: to the left, a list of those users within the cluster that are not currently assigned to the selected client; and to the right, a list of those already assigned. On initial display, the contents of the two lists reflect the current state of user assignment at the client.

Available Users Section

The list to the left displays users that have not yet been associated with the current client. Users are listed alphabetically. The following properties are shown:

- Username
- Full Name
- Email

User Buttons

Two buttons, shown between the two lists, control the assignment of users to, and removal of users from, the current client:

- Assign the selected Users to this Client: this button is enabled when one or more users are selected in the Available Users grid. Invocation moves the users to the Assigned Users grid.
- Remove the selected Users from this Client: this button is enabled when one or more users are selected in the Assigned Users grid. Invocation moves the users to the Available Users grid.

Note that double-clicking a user in either list has the effect of removing the user from that list and moving it to the other list.

Assigned Users Section

The list to the right displays users that have been associated with the current client. Users are listed alphabetically. The following properties are shown:

- Username
- Full Name
- Email

OK and Cancel

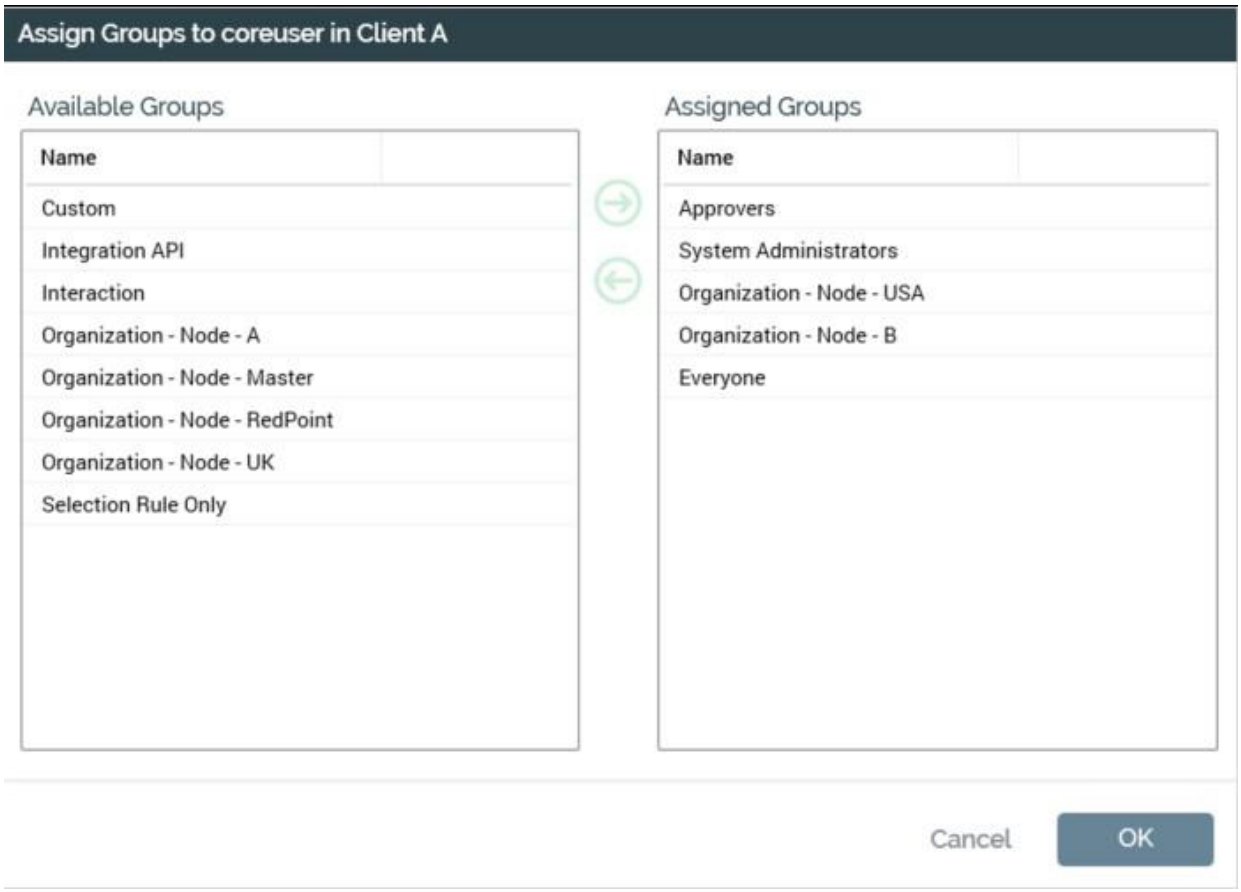
Clicking the OK button at the bottom of the dialog closes the dialog. Any changes to the client's user community are reflected in the Users list. If changes were made, the client's Persistence state is set to Edited, and it is necessary to click Save to make the changes permanent.

Clicking Cancel closes the dialog without saving.

Assign Groups

The Assign Groups dialog, displayed on invocation of Choose Groups at the Users list in the Client tab, allows you to manage the assignment of user groups to the currently selected user, in the context of the currently selected client. This last point is significant, as a given user may have access to more than one client but perform different roles in each.

The list of user groups made available in the dialog is, by default, limited to the Everyone and System Administrators groups created at client installation. In addition, any custom groups created manually in the User Groups administration interface in the RPI client application are also displayed.



On invocation, the dialog presents two lists of user groups: to the left, a list of those user groups that are not currently assigned to the selected user in the current context; and to the right, a list of those already assigned. On initial display, the contents of the two lists reflect the current state of user group assignment at the user in the context of the current client.

Available Groups Section

The list to the left displays user groups that have not yet been associated with the current user. Groups are listed alphabetically. A single property (Name) is shown.

User Groups Buttons

Two buttons, shown between the two lists, control the assignment of user groups to, and removal of user groups from, the current user:

- Assign the selected Groups to this User: this button is enabled when one or more user groups are selected in the Available Groups grid. Invocation moves the groups to the Assigned Groups grid.
- Remove the selected Groups from this User: this button is enabled when one or more user groups are selected in the Assigned Groups grid. Invocation moves the groups to the Available Groups grid. Note that you cannot remove the Everyone group from a user.

Note also that double-clicking a group in either list has the effect of removing the group from that list and moving it to the other list.

Assigned Groups Section

The list to the right displays user groups that have been associated with the current user in the context of the current client. Groups are listed alphabetically. A single property (Name) is shown.

OK and Cancel

Clicking the OK button at the bottom of the dialog closes the dialog, and immediately saves any changes made therein to the core operational database (note the difference in behavior from the Assign Users dialog).

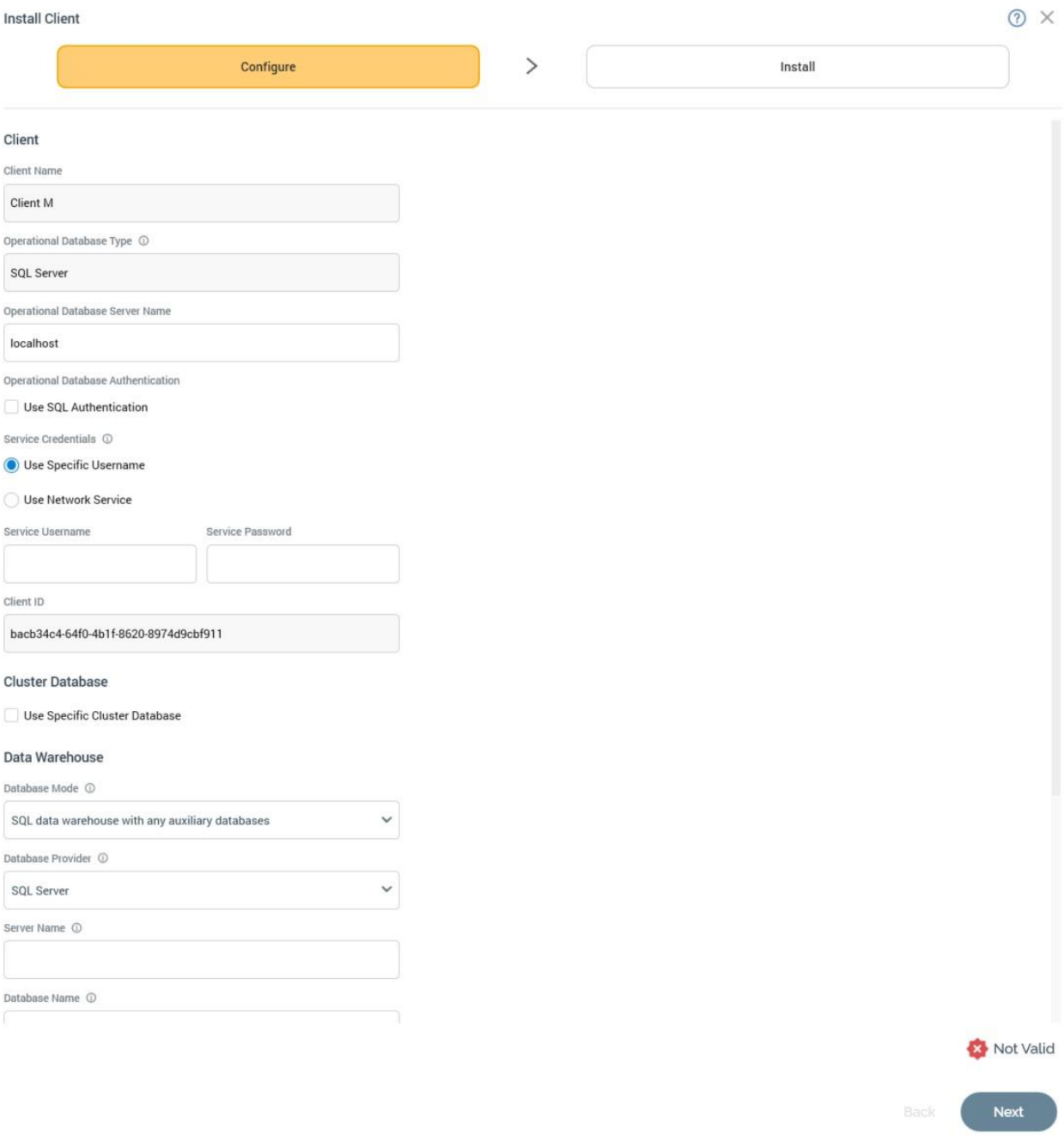
Clicking Cancel closes the dialog without saving.

Installing a Client

Having added a client to the cluster, you need to install it. Installing a client creates the operational databases required to persist any RPI collateral created by the client's users, and also allows you to capture the details of the data warehouse to which the client's users will connect to undertake customer or prospect communications.

Note that, unlike the core and node roles, there is no equivalence for updating a client.

On highlighting a client with a Status of Not Installed and Persistence state of Unchanged, the Install button is enabled. Clicking it displays the Install Client overlay.



The overlay contains the following:

Install Client Section

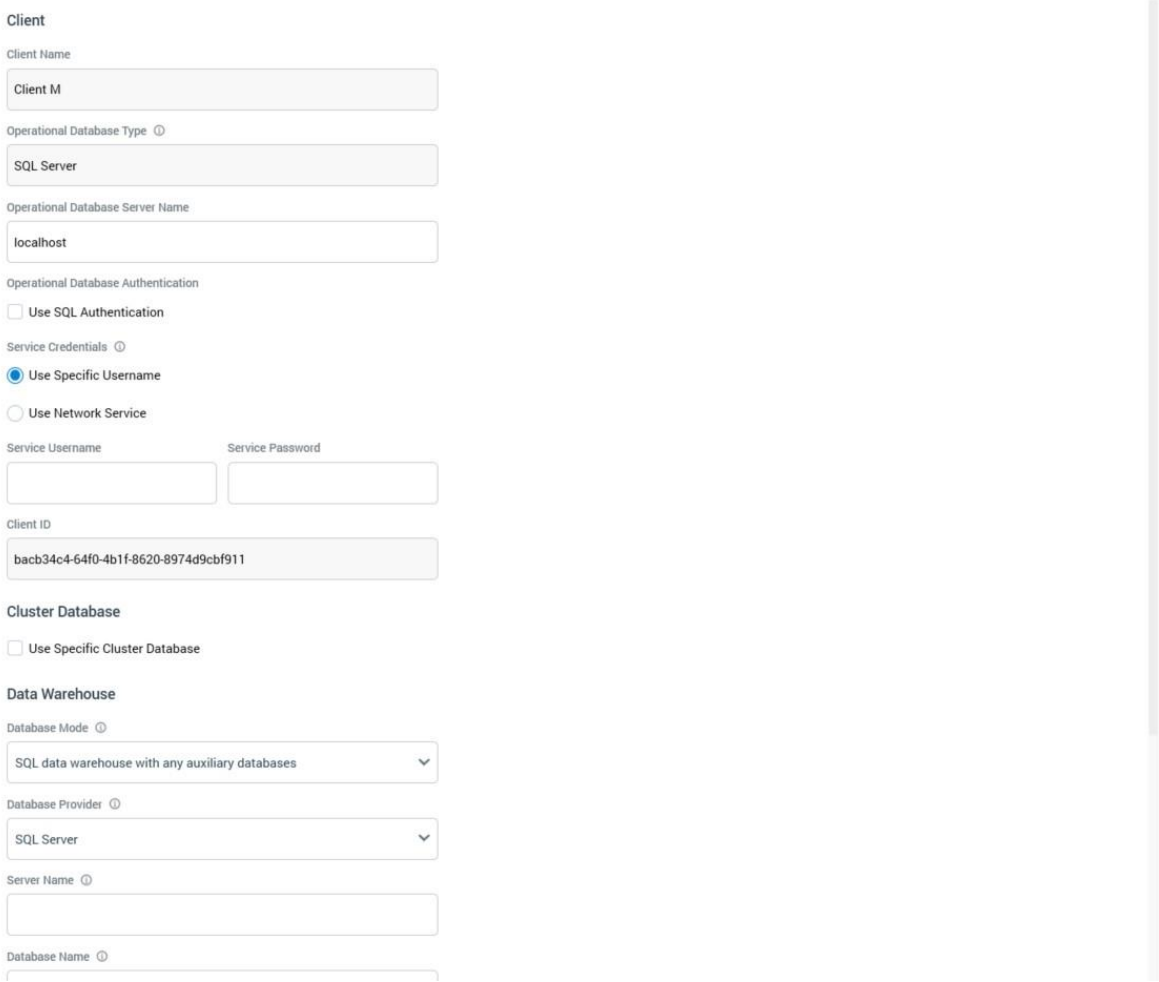
Installation of a client is carried out in a wizard style, using two sub-interfaces:

- Configure
- Install

A graphical representation of current progress through the wizard is displayed within the top section:

Configure Sub-interface

The Configure sub-interface allows you define a series of properties required by Server Workbench in advance of installing the client.



It consists of Client, Cluster Database, Data Warehouse and End User License Agreement sections.

Client Section

This section allows you to define properties relating to the client's operational databases (Interaction_XXX and InteractionAudit_XXX) and service credentials. The following are shown:

- Client Name: read-only.
- Operational Database Type: this read-only property displays the platform on which the operational databases are installed.
- Operational Database Server Name: the name of the server on which the client's operational databases are to be installed.

- Operational Database Authentication: containing a Use SQL authentication checkbox. When unchecked, connection will be made using Windows authentication. When checked, a SQL Server login username and optional password can be provided instead.
- Service Credentials: two radio buttons are available:
  - Use Specific Username: selected by default. When chosen, you can provide a domain Service Username (with sufficient privileges), that will be used to run the client's Windows services. An optional Service Password can also be supplied. ○ Use Network Service: when selected, the client's services will run under the current machine's Network Service account. The account must be a member of the local Administrators group.
- Client ID: a read-only GUID.

Cluster Database Section

This section allows you to specify the location of the cluster's operational databases (Pulse and PulseLogging).

- Use Specific Cluster Database: this checkbox should be checked in the event that the cluster operational databases are located on a server different to the client operational databases. It is unchecked by default. When checked, the Operational database type, Cluster database server name and Use SQL authentication properties are displayed.
- Operational Database Type: this read-only property displays the platform on which the operational databases are installed.
- Cluster Database Server Name: this property is displayed when Use specific cluster database is checked. It is blank by default, and mandatory if displayed.
- Operational Database Authentication: containing a Use SQL authentication checkbox. This checkbox is displayed when Use specific cluster database is checked. It is unchecked by default. If checked, the Operational Database Username and Password properties are displayed.
- Operational Database Username: this property is displayed when Use SQL authentication is checked. It is blank by default, and mandatory if displayed.
- Operational Database Password: this property is displayed when Use SQL authentication is checked. An optional password-masked field, it is blank by default.

Data Warehouse Section

This section allows you to capture details of the data warehouse to which the client will connect. The following properties are shown:

- Database Mode: this dropdown property exposes the following values:
  - SQL databases with any auxiliary databases (the default) ○ NoSQL databases only ○ NoSQL data

warehouse with SQL auxiliary databases

- Database Provider: a dropdown field that allows you to choose the type of database. Available values are constrained by the selected Database mode.

  If in SQL database mode, the following databases are available:

  - Actian VectorH ○ AWS Redshift ○ Azure Database for MySQL ○ Azure Database for PostgreSQL ○ Azure Synapse Analytics ○ Azure SQL Database ○ DB2 ○ Google BigQuery ○ GreenPlum ○ MariaDB ○ MySQL ○ Netezza ○ Oracle ○ PostgreSQL ○ SAP HANA ○ Snowflake ○ SpliceMachine ○ SQL Server (default) ○ SQL Server PDW ○ Sybase IQ ○ Teradata v15 ○ Teradata v16 ○ Vertica

  If in a NoSQL database mode, the following databases are available: ○ Couchbase

  - MongoDB Document (also used for CosmosDB)

If SQL Server is selected, the following properties are shown:

-
-
  Server name
  Database name
- Database schema
- Use SQL authentication: checkbox; when checked, enables:
  - Username ○ Password

If Netezza is selected, the following properties are shown:

- Data Source Name (DSN)
- Database schema

If Oracle is selected, the following properties are shown:

- Server name
- Database name
- Database schema
- Use TNS name: if checked, the Server name will be used as the TNS name.
- Use load balancing: checkbox
- Use explicit credentials: checkbox; when checked, enables: ○ Username ○ Password

If Teradata v15 or v16 is selected, the following properties are shown:

- Server name
- Database schema
- Use explicit credentials
- Username
- Password

If GreenPlum is selected, the following properties are shown:

- Data Source Name (DSN)
- Database schema

If MySQL is selected, the following properties are shown:

- Data Source Name (DSN)
- Database schema

If Sybase IQ is selected, the following properties are shown:

- Server name
- Database name

-
-

  Database schema
  Use explicit credentials: checkbox; when checked, enables: ○
  Username ○ Password

If AWS Redshift is selected, the following properties are shown:

- Data Source Name (DSN)
- Database schema

If PostgreSQL is selected, the following properties are shown:

- Data Source Name (DSN)
- Database schema

If Actian VectorH is selected, the following properties are shown:

- Data Source Name (DSN)
- Database schema

If DB2 is selected, the following properties are shown:

- Data Source Name (DSN)
- Database schema

If Splice Machine is selected, the following properties are shown:

- Data Source Name (DSN)
- Database schema

If Azure SQL Database is selected, the following properties are shown:

- Server name
- Database name
- Database schema
- Use explicit credentials: leave this unchecked to use Azure Managed Identity.
- Username: shown if Use explicit credentials checked
- Password: shown if Use explicit credentials checked If Vertica is selected, the following properties are shown:

- Server name
- Database name
- Database schema

If MariaDB is selected, the following properties are shown:

- Data Source Name (DSN)

-
-
- Database schema

If Azure SQL Database is selected, the following properties are shown:

Server name
Database name

- Database schema

If SAP HANA is selected, the following properties are shown:

- Server name
- Database name
- Database schema

If Azure Database for MySQL is selected, the following properties are shown:

- Data Source Name (DSN)
- Database schema

If Azure Database for PostgreSQL is selected, the following properties are shown:

- Server name
- Database name
- Database schema
- Use explicit credentials: checkbox; when checked, enables:
    - Username ○ Password

If Google BigQuery is selected, the following property is shown:

- Database schema: defaults to the value ‘default’

If Snowflake is selected, the following properties are shown:

- Password
- DSN
- Database schema

If MongoDB document is selected, the following property is shown:

- Connection configuration: this multiline field is mandatory if displayed. It defaults to the following:
  ‘{ "ConnectionString": "mongodb://localhost:27017", "DatabaseName": "_dbname_" }’

  This option is also used if connecting to an Amazon DocumentDB database. An example connection string:

  ‘{"ConnectionString": "mongodb://rpiadmin:vh93e3cXqRYi@docdb-manila.cxlh5tkbvodq.apsoutheast-2.docdb.amazonaws.com:27017?ssl=false&replicaSet=rs0","DatabaseName":

-
-

"rpi"}’.

Note that the RPI server must be hosted on an Amazon EC2 Windows server within the same VPC.

If Couchbase is selected, the following property is shown:

-

  Connection configuration: this multiline field is mandatory if displayed. It defaults to the following:
  ‘{ "Username": "_username_", "Password": "_password_", "servers" : ["http://localhost:8091/pools"] }’

To leverage PartitionKey functionality at a CosmosDB database, please configure the connection string as per the following example:

```
{
   "ConnectionString":"mongodb://...",
   "DatabaseName":"sample",
   "BatchUploadSize":1000,
   "UseBatchForUpdates":true,
   "PartitionKey":{
      "Mode":"CombinedProperties",
      "Keys":["FieldName1","FIeldName2"],
      "Prefix":"RPI"
   }
}
```

If using PartitionKey at interaction workflow execution, a partitionKey property will be appended for every NoSQL document created within a CosmosDB collection when the PartitionKey setting is configured within the Cosmos DB database's connectionstring, and one or more Keys are found in the document's properties. The partitionKey property will contain a value sourced as follows:

If Mode is set to 'CombinedProperties', Key will be set to the concatenated values of Keys found in the NoSQL Document.

If Mode is set to 'RandomSuffix', Key will be set to a random GUID.

If Mode is set to 'PreCalculatedSuffix', Key will be set to a concatenation of Prefix and a random GUID.

Keys are delimited using the underscore character. Values are always persisted as text.

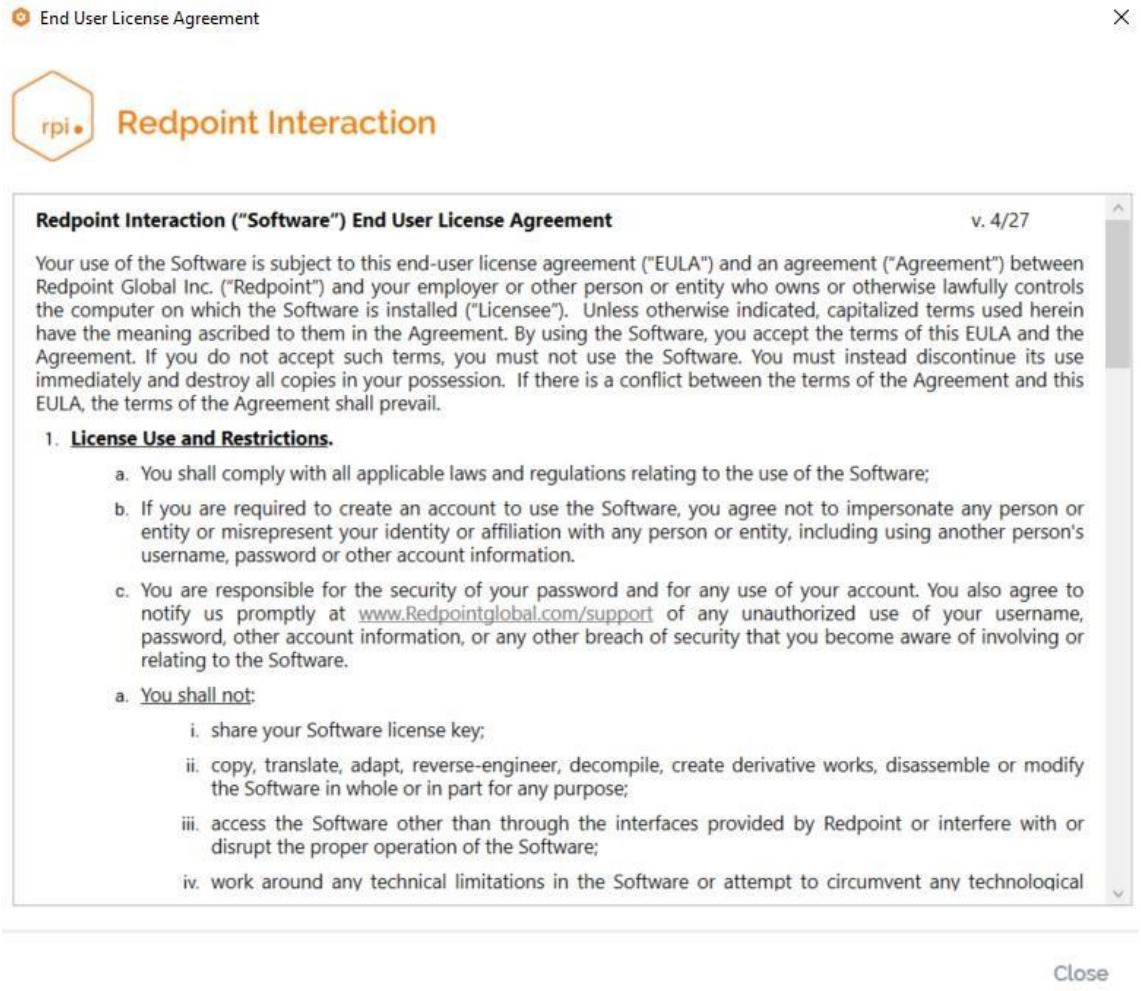
If using PartitionKey in a realtime cache context, a partitionKey property will be created for every NoSQL document in the specified CollectionName. The partitionKey value will be sourced from the NoSQL document that matches the specified PartitionKeys (please see the MongoDB Cache configuration documentation). If there is no matches, a partitionKey will not be created.

Irrespective of the type of data warehouse specified, the requisite database drivers must be installed in order for connection to be made.

End User License Agreement Section

This section is displayed at the bottom of the Configure sub-interface. It contains the following:

- Check this box to accept the Redpoint Interaction End User License Agreement: this checkbox is unchecked by default. You must accept the EULA before proceeding with installation of the client.
- View Redpoint Interaction End User License Agreement: clicking this button displays the End User License Agreement in a separate Window.



You can close the Window using the button at the bottom right.

Validation

If you have not correctly completed the mandatory fields within the Configure sub-interface, a validation errors indicator is displayed to the bottom right:

Clicking the indicator displays details of the errors in a dialog.

You cannot progress to the Install interface when outstanding validation errors remain.

If the configuration is valid, a green indicator is shown:

You may now move to the next wizard step.

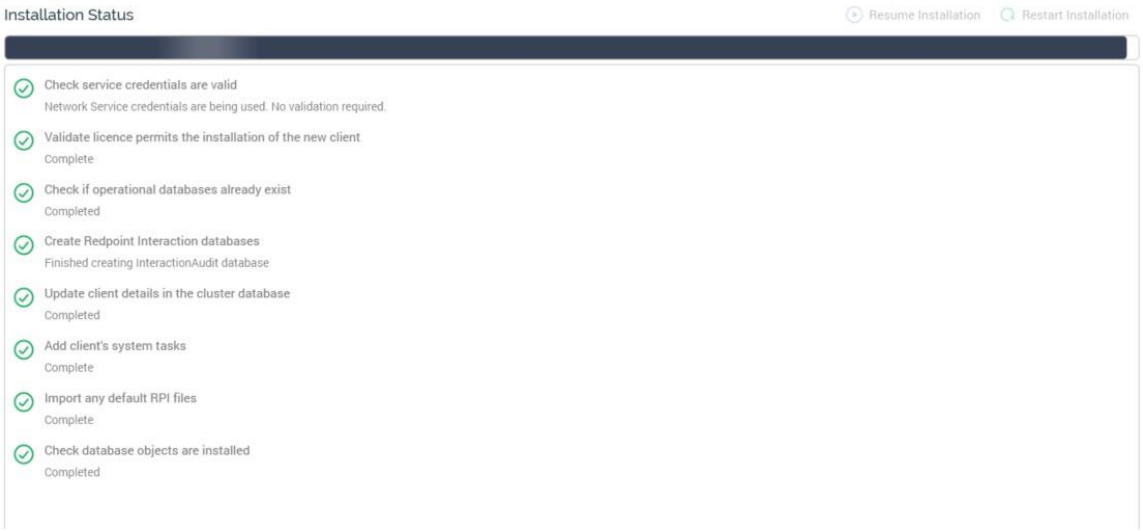
Navigation

You can move forwards through the Install Client wizard by clicking on the Next button at the bottom right. Doing so presents an “Are You Sure?” dialog, prior to display of the Install subinterface.

If you click Next without accepting the EULA, a message (‘You must accept the End User License Agreement before installing the software for this client’) is shown, and you are unable to proceed with the installation.

Install Sub-interface

This interface is used to provide visibility of the tasks executed during installation of a client.



Toolbar

Two buttons are shown at the toolbar:

- Resume Installation: this button is available when the installation’s status is Has Been Paused on Warnings. It allows you to carry on the installation at the point where the warning occurred.
- Restart Installation: this button is available when the status is one of Completed with Warnings or Has a Dependent Job with Errors. It allows you to repeat the installation steps from the beginning.

Installation Steps

Details of status information provided at each installation step, and overall progress, are provided in the Core Pre-Install Checklist documentation.

Navigation

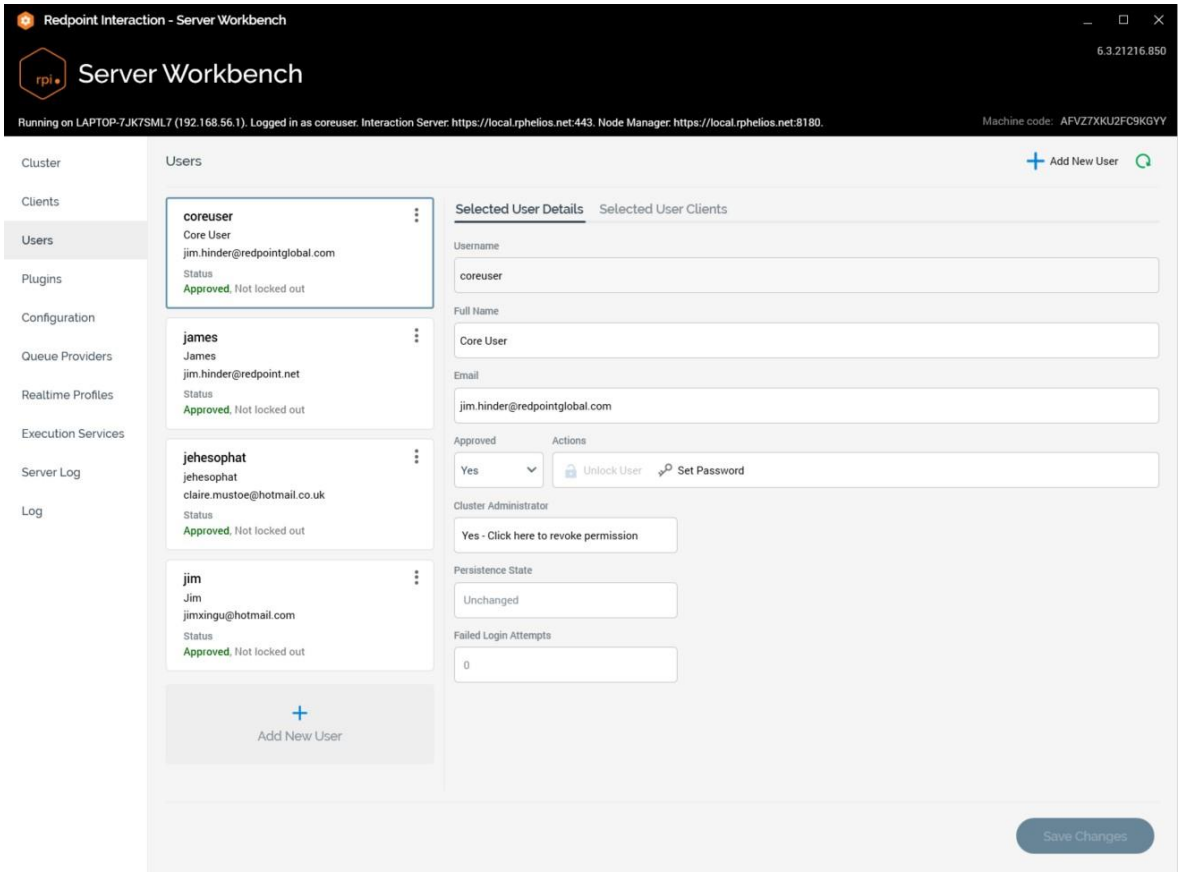
On completing the installation, the Next button, displayed at the bottom right, is replaced by a Finish button. Clicking Finish closes the Install Client overlay and displays the Server Workbench tab set, which includes all tabs. The Clients tab is displayed, the installed client's status is updated to Installed, and its connection strings are shown.

If you click Finish prior to completing the installation, you will be advised to wait until all steps are complete.

You can return to the Configure sub-interface using the Back button.

Users

Users can be created, managed and deleted using a dedicated Server Workbench tab.



A user can be associated with one or more clients (a process which can be managed in the same context).

Note that users can also be created and managed from within the RPI client application. On creation of a user in that context, an automatic link to the client within which the creating user is working is also added. On deletion of a user in the client application, the results of this action depend on the number of clients with which the user is associated. If associated with multiple clients, the user record remains, but the link to the client is deleted. If associated with a single client, the link between client and user, and the user record itself, are deleted.

The Users tab consists of the following:

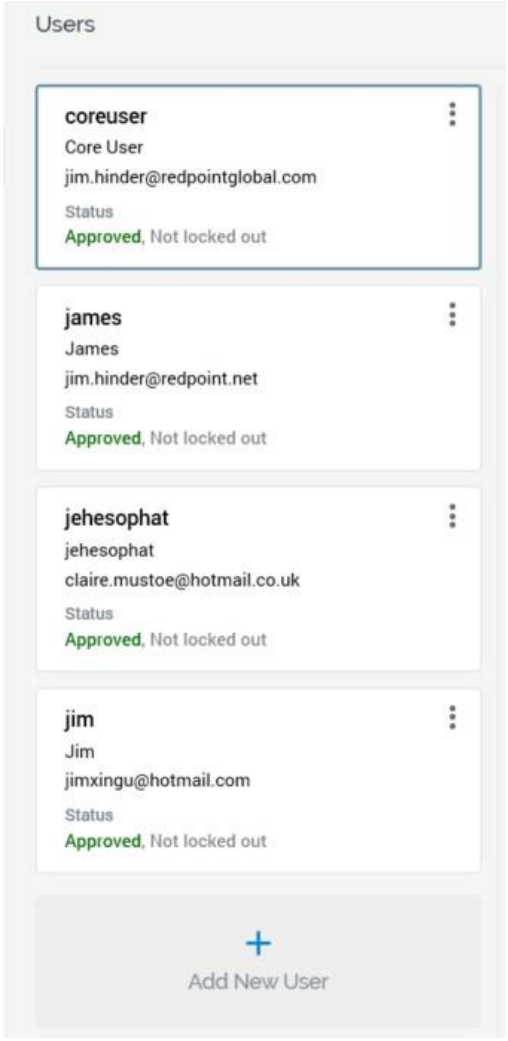
Toolbar

The Users tab toolbar exposes the following options:

- Add New User: clicking this button adds a new user to the bottom of the Users list. The new user's default Name is "username" and Full name "New User." If "username" already exists, the default name is set to "username2" (this numerical value can be incremented).
- Refresh: clicking this button loads the latest user information from the core operational database. Note that an "Are You Sure?" dialog is shown if unsaved client or user changes are present.

Users List

A list of users configured at the current cluster is displayed to the left.



When no users have yet been set up in the cluster, the list displays an advisory message (“Add and save a user, then choose clients to which he or she will have access”).

The Users list contains a list of the users that have been created in the cluster. For each user, the following properties are displayed:

- Username
- Full Name
- Email
- Status:
    - Approval status ○ Locked out status
- Actions: exposing a single option:
    - Remove: selecting this option sets the selected user’s Persistence state to

Deleted (unless its status was Added, in which case it is removed from the grid). You can no longer select a Removed user. Any Removed users are deleted permanently on invocation of Save.

Users are ordered alphabetically by Username.

On initial cluster commissioning, a default user, with the following properties, is created:

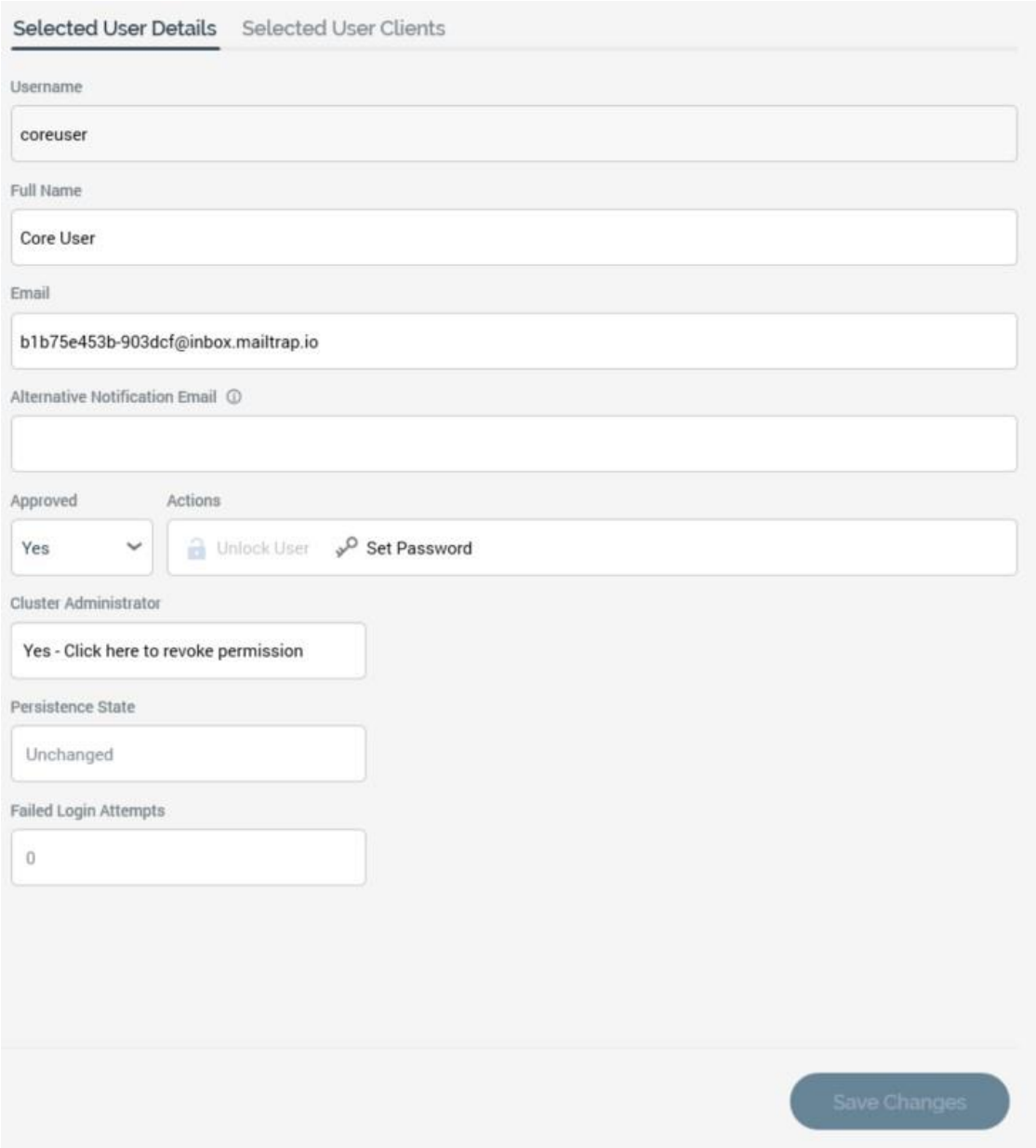
- Username: “coreuser”
- Full Name: “System administrator”
- Email: “notset@default.com”
- Password: “.Admin123”

Selected User Tabset

This section displays an expanded view and editable properties of the user selected currently in the users list. Two tabs are provided, and are described separately below.

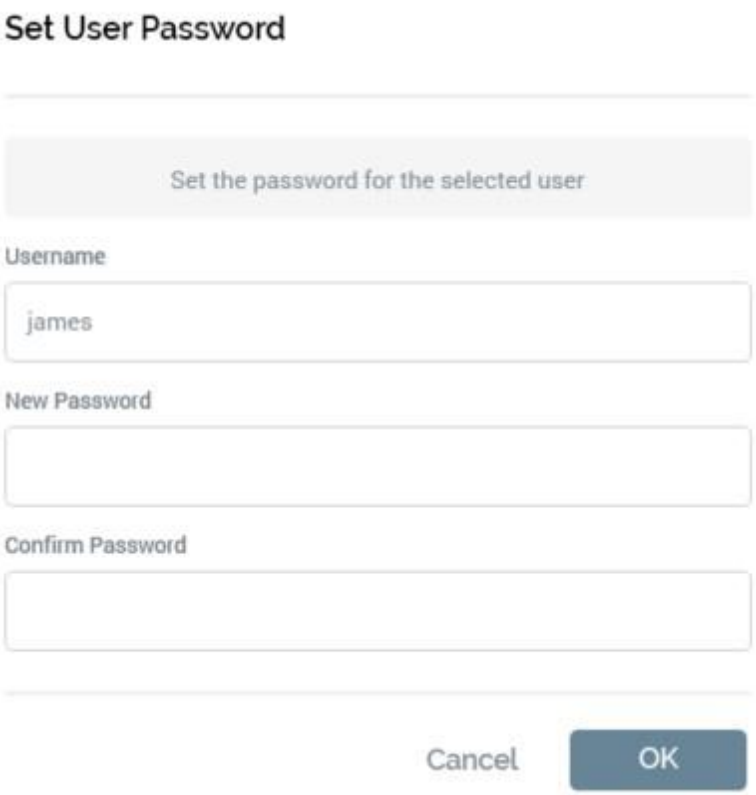
Selected User Details Tab

This tab exposes the following properties:



- Username: this property is mandatory and can be a maximum length of 256 characters. Username must be unique within the cluster and is read-only once a new user has been saved.
- Full name: this property is optional and can be a maximum length of 256 characters.
- Email: this property is mandatory and can be a maximum length of 200 characters. The value provided must be unique within the cluster and must be a valid email address.
- Alternative Notification Email: this property allows you to define a second email address, to which notification emails will be sent, rather than to the provided Email address.

- Approved: a Yes/No dropdown, set to ‘No by default when creating a new user. An unapproved user cannot log into the RPI client application.
- Actions:
    - Unlock User: this option is only available when the currently-highlight user’s account is locked. Its invocation is protected by an “Are You Sure?” dialog. Clicking the button unlocks the user account, allowing the user in question to log into the RPI client application once again. It also resets the user’s failed login attempts count to 0, and removes the Lockout ends property from display.
    - Set Password: this button is available at a previously-saved user, where no unsaved changes exist. Clicking the button displays the Set User Password dialog:

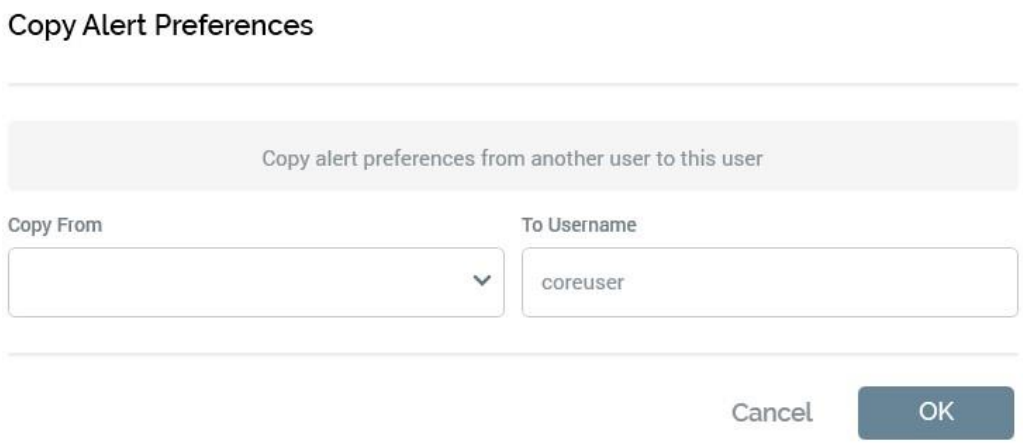


The dialog contains the following:

- Username: read-only.
- New password: it is mandatory to supply a new password. The value provided must adhere to the current password policy.
- Confirm password: also mandatory; the value supplied must match New password.

- Cancel: clicking this button removes the dialog from display without setting the user's password. Note that clicking off the dialog has the same effect.
- OK: clicking OK sets the user's password if the conditions above are met, otherwise displays a validation error.

○ Copy Alert Preferences: this button, which is available when the current user's Persistence State is Unchanged, allows you to copy alert preferences from another user to the current user. Clicking it displays the Copy Alert Preferences dialog:

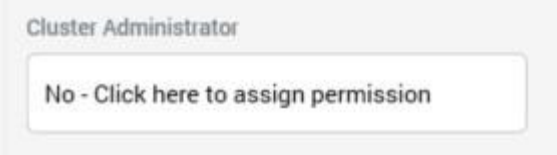


The dialog contains the following:

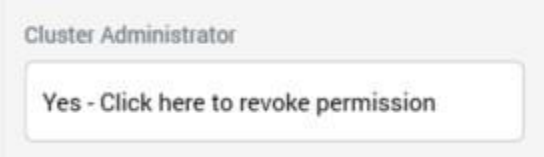
- Copy From: this dropdown property is blank by default. It lists all RPI users other than the current user. For each, the following are displayed:
  '[Username] ([Full Name])'
  It is mandatory to select a user from which to copy alert preferences.
- To Username: a read-only representation of the current user's username.
- OK: you can click OK to copy user preferences to the current user. Following a successful copy of user preferences, an advisory message is displayed at the top of the Server Workbench interface:

- Cancel: clicking this button removes the dialog from display. Clicking off the dialog has the same effect.

- Cluster administrator: only users who are defined as cluster administrators may access Server Workbench. By default, a user is not a cluster administrator, and this property appears as follows:



If you click the accompanying button, the user becomes a cluster administrator, and the property is shown thus:

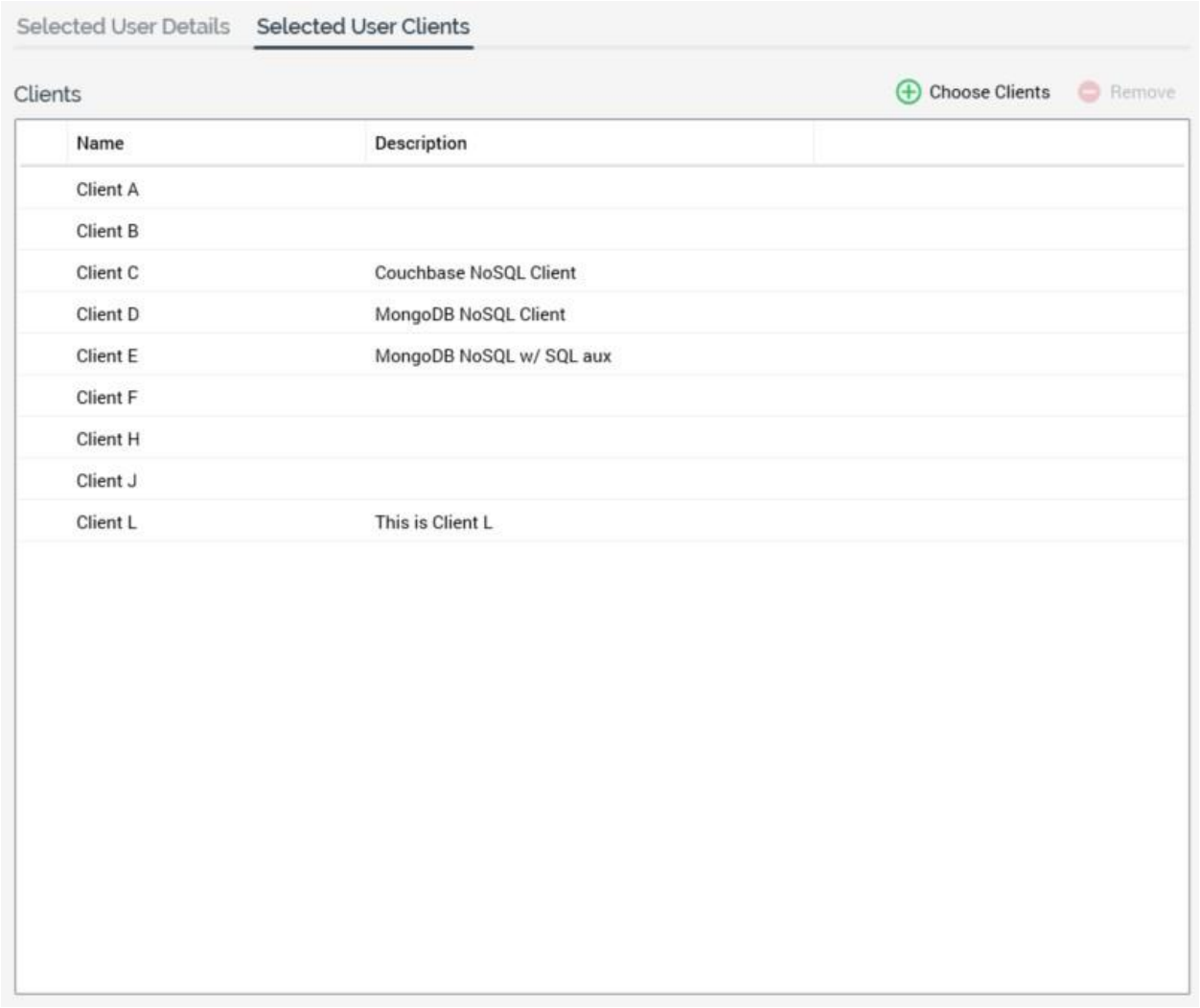


Clicking the button again revokes the cluster administrator permission.
Note that it is not necessary to save changes to this property using the Save Changes button; changes are saved automatically upon clicking the Cluster administrator button.
Note also that you cannot revoke your own cluster administrator permission.

- Persistence State: one of:
  - Unchanged
  - Added
  - Edited
  - Deleted
- Failed login attempts: this read-only property displays the current number of failed login attempts undertaken by the user. It is reset to 0 when the user's account is locked.
- Lockout ends: this read-only date/time property is shown when the user is locked out and displays the time at which the user's account will be automatically unlocked.

Selected User Clients Tab

The Selected User Clients tab exposes the following:



A toolbar is shown above the list and exposes the following buttons:

- Choose Clients: clicking this button displays the Assign Clients dialog (documented separately). The dialog allows you to manage the assignment of clients to the currently selected user.
- Remove Clients: this button is available when one or more client(s) is/are selected. Clicking it removes the selected client(s) from the Clients list.

The Clients list itself displays a read-only list of those clients associated within the currentlyselected user. Clients are displayed in alphabetical order by Name. The following client properties are shown:

- Name
- Description

Footer

A single button is displayed at the bottom of the Users tab:

The button is available when outstanding changes exist in the Clients or Users tabs. Clicking Save persists any client and user changes to the core operational database. Any displayed status icons are removed.

Password Policy

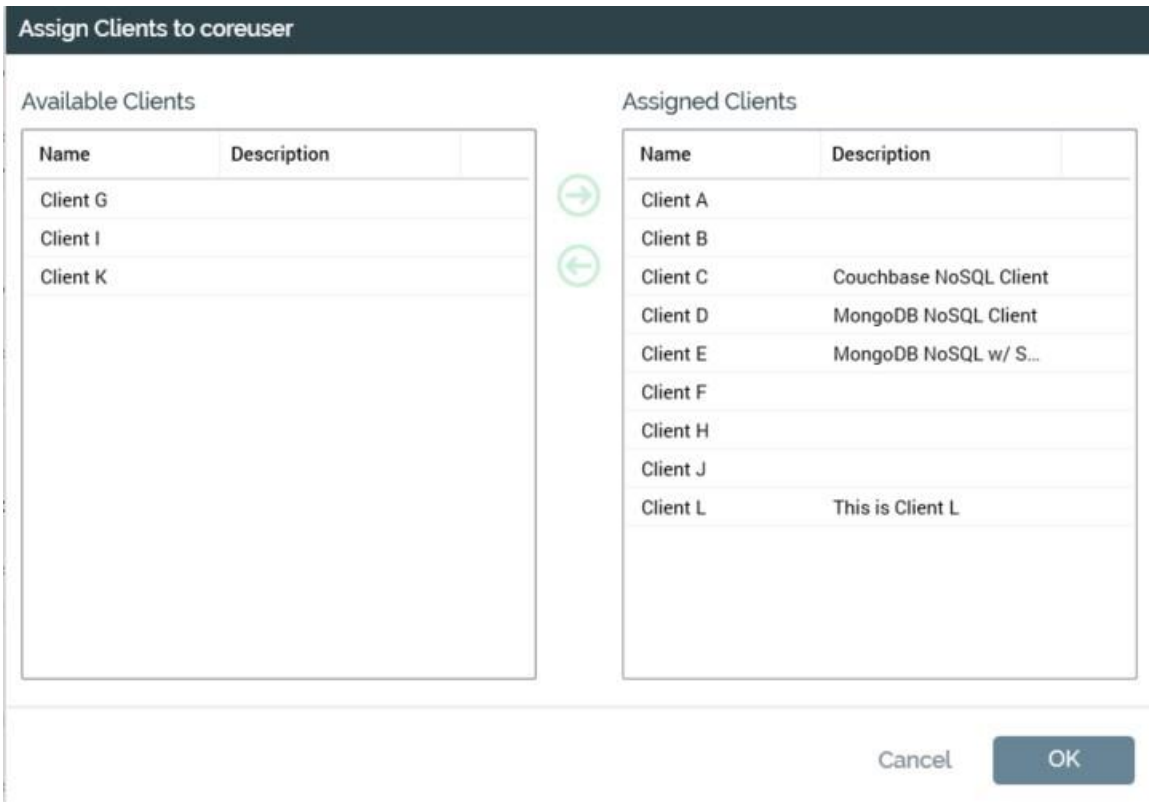
Users' passwords must conform to a password policy, which is applied to all clients in the current RPI installation.

The following cluster configuration settings are used to define the password policy:

Name	**Description**
DefaultAccountLockoutTimespan	Number of minutes a user is locked out for after MaxFailedAcessAttemptsBeforeLockout is reached
MaxFailedAttemptsBeforeLockout	Number of access attempts allowed before a user is locked out (if lockout is enabled)
PasswordRequireDigit	User passwords require a digit
PasswordRequiredLength	Minimum required length of user passwords
PasswordRequireLowercase	User passwords require a lower case letter
PasswordRequireNonLetterOrDigit	User passwords require a non-letter or digit
PasswordRequireUppercase	User passwords require a lower case letter
UserLockoutEnabledByDefault	If true, will enable user lockout when users are created

Assign Clients

The Assign Clients dialog—displayed on invocation of Choose Clients at the Clients list in the User tab—allows you to manage the assignment of clients to the currently selected user.



On invocation, the dialog presents two lists of clients: to the left, a list of those clients within the cluster that are not currently assigned to the selected user; and to the right, a list of those already assigned. On initial display, the contents of the two lists reflect the current state of client assignment at the user.

Available Clients Section

The list to the left displays clients that have not yet been associated with the current user. Clients are listed alphabetically. The following properties are shown:

- Name
- Description

Clients Buttons

Two buttons—shown between the two lists—control the assignment of clients to, and removal of clients from, the current user:

- Assign the selected Clients to this User: this button is enabled when one or more clients are selected in the Available Clients grid. Invocation moves the clients to the Assigned Clients grid.
- Remove the selected Clients from this User: this button is enabled when one or more clients are selected in the Assigned Clients grid. Invocation moves the clients to the Available Clients grid.

Note that double-clicking a client in either list has the effect of removing the client from that list and moving it to the other list.

Assigned Clients Section

The list to the right displays clients that have been associated with the current user. Clients are listed alphabetically. The following properties are shown:

- Name
- Description

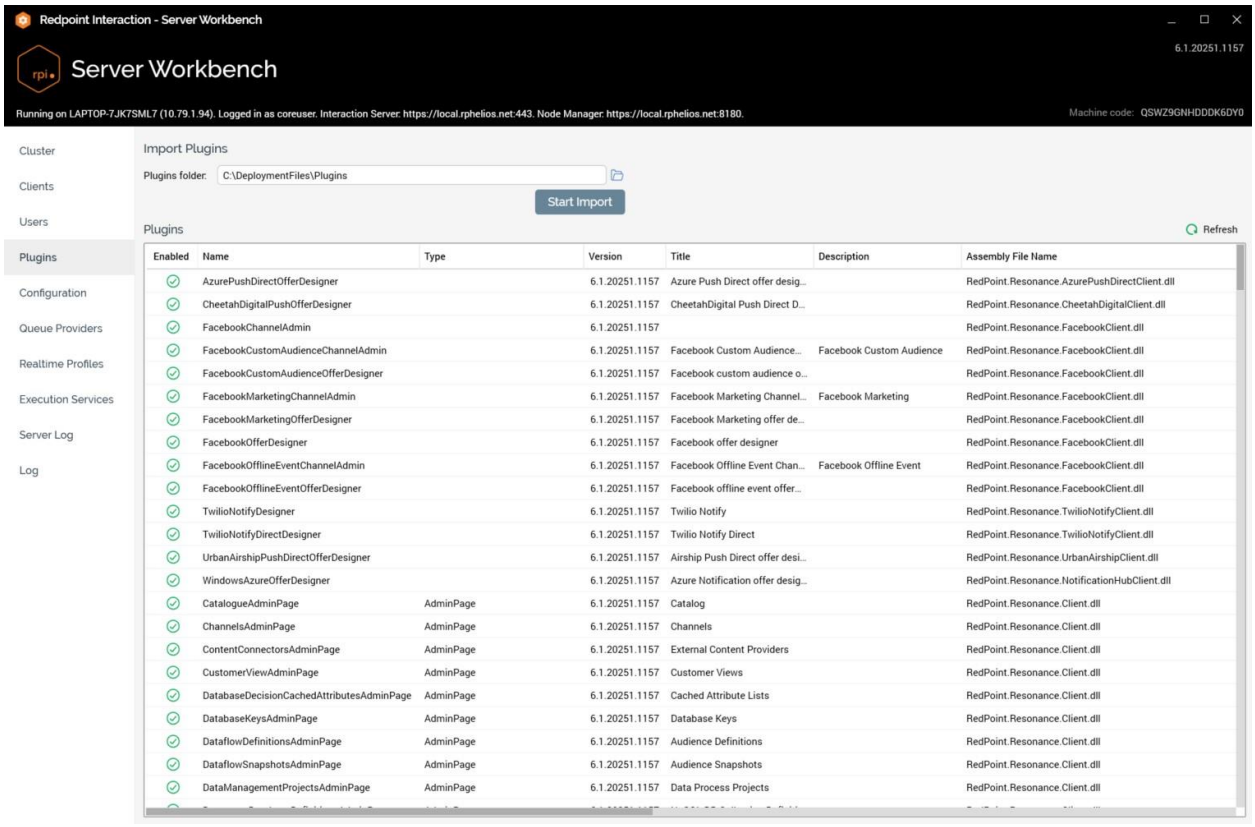
OK and Cancel

Clicking the OK button at the bottom of the dialog closes the dialog. Any changes to the user's client assignment are reflected in the Clients list. If changes were made, the user's Persistence state is set to Edited, and it is necessary to click Save to make the changes permanent.

Clicking Cancel closes the dialog without saving.

Plugins

Server Workbench includes a Plugins tab that allows you to view details of the suite of plugins loaded at the current RPI instance.



The tab also allows you to upload new plugins into the cluster. It consists of the following:

Import Plugins Section

This section allows you to upload plugins into the cluster. It contains the following:

- Plugins folder: you can specify the Windows file system folder in which the new plugins are to be found either manually or by browsing.
- Start Import: this button is enabled when a Plugins folder has been specified. Clicking it starts the import of the plugins in the selected folder and displays an “Importing” icon. Following the importation of plugins, an Import Results overlay is displayed: The overlay contains a grid with the following columns: ○ Loaded ○ Filename ○ Error Message

Note that you cannot load a plugin if its version number is lower than the version already in the cluster.

The Plugins tab can also be used to upload a new license file to apply to the RPI installation. *Plugins*

Section

This section contains a toolbar and a plugins list.

Plugins Toolbar

Clicking the Refresh button loads the latest plugin information from the core operational database.

Plugins List

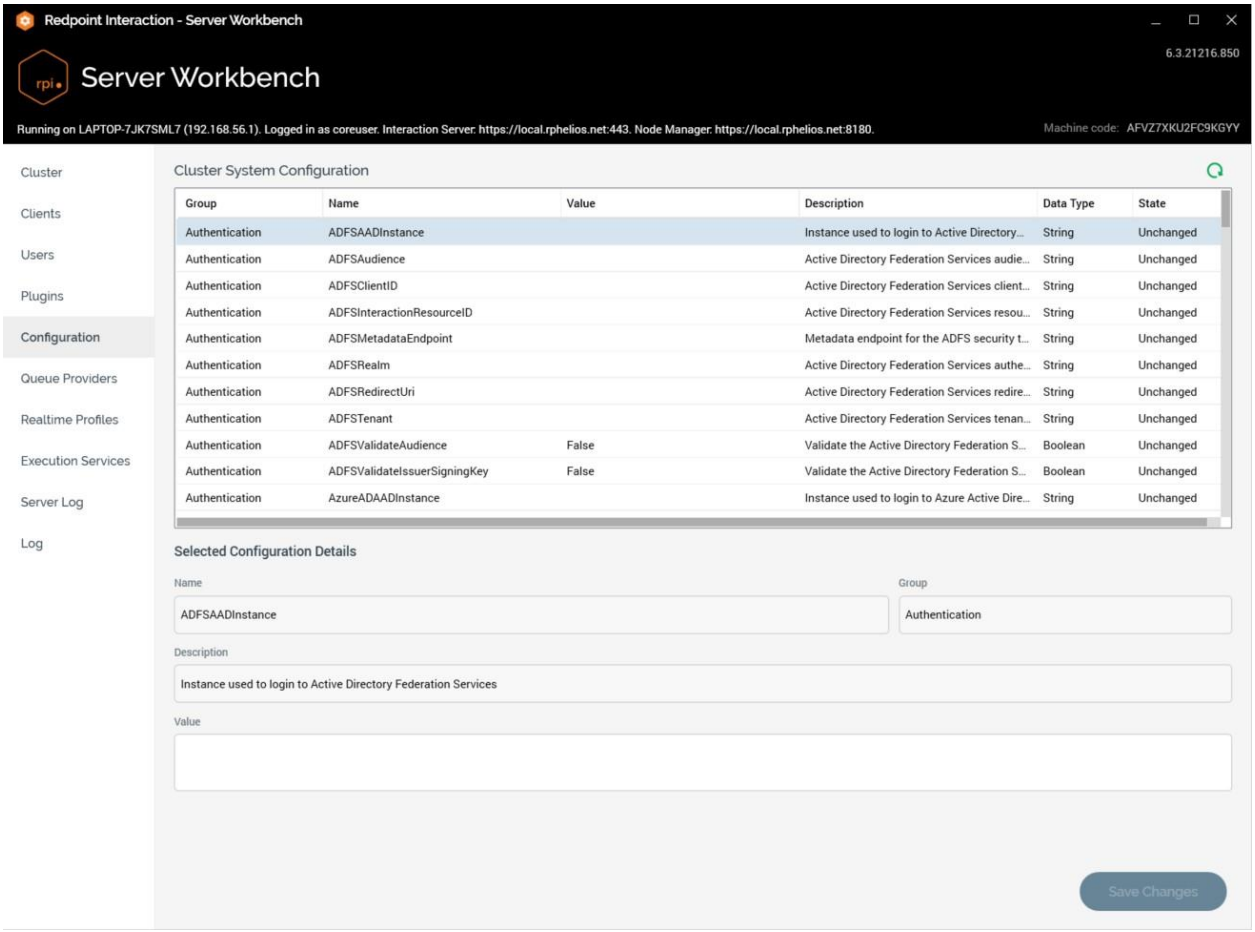
All plugins in the current RPI cluster are listed in a read-only grid. The following columns are shown:

- Enabled
- Name
- Type
- Version
- Permissions
- Title
- Description
- AssemblyFileName
- AssemblyName
- QualifiedName
- XAMLFileName
- ID

Configuration

RPI's inherent flexibility is realized by the provision of a number of configuration settings. Some of these settings are managed centrally in Server Workbench and apply to the cluster as a whole. Others are managed on a client-by-client basis in the System Configuration administration interface in the RPI client application.

Central configuration settings are managed in Server Workbench in the Configuration tab:



The tab consists of Cluster System Configuration and Selected Configuration Details sections.

Cluster System Configuration Section

This section contains a toolbar and system configuration settings list

Toolbar

Two buttons are exposed at the toolbar:

- Save: invocation of this option persists any changes made to system configuration setting values in the core operational database.

- Refresh: clicking this button re-loads the list of configuration settings and their values from the core operational database. System Configuration Settings List

The following columns are shown:

- Group: one of Environment, Personalization or Web
- Name
- Value
- Description
- Data Type
- State: one of Unchanged or Edited

The following configuration settings are displayed:

Group	Name	Default Value	Description
Authentication	AFDSAADInstance	[blank]	Instance used to login to Active Directory Federation Services
Authentication	ADFSAudience	[blank]	Active Directory Federation Services audience that will be used during token validation
Authentication	ADFSClientID	[blank]	Active Directory Federation Services client ID for the RPI native client application
Authentication	ADFSInteractionResourceID	[blank]	Azure Active Directory resource ID representing the RPI Web API
Authentication	ADFSMetadataEndpoint	[blank]	Metadata endpoint for the ADFS security token service that JWTs will be issued from
Authentication	ADFSRealm	[blank]	Active Directory Federation Services authentication realm
Authentication	ADFSRedirectUri	[blank]	Active Directory Federation Services redirect URI for the RPI Web API
Authentication	ADFSTenant	[blank]	Active Directory Federation Services tenant the tokens are issued from
Authentication	ADFSValidateAudience	[blank]	Validate the Active Directory Federation Services audience during token validation

Authentication	ADFSValidateIssuerSigningKey	[blank]	Validate the Active Directory Federation Services security key that signed the security token
Authentication	AzureADAADInstance	[blank]	Instance used to login to Azure Active Directory
Authentication	AzureADAudience	[blank]	Azure Active Directory audience that will be used during token validation

Authentication	AzureADClientID	[blank]	Azure Active Directory client ID for the RPI native client application
Authentication	AzureADInteractionResourceID	[blank]	Azure Active Directory resource ID representing the RPI Web API
Authentication	AzureADRedirectUri	[blank]	Azure Active Directory redirect URI for the RPI Web API
Authentication	AzureADTenant	[blank]	Azure Active Directory tenant the tokens are issued from
Authentication	EnableADFS	False	This server supports Active Directory Federation Services authentication
Authentication	EnableAzureAD	False	This server supports Azure Active Directory authentication
Authentication	EnableOpenIdAuthentication	False	This server supports OpenID authentication
Authentication	OpenIdAudience	[blank]	OpenId application audience identifier

Authentication	OpenIdAuthorizationHost	[blank]	OpenId authorization host address
Authentication	OpenIdClientID	[blank]	OpenId client ID
Authentication	OpenIdCustomScopes	[none provided]	List of custom scopes to request for the OpenId access token
Authentication	OpenIdEnableRefreshTokens	False	If checked, refresh tokens will be requested to renew access tokens
Authentication	OpenIdRedirectURL	[blank]	OpenId redirect address
Authentication	OpenIdLogoutIdTokenParamete r	id_token_hint	Query parameter name used to pass the id token on logout
Authentication	OpenIdUsePKCE	True	If checked, the Proof Key for Code Exchange authorization flow is used
Authentication	OpenIdValidateAudience	True	If checked, the OpenIdAudience must match the audience name supplied in the access

			token
Connectors	DotMailerExternalDynamicCont entSharedFolderPath	C:\temp\Redpoint\dotdigital	DotMailer shared external dynamic content output folder path
Connectors	DynUnsubscribeTempFolderPat h	C:\temp\Redpoint\DynUnsubs cribe	Dyn unsubscribe temporary output folder path
Connectors	FacebookDataFileFolderPath	C:\temp\Redpoint\Facebook Offline Event\	Facebook data file folder path
Connectors	GoogleAdsCustomerMatchData FileFolderPath	C:\temp\RedPoint\Google Ads Customer Match\	Google Ads Customer Match data file folder path

Connectors	InstillerProcessEventFolderPath Format	C:\temp\Redpoint\Instiller\{0}\ Processed	Instiller process data file folder path format
Connectors	MessenteDataFileFolderPath	C:\temp\Redpoint\Messente\	Messente data file folder path
Connectors	MobileConnectDataFileFolderP ath	C:\temp\Redpoint\MobileConn ect\	MobileConnect data file folder path
Connectors	OutboundDeliveryEventFolderP ath	C:\temp\Redpoint\OutboundD eliveryCallbackService	Outbound Delivery event delivery output folder path
Connectors	SendGridEmailMetricsFolderPat h	C:\temp\Redpoint\SendGridCa llbackService	SendGrid email metrics output folder path
Connectors	SendGridWebAPIExternalConte ntProviderSettings		Content provider settings json
Connectors	SilverpopDataFileFolderPath	C:\temp\Redpoint\IBM Marketing Cloud\	Silverpop data file folder path
Connectors	SparkPostSaveEventFolderPath Format	C:\temp\Redpoint\SparkPost\{ 0:N}\Processed\{1:N}\{2:yyyyMM-ddThh_mm_ss}	SparkPost Post data file folder path format
Connectors	UrbanAirshipDataFileFolderPath	C:\temp\Redpoint\UrbanAirshi p	UrbanAirship data file folder path
Database	OracleTNSAdminRootFolderPat h		The folder location of Oracle TNS admin configuration files (tnsnames.ora and sqlnet.ora)
Database	OracleTraceFileLocation	C:\temp\RedPoint Interaction\Oracle\Traces	The folder location where the generated trace files will be written
Database	OracleTraceLevel	0	0 = Tracing is disabled, 1 = PublicAPI, 2 = PrivateAPI, 4 = NetworkAPI/Data, 7 = All traces enabled
Database	OracleTraceOption	0	0 = Generates single trace file, 1 = Generates multiple trace files
Environment	AccessTokenExpiryMinutes	20	Number of minutes access tokens remain valid after being issued

Environment	AllowInsecureHttp	False	Controls whether the server allows insecure HTTP requests
Environment	AllowSavingLoginDetails	True	When a user successfully logs into this server, allow the servername and credentials to be saved
Environment	AlwaysShowClientsAtLogin	False	Force the Choose Clients dialog to appear at login even if the user has access to only one client
Environment	ApplicationSupportURL	https://support.Redpointglobal.com/	Web page URL that points to the application support page
Environment	AuditTaskEvents	False	Controls if system task stop and start events should be audited
Environment	AzureADAADInstance	[blank]	Instance used to login to Azure Active Directory
Environment	AzureADAudience	[blank]	Azure Active Directory Audience that will be used during token validation
Environment	AzureADClientID	[blank]	Azure Active Directory client ID for the RPI native client application
Environment	AzureADInteractionResourceID	[blank]	Azure Active Directory resource ID representing the RPI Web API
Environment	AzureADRedirectUri	[blank]	Azure Active Directory redirect URL for the RPI Web API
Environment	AzureADTenant	[blank]	Azure Active Directory tenant the tokens are issued from

Environment	DefaultAccountLockoutTimesp an	5	Number of minutes a user is locked out for after MaxFailedAcessAttempts BeforeLockout is reached

Environment	DeploymentFileShare	\\[server]\RPI Deployment	Path of the deployment files share
Environment	EnableADFS	False	This server supports Active Directory Federation Services authentication
Environment	EnableAzureAD	False	This server supports Azure Active Directory authentication
Environment	EncryptionKey	C:\Resonance123	System wide encryption key
Environment	GlobalFileOutputAllowAllServer Locations	True	If checked, clients are allowed to specify any server location for data extract and export activities
Environment	GlobalFileOutputAllowClientOve rrides	True	If checked, clients can configure and use the local FileOutput configuration settings
Environment	GlobalFileOutputDirectory	C:\RPI File Output	Folder used to store any file assets exported via interactions or selection rules
Environment	HelpStartPageURL	\\[server]\RedpointInteraction Help/index.html	URL of the help start page
Environment	HelpTrainingStartPageURL	http://Redpoint-cbts.Redpointglobal.com/Redpoint-Interaction-CBT-Links.html	URL of the training materials start page

Environment	ListenerQueueIsEnabled	False	If checked, the system will monitor the listener queue and execute any packages found
Environment	ListenerQueueMaxBatchSize	50	The maximum number of listener queue packages processed by a single execution thread
Environment	ListenerQueuePath	[blank]	The listener queue path
Environment	ListenerQueueThreadPoolSize	10	The number of listener queue packages that can be processed concurrently
Environment	ListenerQueueTimeoutMinutes	60	The maximum number of minutes to allow for processing a listener

			queue package
Environment	ListenerQueueUseMessageLoc ks	True	If set and supported by the queue provider, messages remain locked on the queue until processed
Environment	MapControlConfiguration	Bing	Map control API configuration
Environment	MaxFailedAttemptsBeforeLock out	5	Number of access attempts allowed before a user is locked out (if lockout is enabled)
Environment	OperationalDatabaseType	SQLServer	The type of database server on which the cluster database is to be created
Environment	PasswordRequireDigit	True	User passwords require a digit
Environment	PasswordRequiredLength	6	Minimum required length of user passwords

Environment	PasswordRequireLowercase	True	User passwords require a lower case letter
Environment	PasswordRequireNonLetterOrDi git	True	User passwords require a non-letter or digit
Environment	PasswordRequireUppercase	True	User passwords require a lower case letter
Environment	ServerDateTimeSource	1	0 = Server OS, 1 = Operational Database Instance
Environment	ServiceHostName	[not set]	Host name client connects to
Environment	ServiceHostPortNumber	443	Port number client connects on the service host
Environment	SMTPCredentials	[not set]	The credentials used to sign into the SMTP server
Environment	SMTPEnableSSL	False	Check if an SSL connection is required to connect to the SMTP server
Environment	SMTPPortNumber	25	Port number used to connect to the SMTP server

Environment	SMTPServerName	10.100.1.10	SMTP server to be used by the application for emailing Environment
Environment	SMTPUseCredentials	False	Controls if authentication credentials are passed through to the SMTP server
Environment	TaskTimeout	60	Number of minutes a task can execute before it is cancelled. Applies to queue tasks only
Environment	TriggerCheckCriteriaInterval	60	Number of seconds interval between checking if trigger constraints have been met

Environment	TriggersMaxDaysInactive	180	Maximum number of days a trigger can remain in an activated state without being fired
Environment	UserLockoutEnabledByDefault	True	If true, will enable user lockout when users are created
Environment	WorkflowPauseAllTimeout	10	The timeout limit in minutes that the workflow manager will wait for all workflows to pause
Integration API	APIAccessTokenExpiryMinutes	60	Number of minutes access tokens remain valid after being issued
Integration API	ApiMaxSearchResultCount	256	Number of managed asset search results returned per call
Integration API	ApiSearchClientNames	[String Values]	List of RPI client names on which to apply ECP managed asset indexing
Integration API	ApiSearchIndexingEnabled	True	If checked, the Integration API ECP managed asset indexing service will be enabled
Integration API	ApiSearchIndexingInterval	60	The interval in minutes between ECP managed asset indexing jobs
Integration API	ApiZendeskToken	12CAA197-ED6C-4DCC-B920-F4CC769F2B4C	Zendesk endpoint authorization token
Personalization	EmailSenderAddress	admin@Redpointglobal.com	System email sender address
Personalization	PasswordResetSubject	Resonance Password Reset	Email subject to be used for password reset email
Web	GoalDrivenAssetWebServiceAddress		The Redpoint machine learning web service address

Note that setting AccessTokenExpiryMinutes requires an IIS restart for any changes to be applied.

Selected Configuration Details Section

This section displays the full details of the configuration setting selected currently in the list. The following read-only properties are shown:

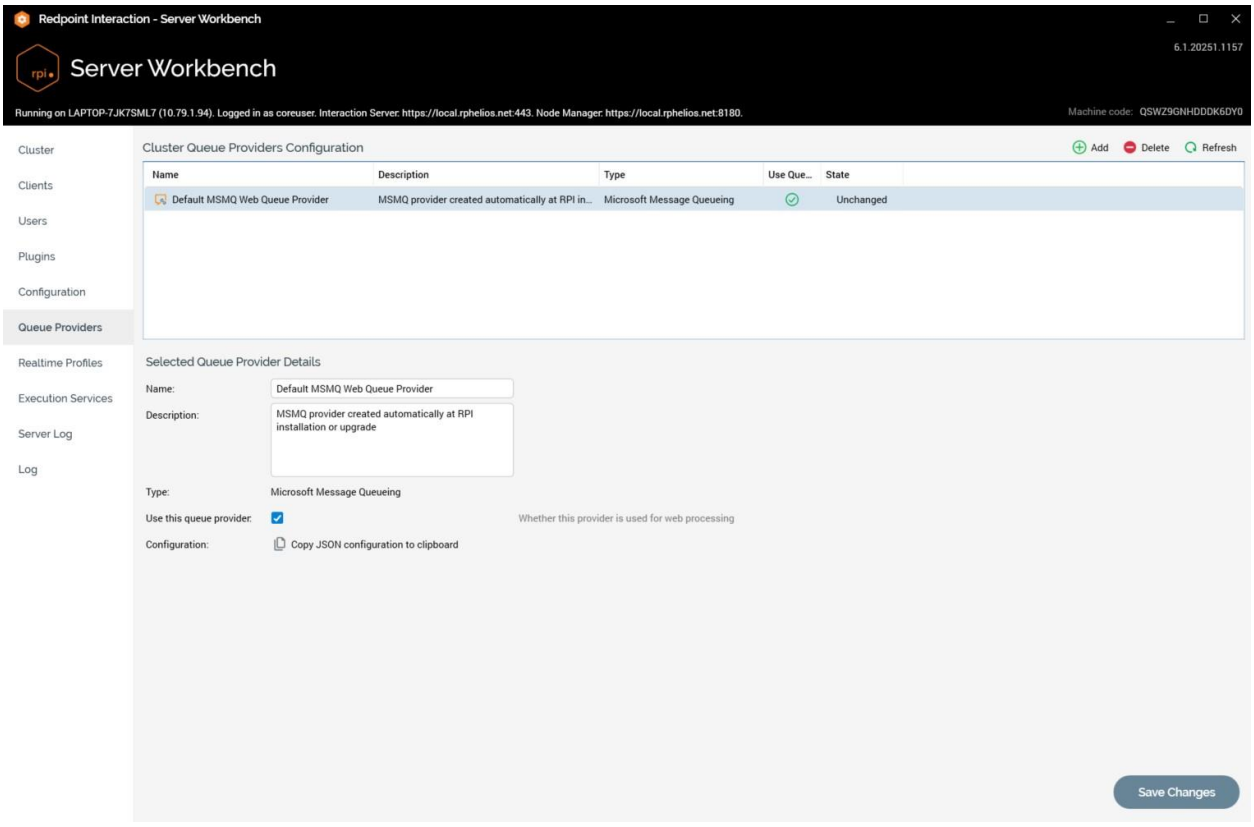
- Name
- Group
- Description

In addition, a single writable property is displayed:

- Value

Queue Providers

The Server Workbench Queue Providers section allows you to configure queue providers at the cluster level.



Typically, such queue providers will be used to host the listener queue, which supports usage of interaction queue listeners and activities.

Further details on queue providers, listeners and activities can be found in the Redpoint Interaction User Guide.

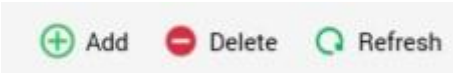
The Queue Providers tab consists of Cluster Queue Providers Configuration and Selected Queue Provider Details sections. Each is documented separately.

Cluster Queue Providers Configuration Section

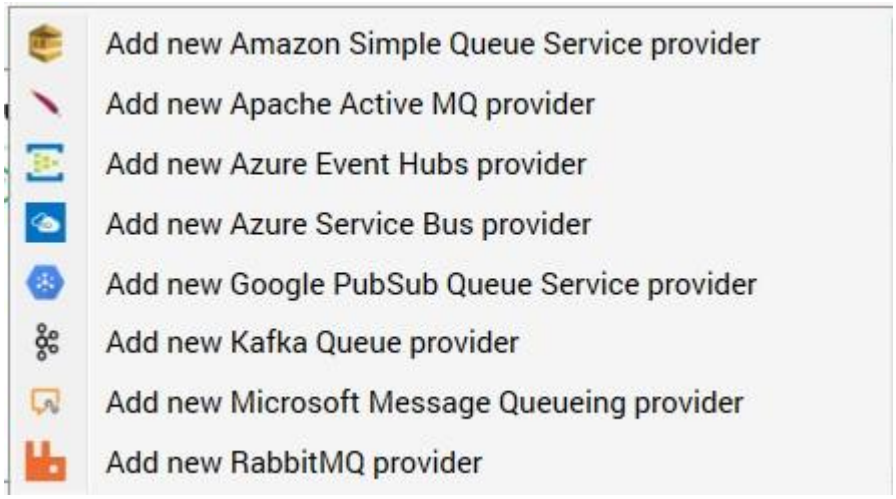
This section consists of a toolbar and a cluster queue providers grid.

Toolbar

The Cluster Queue Providers Configuration toolbar exposes the following:



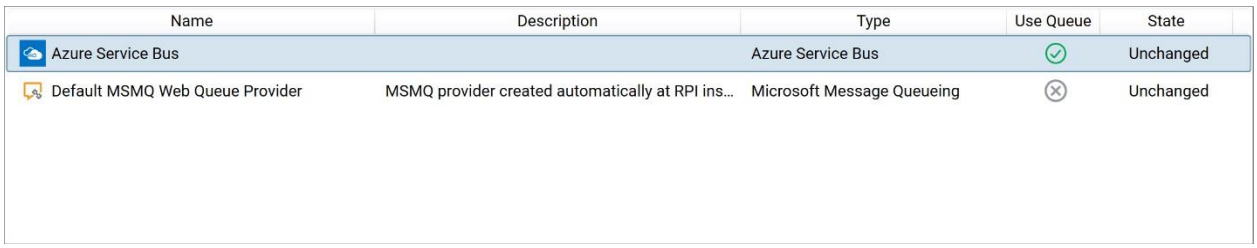
- Add: clicking this button displays a sub-menu, which allows you to select the type of queue provider you wish to add:



- Delete: clicking this button initiates the deletion of the selected queue provider(s). If the provider(s) have been saved previously, they are marked as Deleted, and may no longer be selected. If the provider(s) are yet to have been saved, they are removed from display.
- Refresh: clicking this button reloads the cluster queue providers.

Cluster Queue Providers Grid

The Cluster Queue Providers Configuration grid contains the following read-only columns:

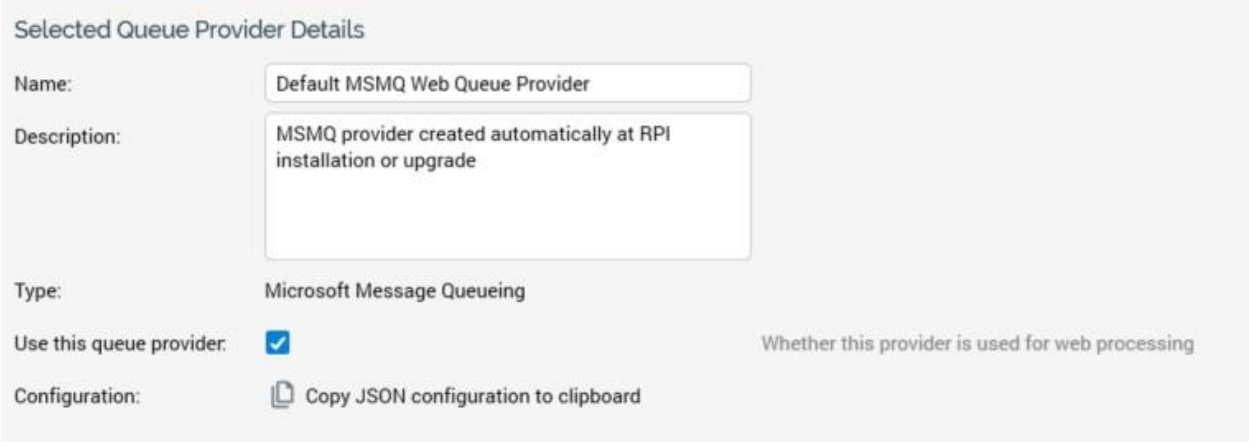


- [Icon]
- Name
- Description
- Type
- Use Queue: displaying a tick or cross
- State

The Default MSMQ Queue Provider, which is created when the Redpoint Interaction server is installed, is displayed by default in the grid.

Selected Queue Provider Details Section

This section exposes the following properties:



- Name: each queue provider must be provided with a unique name, with a maximum length of 100 characters.
- Description: each queue provider can be provided with a description, which can be a maximum length of 1000 characters.
- Type: read-only
- Use this queue provider: this checkbox is unchecked by default. When checked, the provider will be used. Only one provider can be checked.
- Configuration: this property is accompanied by a button. When unsaved queue provider changes are present, the button is disabled, and appears as follows:

  When no unsaved changes exist at queue providers, the button is enabled, and appears as follows:

  Clicking the button copies the queue provider's JSON configuration to the clipboard. This can be used to configure the RPI Realtime configuration file.

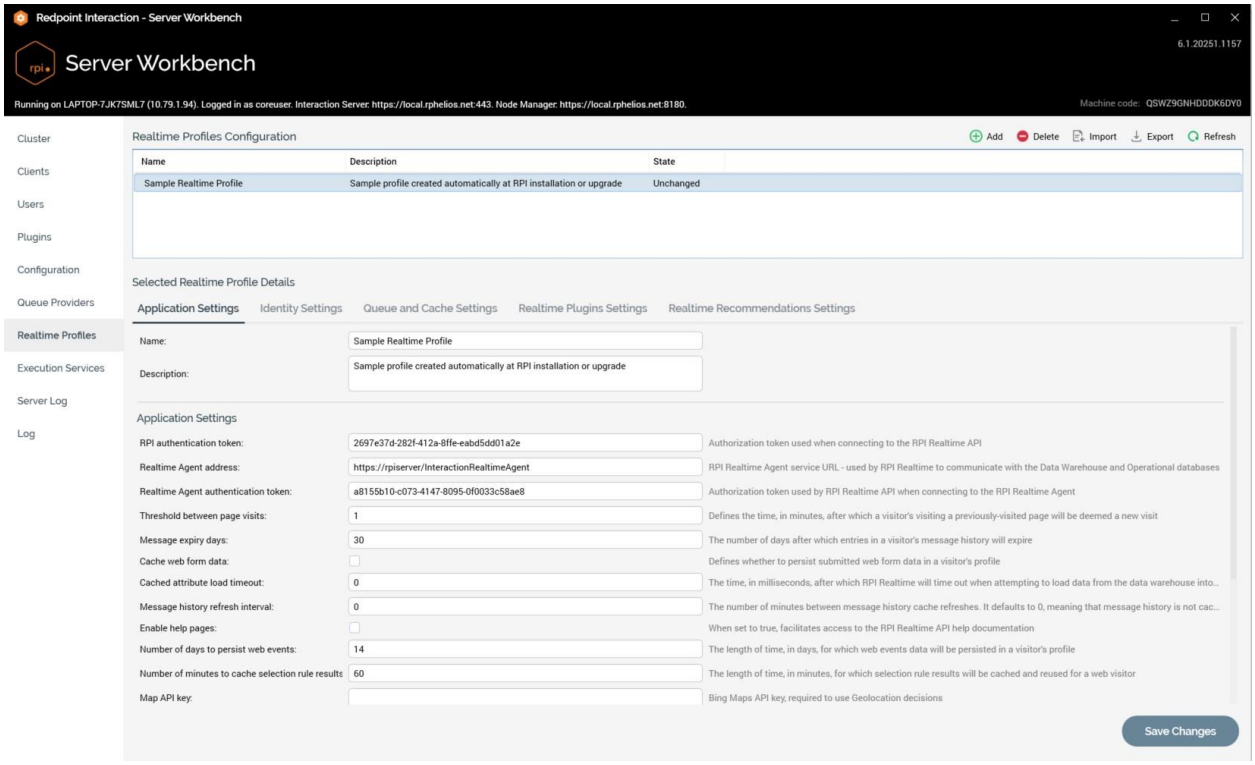
For more information, including details on properties specific to queue provider types, please see the RPI Reference Guide.

Please also see the Active Directory Setup section elsewhere in the Admin Guide.

You can save any changes made to Queue Providers using the button at the bottom of the interface.

Realtime Profiles

This tab allows you to create, edit and export Realtime Profiles. A realtime profile represents the starting point for configuring RPI Realtime. On adding a realtime profile, you can set its properties in accordance with your anticipated RPI Realtime environment. You can then save the profile and export it. Doing so generates an appSettings.json.config file, which can be installed at the RPI Realtime website.
The tab contains the following:



- Realtime Profiles Configuration list
- Selected Realtime Profile Details Each of these is discussed below.

Realtime Profiles Configuration List

The Realtime Profiles Configuration list consists of a toolbar and list. Toolbar

The following buttons appear within the Realtime Profiles Configuration list:

- Add new Realtime Profile: clicking this button adds a new realtime profile to the list. Its default name is 'New Realtime Provider' (uniqueness is ensured through the addition of an incrementing integer).

- Delete selected Realtime Profiles: this button is enabled when one or more realtime profiles are selected in the list. Invocation deletes the realtime profiles without displaying an 'Are You Sure?' dialog.
- Import: this button allows you to import a realtime profile from a file. Clicking it displays the Import Realtime Profile Windows file system dialog, in which the File type defaults to 'Config Files (*.config)'. You can select a realtime profile file and click Open to proceed the with import. An imported profile's name is 'New Imported Realtime Profile' (an incrementable integer can be appended to ensure uniqueness). The new realtime profile's settings are as those saved in the imported file.
- Export: clicking this button allows you to export the currently-selected realtime profile. It is enabled when the current realtime profile's state is Unchanged. Invocation displays the Save Realtime Profile As Windows file system dialog. By default, the Documents folder is selected, the file's name is 'appsettings', and file type 'JSON'. Having saved the realtime profile, an information message is displayed. Following export, a realtime profile file is generated, and settings from the realtime profile are reflected at the same.
- Refresh: clicking this button reloads details of realtime profiles from the operational database.

In addition, you can save any changes you make to realtime profiles using the Save Changes button shown at the bottom right of the interface.

List

All current realtime profiles are listed in alphabetical order.

For each, the following read-only information is shown:

- Name
- Description
- State

A single realtime profile is created by default. Its name is 'Sample Realtime Profile', and description 'Sample profile created automatically at RPI installation or upgrade'.

Selected Realtime Profile Details

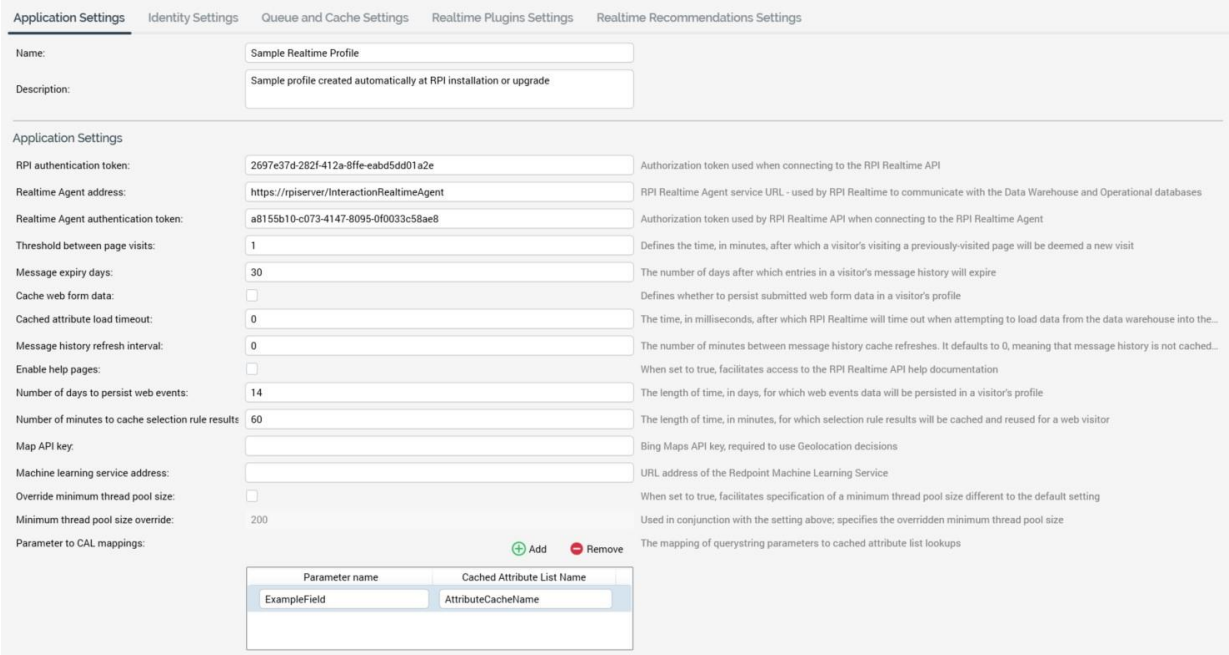
The Selected Realtime Profile Details section, shown below the Realtime Profiles Configuration list, contains a tabset, which contains five tabs:

- Application Settings
- Identity Settings
- Queue and Cache Settings
- Realtime Plugins Settings
- Realtime Recommendations Settings

Each is discussed separately below.

Application Settings

This tab is used to manage the following realtime profile settings:



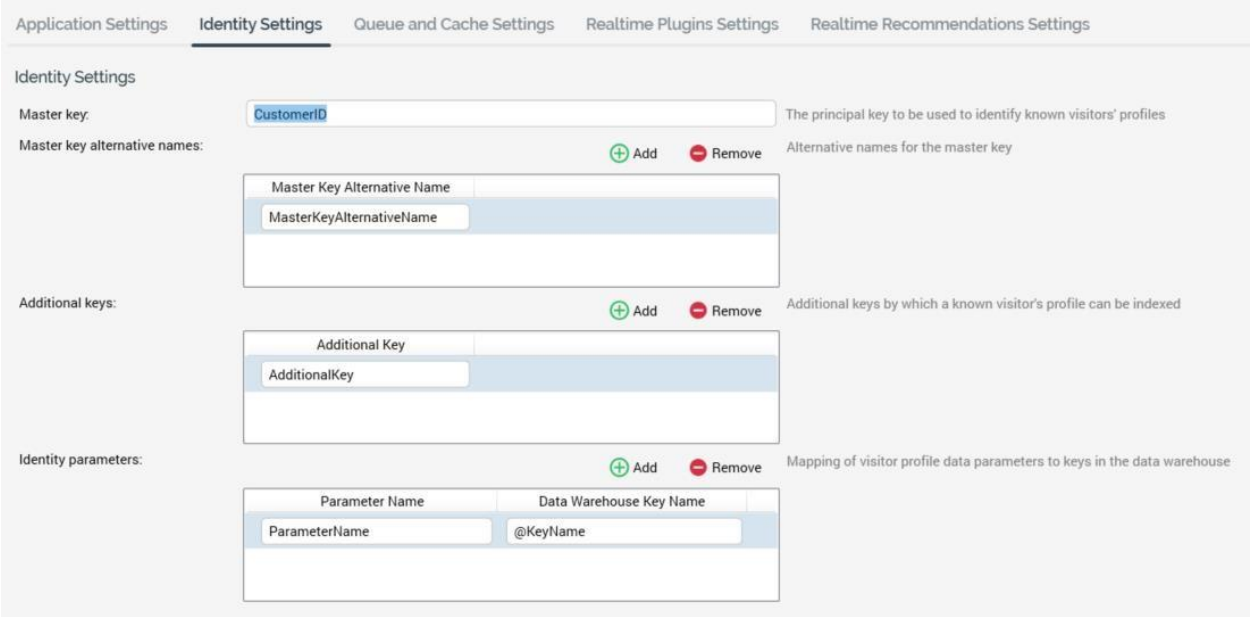
- Name: this mandatory string property can be a maximum of 100 characters in length. The value provided must be unique across realtime profiles.
- Description: this optional string property can be a maximum of 1000 characters in length.
- RPI authentication token: this mandatory string (GUID) property represents the authorization token to be used when connecting to the RPI Realtime API. An example GUID is provided.
- Realtime Agent address: this mandatory string property represents the address of the RPI Realtime Agent service, which is used by RPI Realtime to communicate with the RPI data warehouse and operational database. It defaults to the value 'https://rpiserver/InteractionRealtimeAgent'.
- Realtime Agent authentication token: this mandatory string (GUID) property represents the authorization token to be used by the RPI Realtime API when connecting to the Realtime Agent. An example GUID is provided.
- Threshold between page visits: this mandatory integer property defines the time, in minutes, after which a visitor's visiting a previously-visited web page will be deemed a new visit. It defaults to the value 1.
- Message expiry days: this mandatory integer property represents the number of days after which entries in a visitor's message history will expire. It defaults to the value 30.
- Cache web form data: this checkbox allows you to specify whether to persist submitted web form data in a visitor's profile. It is unchecked by default.

- Cached attribute load timeout: this mandatory integer property represents the time, in milliseconds, after which RPI Realtime will time out when attempting to load data from the data warehouse. It defaults to the value 0.
- Message history refresh interval: this mandatory integer property represents the number of minutes between message history cache refreshes. It defaults to 0, meaning that message history is not cached. If set to a value greater than 1, message history is cached.
- Enable help pages: this checkbox, which is unchecked by default, allows you to provide access to the RPI Realtime API help documentation.
- Number of days to persist web events: this mandatory integer property represents the length of time, in days, for which web events data will be persisted in a visitor's profile. It defaults to the value 14.
- Number of minutes to cache selection rule results: this mandatory integer property represents the length of time, in minutes, for which selection rule results will be cached and reused for a web visitor. It defaults to the value 60.
- Map API key: this optional string property (blank by default) allows you to specify a Bing Maps API key, which is required to use Geolocation decisions.
- Machine learning service address: this optional string property (blank by default) allows you to specify the URL address of the Redpoint Machine Learning Service.
- Override minimum thread pool size: this checkbox, which is unchecked by default, facilitates specification of a minimum thread pool size different to the default setting.
- Minimum thread pool size override: this integer property is enabled and mandatory when Override minimum thread pool size is checked. It allows you to specify the overridden minimum thread pool size. It defaults to the value 200
- Parameter to CAL mappings: this property consists of toolbar and table.
    - Toolbar: exposing the following options:
        - Add New Item: clicking this button adds a new row to the table. By default, its Parameter Name is 'Example Field', and its Cached Attribute List Name 'AttributeCacheName'.
        - Remove Items: this button is enabled when one or more rows are selected in the table. Invocation removes the row(s) without display of an 'Are You Sure?' dialog.
    - Table, containing the following columns:
        - Parameter Name: this mandatory property must be unique within the table.
        - Cache Attribute List Name: mandatory

      Note that provision of rows within the table is optional.

Identity Settings

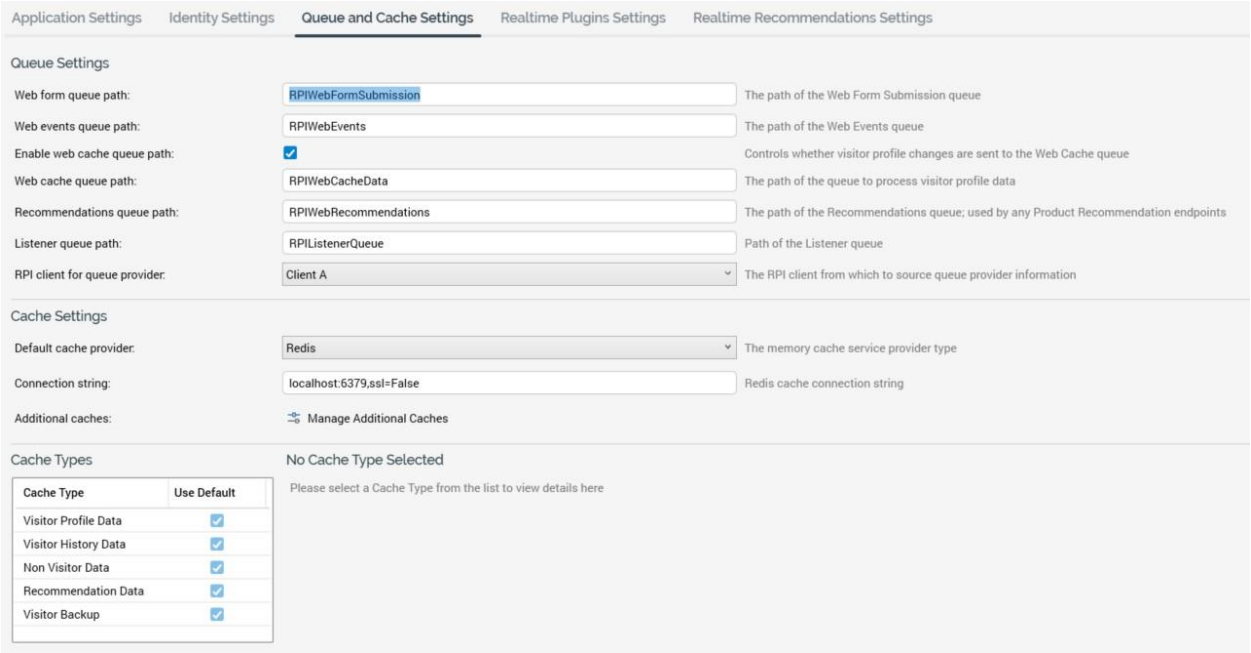
This tab is used to manage the following realtime profile settings:



- Master key: this optional string property represents the principal key to be used to identify known visitors' profiles.
- Master key alternative names: this table allows you to specify alternative names for the master key. You can add or delete values as required.
- Additional keys: this table allows you to specify additional keys by which a known visitor's profile can be indexed. You can add or delete values as required.
- Identity parameters: this table allows you to define the mapping of visitor profile data parameters to keys in the data warehouse. You can add or delete values as required.

Queue and Cache Settings

This tab is used to manage the following realtime profile settings:



- Queue Settings section: exposing the following properties: ○ Web form queue path: this optional string property allows you to specify the path of the Web Form Submission queue.
  - Web events queue path: this optional string property allows you to specify the path of the Web Events queue.
  - Enable web cache queue path: this checkbox, which is unchecked by default, allows you to specify whether visitor profile changes are sent to the Web Cache queue.
  - Web cache queue path: this string property allows you to specify the path of the queue to process visitor profile data. It is enabled, and mandatory, when Enable web cache queue path is checked.
  - Recommendations queue path: this optional string property allows you to specify the path of the Recommendations queue which is used by any Product Recommendation endpoints.
  - Listener queue path: this optional string property allows you to specify the path of the Listener queue.
  - RPI client for queue provider: this dropdown property lists all installed RPI clients in the current cluster. It allows you to select the RPI client from which to source queue provider information. Selection of a client is mandatory.

- Cache Settings section, containing the following properties:
  - Default cache provider: this dropdown lists all supported memory cache provider types. Selection of a value therein enables additional properties, display of which is contingent on the value chosen.

    If Redis:

- Connection string If Azure Redis: Connection string

If SAP Hana:

- Server name
- Schema
- User name
- Password

If Memcached:

- Section name: the Web.config section containing the Memcached configuration

If Cassandra:

- Keyspace
- Contact points
- Username
- Password

If Couchbase:

- Server name
- Username
- Password
- Bucket Name If MongoDB:

- Server name
- Port
- Schema
- Collection name

If NCache:

- Cache ID
- Server List If CosmosDB:

- Database ID Connection String

If MemoryDB Redis:

- Host
- Port
- SSL

- Username
- Password

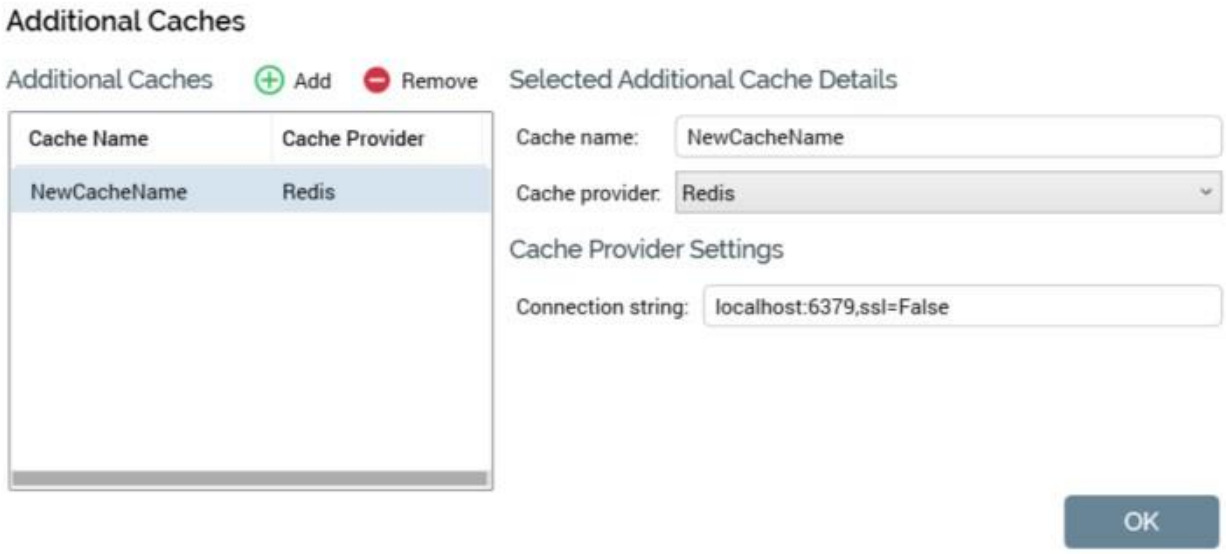
All cache provider-specific properties are mandatory strings.

- Additional Caches: accompanied by a button:
  - Manage Additional Caches: clicking this button displays the Additional Caches overlay (see below).

- Cache Types list: this read-only list contains four rows. The following columns are displayed:
  - Cache Type: this read-only column contains the following values:
    - Visitor Profile Data
    - Visitor History Data
    - Non Visitor Data
    - Recommendation Data
    - Visitor Backup ○ Use Default: a writeable checkbox.

- Selected Cache Type section: this section displays details of the cache type selected to the left. When a cache type is not selected, a message advises to select one to view its details. The section exposes the following properties: ○ Cache Type: read-only.
  - Use default cache: this checkbox is checked by default. When unchecked, the Cache name property is displayed.
  - Cache name: this property is displayed when Use default cache is unchecked. A dropdown field, it lists all configured additional caches. It is mandatory if shown.
  - Days to persist: this mandatory integer property accepts a maximum value of 999,999,999. ○ Compress data: this checkbox is only shown when one of Visitor Profile or History Data is selected. It is checked by default.

The Additional Caches overlay contains the following:



- Additional Caches list: consisting of a toolbar and a list.
  - Toolbar: exposing the following options:
    - Add Additional Cache: clicking this button adds an additional cache to the list, with defaults as follows:
- Cache Name: 'NewCacheName' (an incrementable integer can be added to ensure uniqueness).
- Cache Provider = 'Redis'.
    - Remove Additional Cache: clicking this button removes the selected additional cache from the list without displaying an 'Are You Sure?' dialog.
  - List: containing the following read-only columns:
    - Cache Name
    - Cache Provider

    - Selected Additional Cache Details section: this section displays details of the additional cache selected to the left. When a cache is not selected, a message advises to select one. The section exposes the following properties:
  - Cache name: this mandatory property must be unique within the current realtime profile.
    - Cache provider: a dropdown, exposing the following values:
    - Redis (the default)
    - AzureRedis
    - SAPHana Database
    - Memcached
    - Cassandra
    - Couchbase

- MongoDB
- Ncache
- CosmosDB
- MemoryDB Redis

- Cache Provider Settings section: the contents of this section depend on the value selected at the Cache provider dropdown. All properties shown are mandatory.
  - If Redis:
    - Connection string: defaulting to the value 'localhost:6379,ssl=False'.
  - If AzureRedis:
    - Connection string: defaulting to the value 'localhost:6379,ssl=False'.
  - If SAPHana:
    - Server name: defaulting to the value 'ServerHostname'.
    - Schema: defaulting to the value 'SchemaName'.
    - User name: defaulting to the value 'User'.
    - Password: a password-masked field with no default value. ○ If Database:
    - Database ID: defaulting to the value '11111111-1111-1111-1111-111111111111'
  - ○ If Memcached:
    - Section name: defaulting to the value 'memcached' ○ If Cassandra:
    - Keyspace
    - Contact points
    - Username
    - Password ○ If Couchbase:
    - Server name
    - Username
    - Password
    - Bucket Name ○ If MongoDB:
    - Server name
    - Port
    - Schema
    - Collection name ○ If NCache:
    - Cache ID
    - Server List ○ If CosmosDB:
    - Database ID
    - Connection String ○ If MemoryDB Redis:
    - Host
    - Port

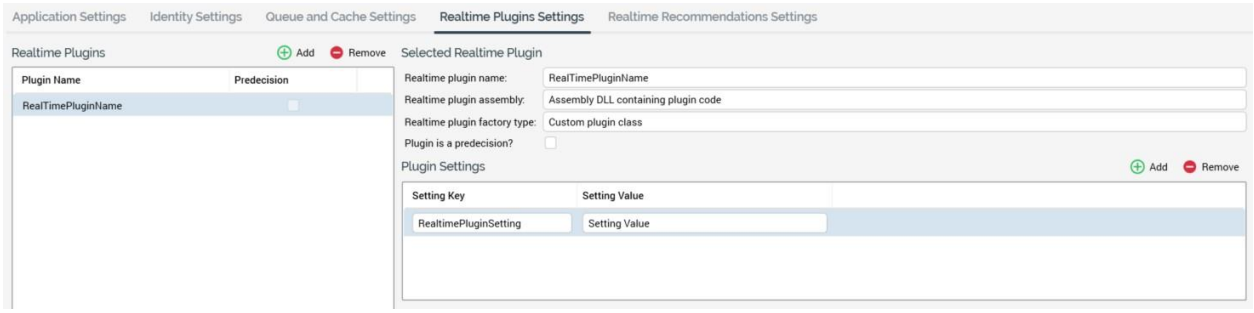
- SSL
- Username
- Password

The overlay also contains a footer, containing a single button:

- OK: clicking this button saves any changes and removes the overlay from display. Note that any validation errors must be resolved before the overlay can be closed.

Realtime Plugins Settings

This tab is used to manage the realtime plugins configured at the current RPI Realtime installation.



It consists of the following:

- Realtime Plugins: consisting of a toolbar and a list ○ Toolbar: exposing the following options:
  - Add new Realtime Plugin: clicking this button adds a new realtime profile to the list. Its default name is 'RealTimePluginName' (an incrementable integer can be appended to ensure uniqueness).
  - Remove Realtime Plugin: clicking this button removes the currently-selected realtime profile. An 'Are You Sure?' dialog is not shown.

  ○ List: displays all realtime plugins configured at the current realtime profile. The following read-only properties are shown:
  - Plugin Name
  - Predecision: a checkbox
- Selected Realtime Plugin section: when a realtime plugin is not selected, a message advises to select on from the list to view its details. The following properties are displayed: ○ Realtime plugin name: mandatory.

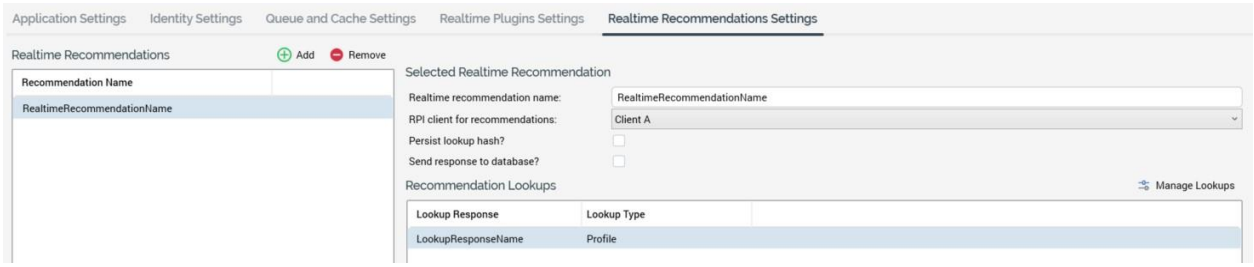
  ○ Realtime plugin assembly: mandatory. Defaults to the string 'Assembly DLL containing plugin code'.

  ○ Realtime plugin factory: mandatory. Defaults to the string 'Custom plugin class'.

- Plugin is a predecision?: this checkbox is unchecked by default. If checked, Predecision filter type and Precision Filter Contexts are shown. ○ Predecision filter type: accompanied by two radio buttons:
  - Include: selected by default
  - Exclude
- Predecision Filter Contexts: this property is only displayed if Plugin is a predecision checked. It consists of a toolbar and a list.
  - Toolbar: exposing the following options:
    - Add new Plugin Filter: clicking this button adds a new plugin filter to the list. Its default name is 'PluginFilter' (an incrementable integer can be appended to ensure uniqueness).
    - Remove Plugin Filter(s): clicking this button removes the selected plugin filter contexts without displaying an 'Are You Sure?' dialog.
  - List: one row is present in the list by default. It contains a single updateable field:
    - Filter Context: provision of a value in this context is mandatory. The supplied value must be unique within the list.
- Plugin Settings: this property consists of a toolbar and a list.
  - Toolbar: exposing the following options:
    - Add new Plugin Setting clicking this button adds a new plugin setting to the list. Its default Setting Key is 'RealtimePluginSetting' (an incrementable integer can be appended to ensure uniqueness), and Setting Value is 'Setting Value'.
    - Remove Plugin Filter(s): clicking this button removes the selected plugin settings without displaying an 'Are You Sure?' dialog.
  - List: one row is present in the list by default. It contains the following updateable fields:
    - Setting Key: provision of a value in this context is mandatory. The supplied value must be unique within the list.
    - Setting Value: provision of a value in this context is mandatory.

Realtime Recommendations Settings

This tab is used to manage RPI Realtime's Recommendations settings.

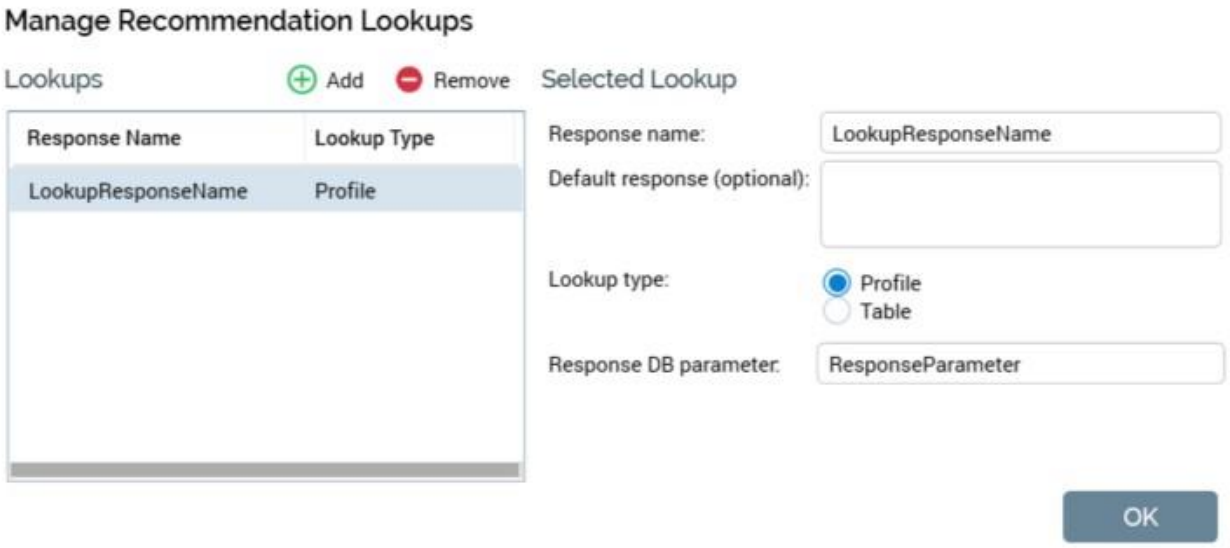


It consists of the following:

- Realtime Recommendations list: consisting of a toolbar and a list.

    - Toolbar: exposing the following options:
        - Add new Realtime Recommendation: clicking this button adds a new realtime recommendation to the list. Its default name is 'RealtimeRecommendationName' (an incrementable integer can be appended to ensure uniqueness)
        - Remove Realtime Recommendation: clicking this button removes the currentlyselected realtime recommendation from the list without displaying an 'Are You Sure?' dialog.
    - List: containing a single read-only column:
        - Recommendation Name
- Selected Realtime Recommendation section: exposing the following properties:
    - Realtime recommendation name: mandatory; must be unique within the realtime profile.
    - RPI client for recommendations: a dropdown field, listing all of the clients in the current RPI cluster.
    - Persist lookup hash?: a checkbox, which is unchecked by default.
    - Send response to database?: a checkbox, which is unchecked by default.
- Recommendation Lookups list: consisting of a toolbar and a list.
    - Toolbar: exposing a single option:
        - Manage Lookups: clicking this button displays the Manage Recommendation Lookups overlay (see below).
    - List: containing a single row by default, with default values as follows: Lookup Response: 'LookupResponseName', Lookup Type: ‘Profile’. the list contains the following read-only columns:
        - Lookup Response
        - Lookup Type

The Manage Recommendation Lookups overlay contains the following:



- Lookups list: consisting of a toolbar and a list.
  - Toolbar: exposing the following options:
    - Add new Recommendation Lookup: clicking this button adds a new row to list, with default values as follows:
- Response Name: 'LookupResponseName'. An incrementable integer can be appended to ensure uniqueness.
- Lookup Type: 'Profile'
    - Remove selected Recommendation Lookup: clicking this button removes the currently-selected lookup without display of an 'Are You Sure?' dialog.
  - List: containing the following read-only columns:
    - Response Name
    - Lookup Type
- Selected Lookup section: if a lookup is not selected, a message advising to select one from the list is displayed. The section exposes the following properties:
    - Response name: mandatory. The property defaults to the value 'LookupResponseName'. An incrementable integer can be appended to ensure uniqueness.
    - Default response: an optional, multi-line property.
    - Lookup type: accompanied by two radio buttons:
      - Profile: selected by default. When selected, the Response DB parameter property is shown, and the Schema name, Table name and Refresh trigger (minutes) properties are hidden.
      - Table: when selected, the Schema name, Table name and Refresh trigger (minutes) are shown, and the Response DB parameter is hidden.
    - Response DB parameter: this property is mandatory when shown. It defaults to the value 'ResponseParameter'.

- Schema name: this property is mandatory when shown. It defaults to the value 'dbo'.
- Table name: this property is mandatory when shown. It defaults to the value 'Recommendation_Table'.
- Refresh trigger (minutes): this integer property is mandatory when shown. It defaults to the value 10 and accepts a minimum value of 0 and a maximum value of 999,999,999.

- Table Lookup Additional Fields list: consisting of a toolbar and a list.
  - Toolbar: exposing the following options:
    - Add new Additional Field: clicking this button adds new row to grid, with defaults as follows:
- Database Field: 'FieldName' (an incrementable integer can be appended to ensure uniqueness).
- Default Value: 'DefaultValue'

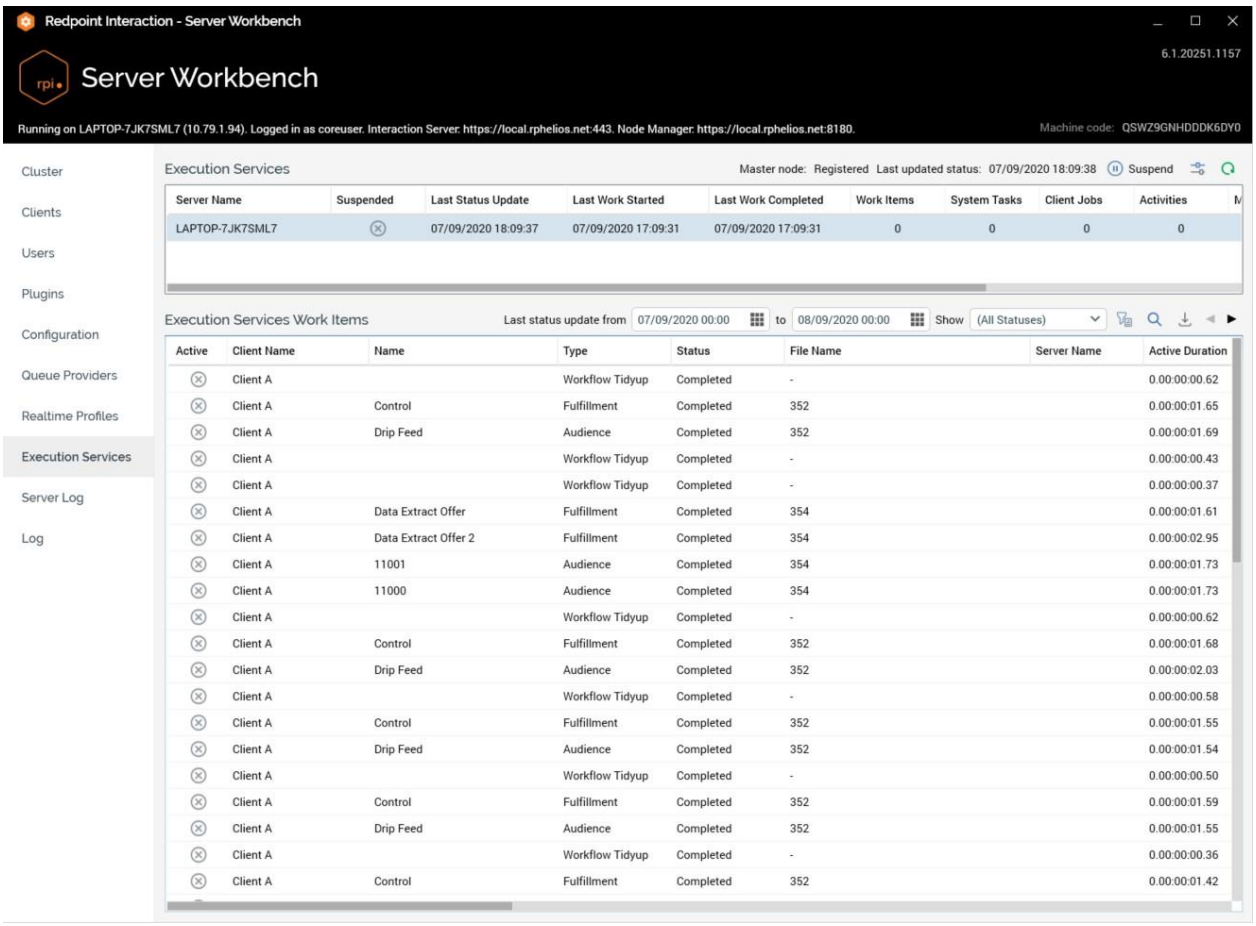
- Remove selected Additional Field(s): clicking this button removes the selected rows from the grid without display of an 'Are You Sure?' dialog.

- List: containing a single one row by default. The list exposes the following writeable columns:
  - Database Field: mandatory
  - Default Value: optional

- Footer: exposing a single button:
  - OK: clicking this button saves any changes and closes the overlay. Note that you cannot close the overlay until any validation errors therein have been resolved.

Execution Services

The Execution Services tab provides visibility of work undertaken at the current RPI cluster's Execution Service(s).



The tab contains Execution Services and Execution Services Work Items grids.

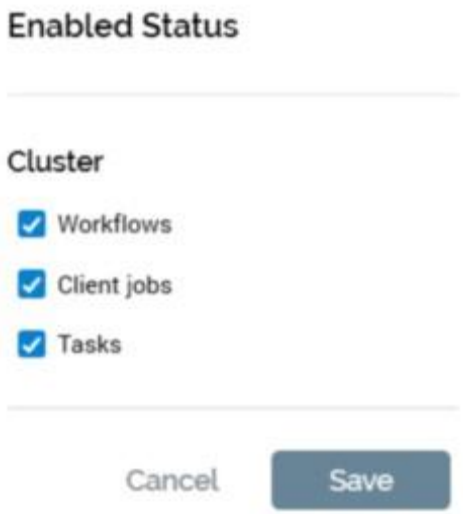
Execution Services Grid

The upper grid displays the list of Execution Services installed in the current Redpoint Interaction cluster.

A toolbar shown above the grid exposes the following:

- Master node: this read-only property indicates if a master Node Manager is currently registered as available to assign work. It is set to one of Registered or Not registered.
- Last updated status: the most recent date/time at which the master Node Manager updated its status.
- Suspend/Resume: this button allows you to Suspend or Resume the selected Execution Service. An 'Are You Sure?' dialog is shown when you suspend a service. When suspended, Workflows, Client jobs and Tasks behave as if not enabled (for more information, see the Manage/View Enabled Status at Cluster documentation below).

- Manage/View Enabled Status at Cluster: clicking this button displays the Enabled Status dialog, which allows you to enable or disable types of activities undertaken at the currently-selected Execution Service across all clients.



The button's icon is augmented as follows if one or more contexts of execution currently disabled:

The dialog contains a single Cluster lists, which displays three writeable checkboxes: ○ Workflows ○ Client jobs

- ○ Tasks

All are checked by default. The following apply when any of the checkboxes are unchecked.

- ○ Workflows: any new workflows initiated at the cluster will start playing, but no activity will take place therein. Within playing workflows, the next activity's execution will not commence.
- ○ Client jobs: any new client jobs initiated at the cluster will not execute. Any client jobs already executing are unaffected.
- ○ Tasks: any new system tasks initiated at the cluster will not execute. Any system tasks already executing are unaffected.

When checking a previously-unchecked checkbox, the following apply:

- ○ Workflows: any new workflows initiated at the cluster start playing. Within playing workflows, next activity execution will commence.
- ○ Client jobs: any new client jobs initiated at the cluster are executed. ○ Tasks: any new system tasks initiated at the cluster are executed.

Two buttons appear at the bottom of the dialog:

  - Save: clicking this button persists any changes made within the dialog and applies the same to the selected Execution service.
  - Cancel: clicking this button removes the dialog from display, abandoning any changes made therein. Clicking off the dialog has the same effect.
- Refresh: clicking this button reloads the list of Execution Services.

The read-only grid lists all of the current Redpoint Interaction cluster's Execution Services. For each, the following properties are displayed:

- Server Name
- Suspended
- Last Status Update
- Last Work Started
- Last Work Completed
- Work Items
- System Tasks
- Client Jobs
- Activities
- Max Workload
- Service ID

Execution Services Work Items Grid

The lower grid lists execution work items for the listed Execution Service(s).

A toolbar displayed above the grid exposes the following:

- Last status update from: this date/time defaults to today, 12:00AM.
- Last status update to: this date/time defaults to tomorrow, 12:00AM.
- Show: this dropdown allows you to filter the list of work items by status. It exposes the following values:
  - (All Statuses):
    the default
  - New
  - Initializing ○
    Running ○
    Suspended
  - Completed
    ○ Failed ○

Terminated ○
Invalid

- Show/Hide Execution Service Search Options: a toggle button, which, when selected, displays the Execution Service Search Options dialog. The dialog allows you to filter the list of work items by:
    - Client name ○
      Server name ○
      File name
- Search: clicking this button refreshes the grid with only work items that match the supplied criteria.
- Export Search Results: this button is enabled when results are displayed in the grid. Invocation displays the Save Execution Service Search Results Windows file system dialog. The default filename provided is 'Execution Service Search Results [yyyy-mmdd].txt'. You can click Save to initiate the file save (the file is saved directly, without creation of a job). The file itself contains a tab-delimited dump of the data displayed in the grid. A header row is provided.
- Previous page: this button is enabled when the list of work items will not fit into a single page, and a page other than the first is displayed. Invocation displays the previous page's worth of data.
- Next page: this button is enabled when the list of work items will not fit into a single page, and a page other than the last is displayed. Invocation displays the next page's worth of data.

The read-only grid lists work items instances matching the current search criteria. The number of records shown per page is controlled by system configuration setting MaxTableSearchRows. Records are presented in descending Added order.

The following columns are displayed:

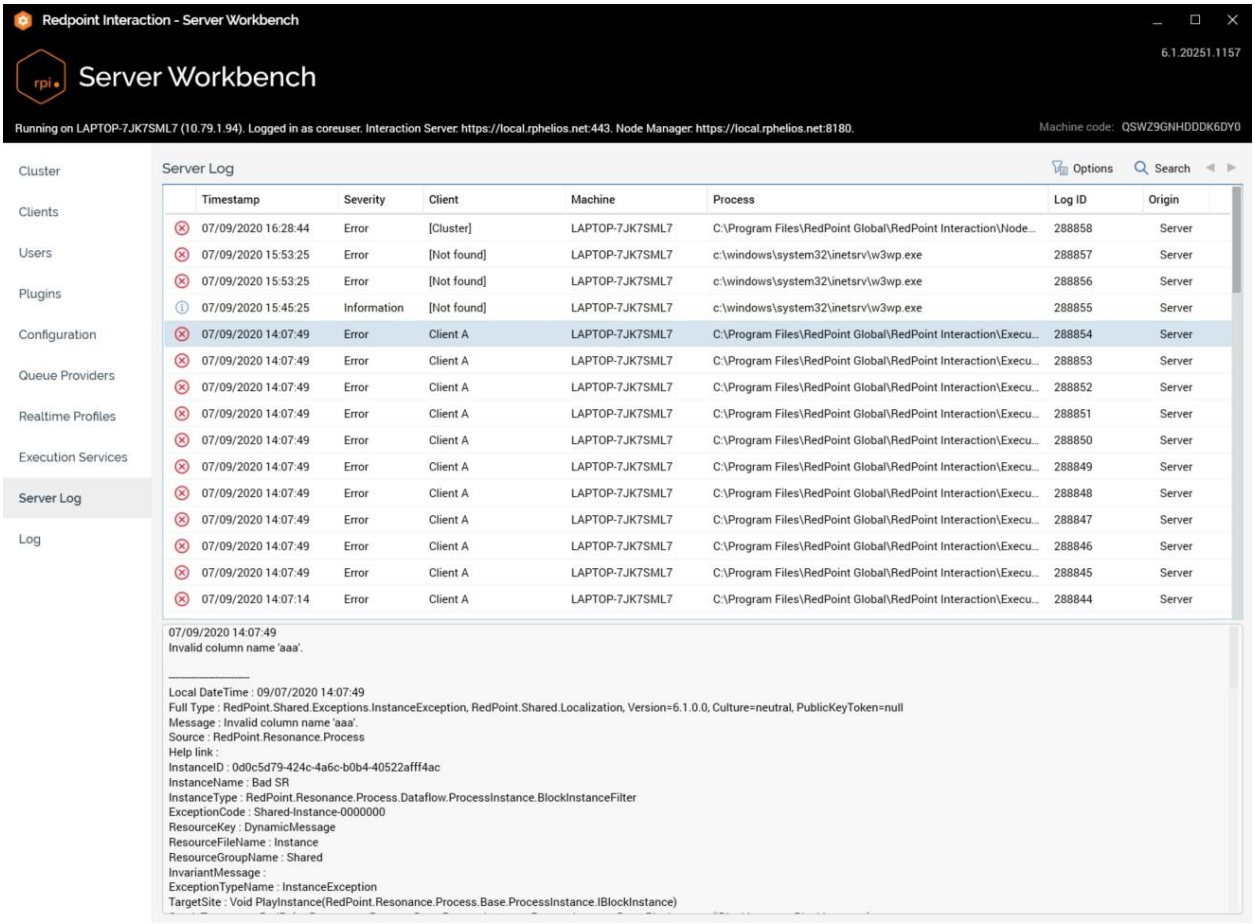
- Active: a tick or cross icon
- Client Name
- Name
- Type
- Status
- File: only if relevant; if populated, accompanied by Open Latest Version and Open File Location buttons
- Server Name: only shown when activity is currently taking place
- Active Duration: days, hours, minutes, seconds
- Added: date/time
- Scheduled: date/time
- Last Status Update: date/time
- Last Status Message

- Execution ID: GUID
- Workflow Association Instance ID: only if relevant
- Activity ID: GUID if relevant
- Workflow Added: date/time
- Workflow Active: a tick or cross icon
- Workflow Execution Status
- Execution Service ID: GUID; only shown when activity taking place
- File ID: GUID; only shown when relevant

You can sort data in the grid by clicking on a column header.

Server Log

The Server Log tab is used to access details of server log messages generated at all clients across the cluster.



It consists of a Server Log and selected log message sections.

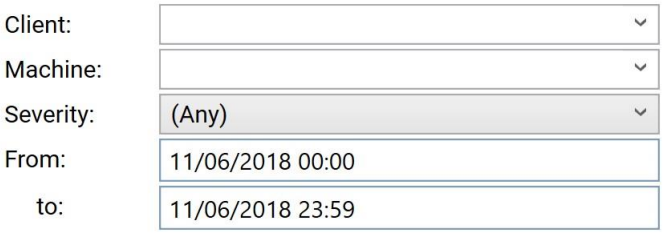
Server Log Section

The server log section contains a toolbar and grid

Server Log Toolbar

The Server Log toolbar exposes the following options:

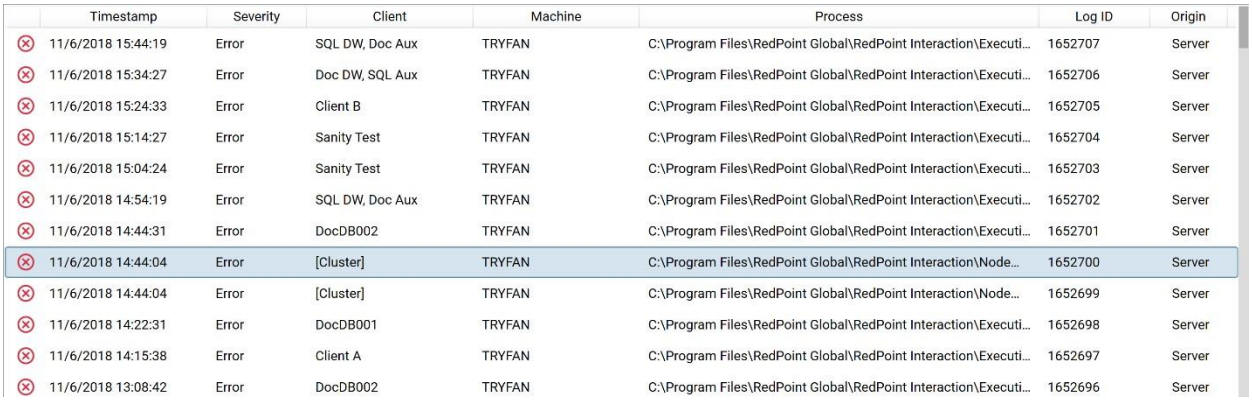
- Options: clicking this button displays a dialog:



  The dialog exposes the following options:
  - Client: dropdown, default blank. All clients configured at the cluster are listed.
  - Machine: dropdown, default blank. All machines within the current cluster are listed. ○ Severity: dropdown, values (Any) (the default), Critical, Error, Information, Warning.
  - From: defaults to today 00:00.
  - to: defaults to today 23:59.
- Search: invocation refreshes the grid with log messages matching the current Options and removes the Options dialog if displayed.
- Previous page: enabled when a previous page's worth of data is available. Invocation displays the same.
- Next page: enabled when a next page's worth of data is available. Invocation displays the same.

Server Log Grid

The Server Log grid lists server log messages that match the currently-specified options. The size of a page within the grid is hard-coded to 100 records.



Timestamp	Severity	Client	Machine	Process	Log ID	Origin
11/6/2018 15:44:19	Error	SQL DW, Doc Aux	TRYFAN	C:\Program Files\RedPoint Global\RedPoint Interaction\Executi...	1652707	Server
11/6/2018 15:34:27	Error	Doc DW, SQL Aux	TRYFAN	C:\Program Files\RedPoint Global\RedPoint Interaction\Executi...	1652706	Server
11/6/2018 15:24:33	Error	Client B	TRYFAN	C:\Program Files\RedPoint Global\RedPoint Interaction\Executi...	1652705	Server
11/6/2018 15:14:27	Error	Sanity Test	TRYFAN	C:\Program Files\RedPoint Global\RedPoint Interaction\Executi...	1652704	Server
11/6/2018 15:04:24	Error	Sanity Test	TRYFAN	C:\Program Files\RedPoint Global\RedPoint Interaction\Executi...	1652703	Server
11/6/2018 14:54:19	Error	SQL DW, Doc Aux	TRYFAN	C:\Program Files\RedPoint Global\RedPoint Interaction\Executi...	1652702	Server
11/6/2018 14:44:31	Error	DocDB002	TRYFAN	C:\Program Files\RedPoint Global\RedPoint Interaction\Executi...	1652701	Server
11/6/2018 14:44:04	Error	[Cluster]	TRYFAN	C:\Program Files\RedPoint Global\RedPoint Interaction\Node...	1652700	Server
11/6/2018 14:44:04	Error	[Cluster]	TRYFAN	C:\Program Files\RedPoint Global\RedPoint Interaction\Node...	1652699	Server
11/6/2018 14:22:31	Error	DocDB001	TRYFAN	C:\Program Files\RedPoint Global\RedPoint Interaction\Executi...	1652698	Server
11/6/2018 14:15:38	Error	Client A	TRYFAN	C:\Program Files\RedPoint Global\RedPoint Interaction\Executi...	1652697	Server
11/6/2018 13:08:42	Error	DocDB002	TRYFAN	C:\Program Files\RedPoint Global\RedPoint Interaction\Executi...	1652696	Server

The following read-only columns are displayed:

- Icon
- Timestamp
- Severity
- Client
- Machine
- Process
- Log ID
- Origin: Server or Client

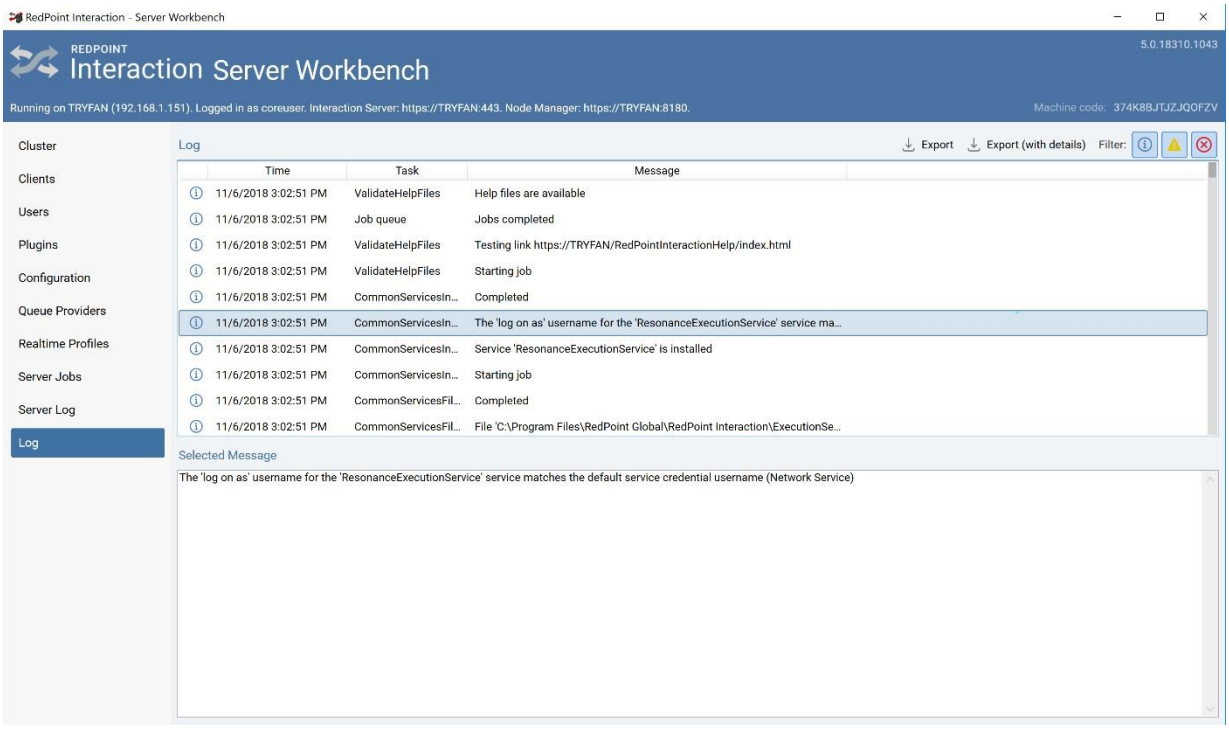
If no log messages are displayed, a message advises that 'No log messages were returned based on the current search criteria'. Log messages are displayed in descending Timestamp order.

Selected Log Message Section

The read-only section below the grid displays full details of the currently-selected log message.

Log

The Log tab is used to access informational, warning and error messages generated during RPI installation, upgrade or validation.



Note that a warning triangle is shown at the Log tab when one or more unread log messages are present therein.

The Log tab consists of Log and Selected Message sections.

Log Section

The Log section contains a toolbar and log messages list.

Log Toolbar

The toolbar exposes the following buttons:

- Export: this button allows you export the currently displayed log messages without full message details. Clicking the button displays the “Export Log to File” Windows file system dialog. The folder displayed defaults to the folder in which the Server Deployment Workbench executable is running. The default export file name is “DeploymentLog_[year][month][day]-[time]”, and default type is Text file.
  You can click Save to create the export file; a confirmatory dialog is displayed post-save. The export file contains log file messages as per the contents of the Log grid. Any filters and/or sorting applied to the grid are also observed within the export file.
  For each log message, the following tab-separated information is displayed: ○
  Time ○ Type ○ Message: full message details, where they exist, are not exported.
  You can also click Cancel to remove the Windows file system dialog from display.

- Export (with details): this button also allows you to save the displayed log messages to a file. However, full details are included for each message (when they exist).
- Clear: invocation of this option is protected by an “Are You Sure?” dialog and clears all log messages from the grid.
- Filter: these toggle buttons allow you to show or hide log messages of specific types:
    - ○ Information ○
    Warning ○ Error

Log Messages List

Log messages that match the current filter criteria are displayed in reverse chronological order.
The following columns are shown:

- [Type]: icon; one of:
    - ○ Information ○
    Warning ○ Error
- Time
- Task: the activity that was executing when the log message was created.
- Message

You can also highlight a single log message at a time in the grid.

Selected Message Section

This section is initially blank. When a log message is selected, its details are displayed here.

Where additional message details exist, they are also displayed. Selected Message is read-only; you can, if desired, highlight and copy the log message details shown therein.

Log Files

Log files are written to C:\ProgramData\Redpoint\DeploymentLogs. One log file per day for each of Core and Node is written.

Core logs are written at:

- Core install
- Core upgrade

The core log filename format is 'Core yyyymmdd.txt'

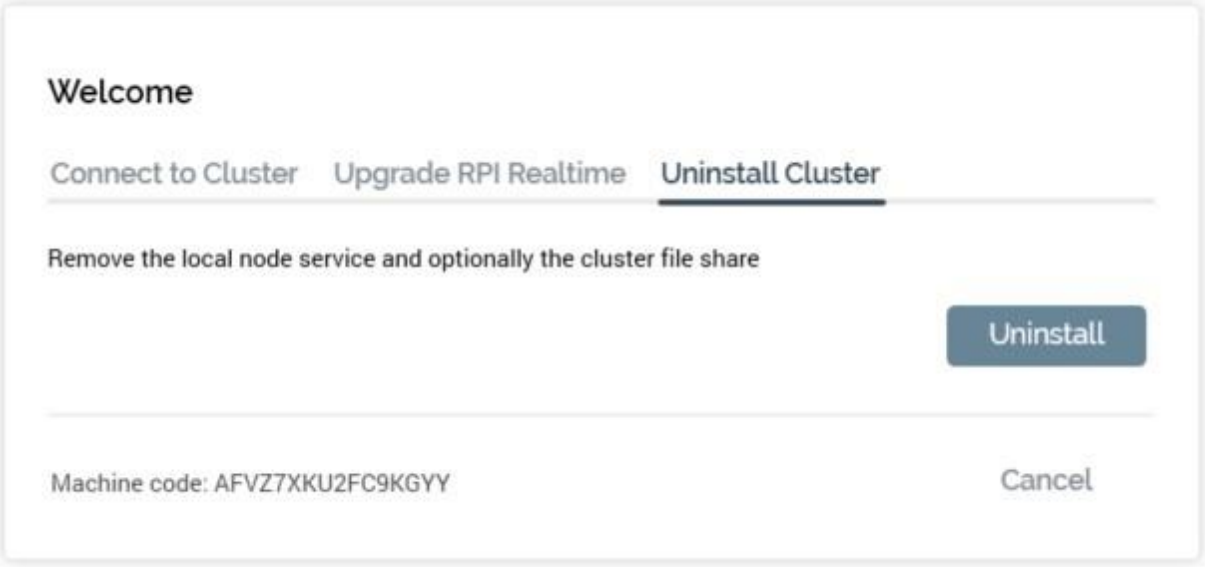
Node logs are written at:

- Node install
- Node upgrade
- Client install

The node log filename format is 'Node yyyymmdd.txt'

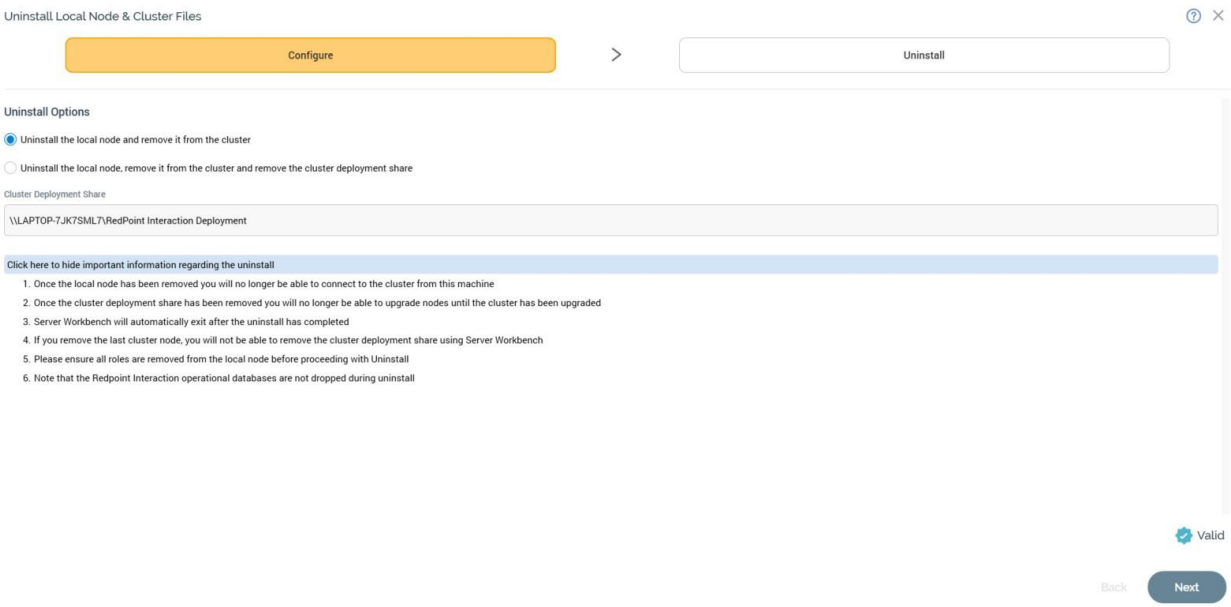
Uninstallation

You can start the process of uninstalling an RPI cluster by clicking the Uninstall button, displayed at the Uninstall Cluster tab in the Welcome overlay shown dialog when Server Workbench is initially displayed:



(Note that the Uninstall option is not available if the Node Manager service is not installed on the local machine).

When you click the button, the Uninstall overlay is displayed over the Cluster tab (with the only other available tab being the Log tab):



Note that, if running Server Workbench on a server in the cluster, and the deployment share is present on another node, the second Uninstall radio button is not shown.

Uninstall Local Node & Cluster Files Section

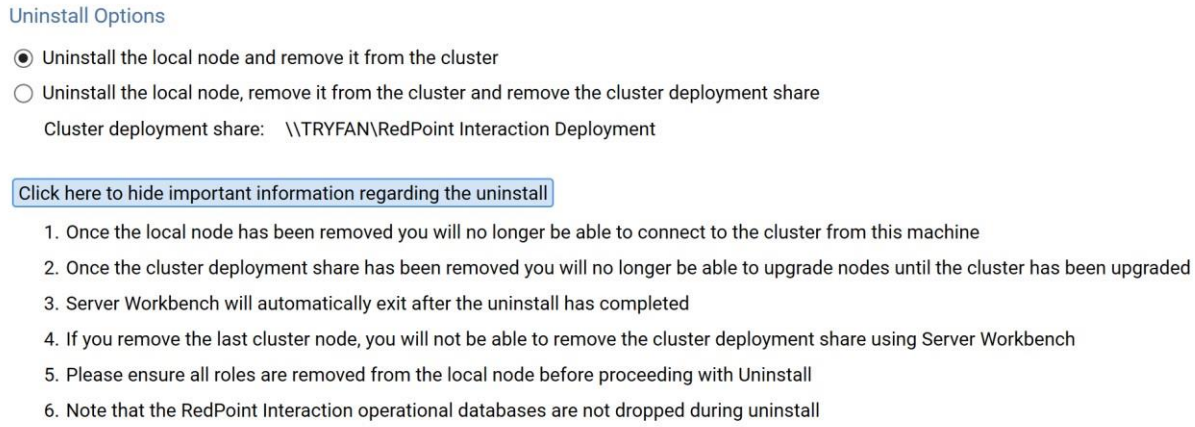
Uninstallation of the core is carried out in a wizard style, using two sub-interfaces:

- Configure
- Uninstall

A graphical representation of current progress through the wizard is displayed within the top section:

Configure Sub-interface

The Configure sub-interface allows you to specify uninstallation options.



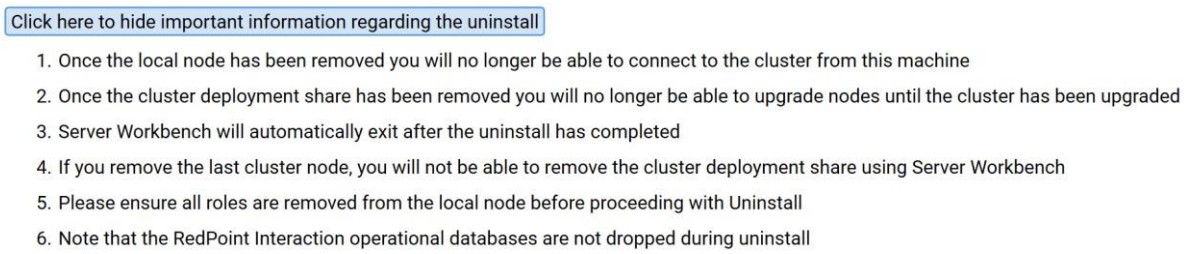
It contains a single section — Uninstall Options.

Uninstall Options

This section contains two radio buttons, which allow you to define the type of uninstallation to perform:

- Uninstall the local node and remove it from the cluster: this option is selected by default.
- Uninstall the local node, remove it from the cluster and remove the cluster deployment share.

A toggle button allows you to control whether or not to display a series of advisory notes:



Validation

Configuration within the Configure sub-interface is always valid, so a green indicator is always displayed. You may now move to the next wizard step.

Navigation

You can move forwards through the Uninstall wizard by clicking on the Next button at the bottom right. Doing so displays the Uninstall sub-interface.

Uninstall Sub-interface

This interface is used to provide visibility of the tasks executed during uninstallation of the core.

Toolbar

A single button is shown at the toolbar:

- Restart Uninstall: this button is available when the status is one of Completed with Warnings or Has a Dependent Job with Errors. It allows you to repeat the installation steps from the beginning.

Navigation

On completing the uninstallation, the Next button, displayed at the bottom right, is replaced by a Finish button. Clicking Finish closes the Server Workbench.

Note that operational databases are not removed during uninstallation of the core.

If you click Finish prior to completing the installation, you will be advised to wait until all steps are complete.

You can return to the Configure sub-interface using the Back button.

Deployment Scripts

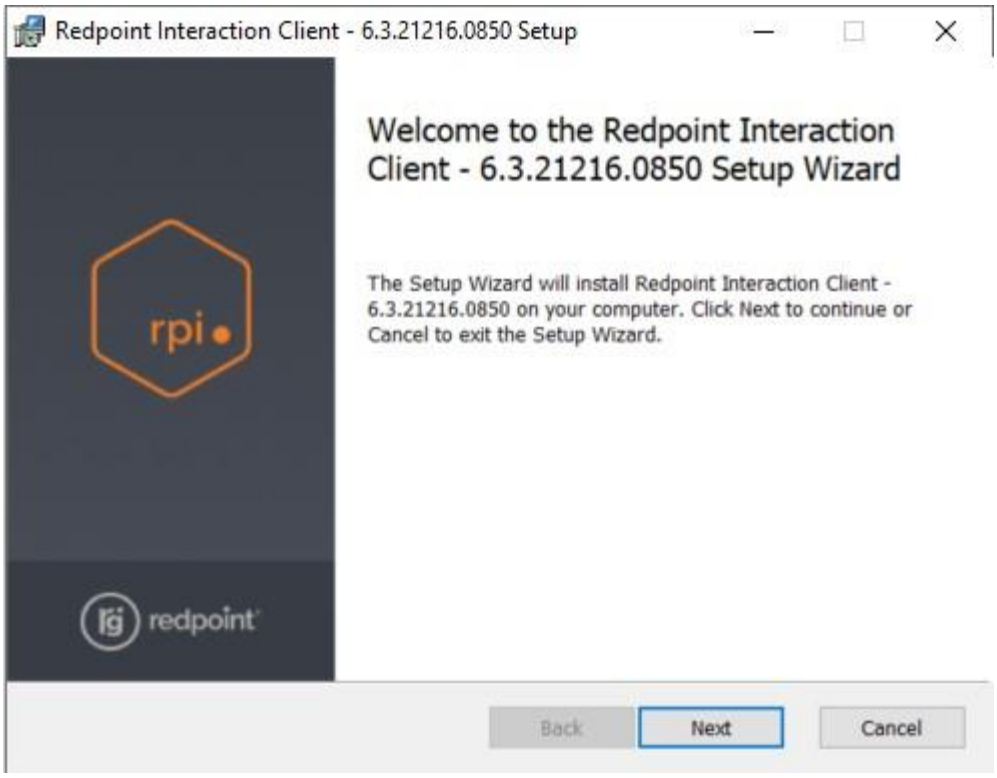
A series of deployment scripts have been provided to automate the process of deploying RPI. These are documented separately in the “Deployment Scripts” document.

RPI Client Application

The RPI client application can be installed by copying the Client folder within the RPI deployment files to an appropriate location, and then running the Interaction.exe application.

Alternatively, the RPI client application can be installed using the installer package in the ClientMSIInstaller folder.

Double clicking the file launches the RPI client installer:



Follow the steps therein to install the RPI client application.

The RPI client is version aware. If a server to which it connects has been upgraded to a later version, a client upgrade will automatically take place. Details of the automatic client upgrade process can be found in the RPI User Guide.

There are contexts within which it is not desirable for the RPI client application to update itself. In these cases, settings within the Interaction.exe.config file, which can be found in the same folder as Interaction.exe, can be used to prevent automatic client updates taking place.

- IsClientUpdateAllowed: this setting defaults to True. If True, the RPI client application will update itself automatically. If False, the client application will not be updated automatically on connecting to a later server version, and a message is displayed instead.
- The message to be shown is controlled by setting ClientUpdateDisallowedMessageOverride. If set to blank (the default), a default message (The RPI Client requires an update that cannot be performed automatically. Please contact your System Administrator for assistance) is displayed. Otherwise, the supplied message is shown.

Note that the installer can also be run by navigating to the 'installer' folder within the Interaction website.

Microsoft Edge WebView2

The RPI client uses the Microsoft Edge WebView2 Control to display HTML content within the application.

The *SkipWebView2RuntimeInstall* setting, which can be found in DeploymentFiles\Plugins\RedPoint.ClientInstaller.exe.config, controls, when Microsoft WebView2 is not already installed at an RPI client machine, whether installation of the RPI client should skip silent install of the same. The setting defaults to 'False', and must be set in advance of the upgrade to the RPI v6.3 server.

When attempting to run client in an environment when WebView2 is not installed, the following message is displayed at the Sign in dialog:

'Microsoft WebView2 Runtime was not found
This application requires the Microsoft WebView2 Runtime to be installed before it can be used. Contact your administrator or click the 'Copy to clipboard button' below to get the URL that can be used to download and install this component.'

The following buttons accompany the message:

- Copy to clipboard: clicking this button copies the following URL to the clipboard: **https://developer.microsoft.com/en-us/microsoft-edge/webview2/#download-section**
- Support: clicking this button opens Redpoint support page in the default browser.

Having upgraded server to v6.3, on connecting to the upgraded server with a previous client version, a WebView2 installer is displayed. If the installer cannot be downloaded (due to operating system or other restrictions), it will be necessary to download it manually from:

https://developer.microsoft.com/en-us/microsoft-edge/webview2/

Command Line Switches

You can initiate the RPI client application from a command Window by navigating to the folder in which the application executable is located, and issuing this command:

Interaction.exe

The following command line switches are available in this context:

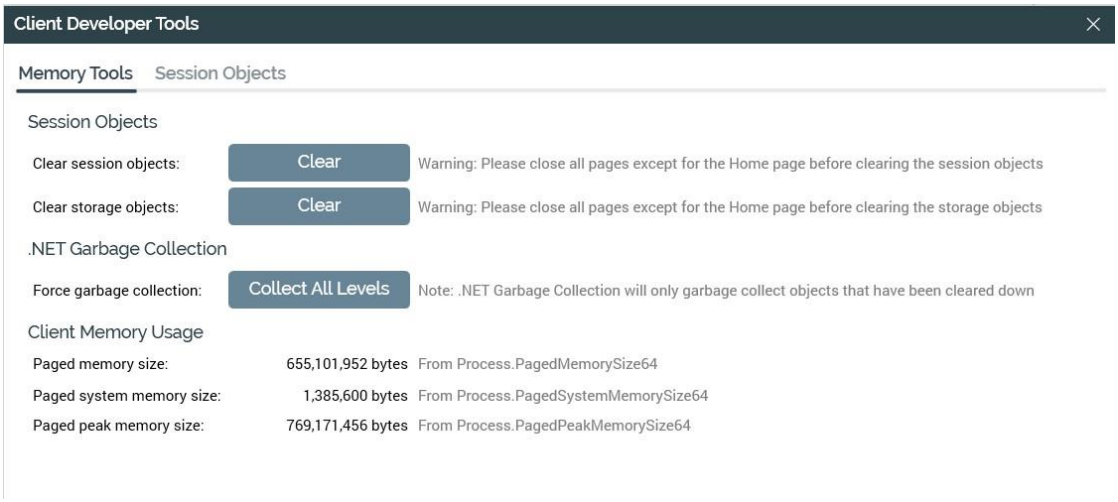
- -devmode: when this switch is applied, a new menu button is made available at the RPI client's header:

  Clicking the button displays a Developer tools context menu, exposing the following options:

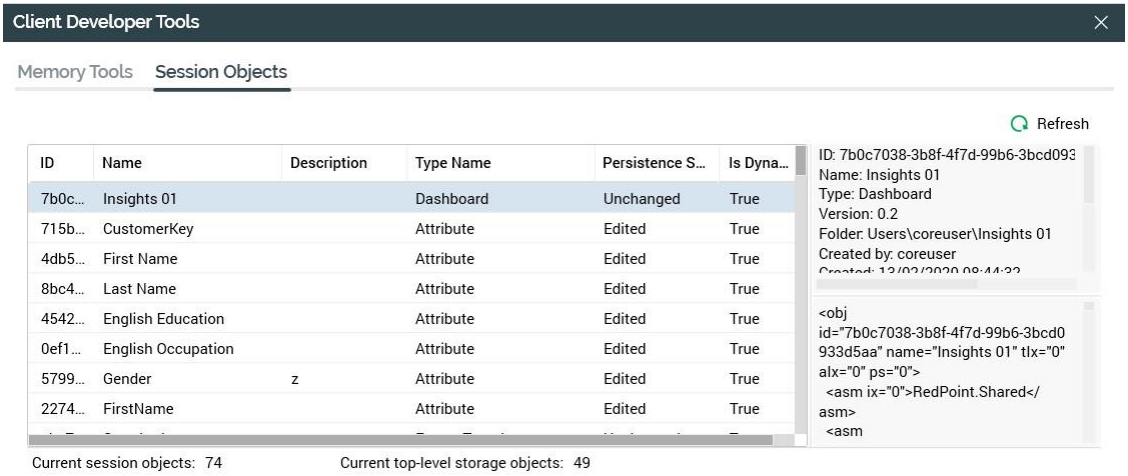
  - Developer Tools: selecting this option displays the Client Developer Tools dialog, with its Memory Tools and Session Objects tabs.

    The Memory Tools tab contains the following:

    - Session Objects section: the buttons within this section allow you to manually force the clearance from memory of RPI client session and storage objects.
    - .NET Garbage Collection: the button within this section allows you to manually initiate .NET garbage collection to free up client memory.
    - Client Memory Usage section: this section provides information about the current state of RPI client application memory usage.

The Session Objects tab provides visibility of the objects current held in the RPI client application's session memory.



Also when running in devmode, files' ID and VCID (Version Control ID) properties, and folders' ID properties, are shown in the following contexts:

- File information tooltip
- File System Dialog's information panel
- File Information Dialog's Details tab

- Force Inactivity Timeout: selecting this option causes the RPI client to behave as if signed out due to inactivity. For more information, please see the Framework section in the RPI Reference Guide.

- -logfullapierrors: when applied, all exceptions thrown when calling the InteractionAPI's methods are logged, with full details on the URI, HTTP Verb and Message Body.

- -logfullapicalls: when applied, all calls to InteractionAPI methods are logged, will full details on the URI, HTTP Verb and Message Body.

Operations

Licenses

When installing the RPI server, you must specify a valid license. A valid license:

- Was issued to the organization for which the RPI server is to be installed.
- Was created using the license machine code of the machines upon which RPI is to be installed (see below).
- Is perpetual or is scheduled to expire in the future.
- Is not constrained by application version or is constrained by the current application version or a higher version number.

If you attempt to install using an invalid license, the "Validate the application license file" check in the core Pre-Install Checklist will fail, and you will not be able to proceed with the installation.

This applies also when upgrading the RPI server to a new version.

When you try to connect your RPI client to a licensed server, you will be able to access functionality as expected if:

- The server has a perpetual license or the license's expiration date is greater than today.
- The license is not constrained by server application version or is constrained by an application version that is greater than or equal to the current server version.

If these conditions are not met, your RPI client will still run; however, you will not be able to access any tasks therein.

If a server license expires, you can upload a new, valid license file in the Plugins tab. When you do so, the new license will be installed automatically. You will then be able to access RPI functionality as required. Note a license must be more recent than the original to be uploaded in this way.

Note that the number of cluster nodes and clients can be constrained using parameters passed during the generation of an RPI license. In event of such constraints being in place, you will be unable to increase the number of entities in question beyond the limits defined by the license.

Note also that knowledge of target servers' license machine codes—details of which are recorded in a number of places throughout this documentation—are essential to the generation of RPI licenses. If charged with responsibility for this task, ensure that you have access to each machine's code during the generation process. Note that a server's machine code can be discovered using the utility provided at:

`Deployment Files\Utilities\MachineCodeUtility\RedPoint.Resonance.MachineCodeUtility.exe`

Running this application from a command Window will display the current server's machine code.

Interaction Server Maintenance

This section is intended for operations staff charged with the responsibility of managing an RPI server installation.

For detailed information on the RPI Operations Interface, please see the interface's User Guide.

Architecture Summary

Full details of the RPI architecture can be found in the Technical Architecture document.

RPI Services

The RPI application server consists of one web and two Windows services.

Name	**Type**	**Description**
Interaction API Service	Web service	Facilitates communication with RPI databases and Windows services.
Node Manager	Windows service	Assigns nodes' workload and performs node upgrades.
Execution	Windows service	Hosts the execution of Interaction workflows; runs scheduled or periodic background system tasks such as housekeeping and fulfillment tracking.

If the SQL Server upon which an RPI service depends is not available (for example, due to the MSSQLSERVER service being stopped) when the service starts, the service continues running and polls for SQL Server availability, rather than shutting down. An event log entry is written advising that the service was unable to start properly due to SQL's unavailability.

On starting the SQL service, the RPI service starts after approximately one minute, and an event log entry is written advising that the service started.

Interaction API Service

All user requests to the server are handled by the Interaction API service.

The status of any of the Windows services has no bearing on the user's ability to log in to the system.

All requests to the Interaction services are made using HTTPS. A valid certificate must be installed on the application server and the service must be configured to use it.

The service should run under an application pool that uses the .NET Framework (v4). The application pool should be running as a Windows/domain user with access to the file system on the application server and access to the data warehouse and operational databases.

If users experience issues logging in, it is recommended to ensure that the Interaction API website can be navigated to in a browser. Its URL is of the format:

https://[server][:port (optional)]/Interaction

The service has a web.config file. This file allows changes to be made to database connections, and logging configuration.

If you navigate to the RPI Interaction API Home Page, you can access Swagger documentation, which documents the API endpoints exposed by the Service.

To enable this option, the Interaction API web.config setting EnableSwagger must be set to True:

```
 <applicationSettings>
    <RedPoint.InteractionAPI.Properties.Settings>
      <setting name="EnableOAuth" serializeAs="String">
        <value>True</value>
      </setting>
      <setting name="EnableSwagger" serializeAs="String">
        <value>True</value>
      </setting>
    </RedPoint.InteractionAPI.Properties.Settings>
</applicationSettings>
```

Execution Service

The Execution Service provides an engine that controls execution of all server-side activities.

The RPI client communicates with the Execution service via the Interaction API web service. Most communication between the Interaction API service and the Execution service is conducted via the Pulse database. The Windows service polls the database regularly to pick up and process the next request.

Workflow instances can be tracked in the Client “Interaction_” database in the table “op_InteractionWorkflows”. If a scheduled trigger is activated, it is persisted in the table “op_InteractionTriggers”. Once the trigger fires, it creates a workflow instance and a record is inserted into “op_InteractionWorkflows”. Audiences are executed within a workflow instance and are tracked separately within the table “op_DataWorkflows”. These tables contain all the state information reported to the client.

Operational Databases

RPI uses two core operational databases:

The operational databases store information necessary for RPI to function. There are three operational databases:

- Pulse: the main operational database, Pulse stores details of cluster operational data, such as client jobs, login credentials and system tasks.
- Pulse_Logging: stores error log records.

Starting and Stopping the RPI Server

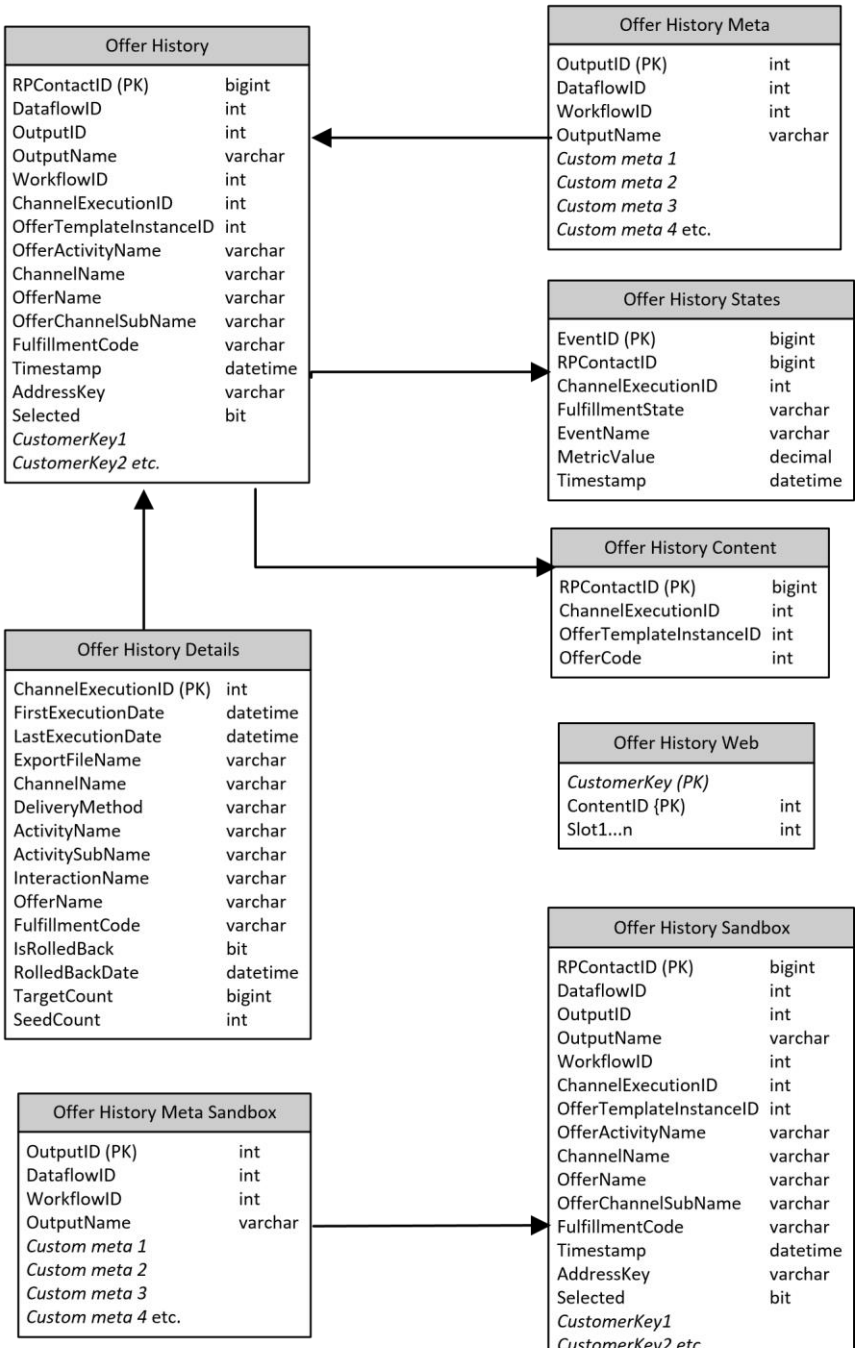
For the RPI server to start correctly, it needs access to the operational databases. If the databases are not available when the server is started it will monitor for their availability on a regular basis. Any issues are reported to the Event Log until the databases become available. Once the databases are started, the server completes initialization.

When the server is stopped, it will attempt to pause any active workflows. If any workflow is in a long-term busy state that cannot be paused, that workflow could be damaged by stopping the server. A common scenario would be processing a long-running SQL query. Care should be taken to stop the Execution service when the system is quiet.

Offer History Tables

The Interaction offer history tables are created in the data warehouse when users create audience definitions in the Administration Interface.

The tables created are as



follows:
The tables are described below:

Offer History

The main offer history table, which tracks a contact's audience output membership and the offer received. The keys persisted are configurable at audience definition administration.

Offer History Meta

This table persists audience output cell level metadata. The metadata persisted is configurable as part of the audience definition administration.

Offer History States

Use to track channel specific states or events that are captured post-execution of an offer activity.

Offer History Content

This table tracks the version of an offer sent to an individual. If the offer contains dynamic content, the instance of the content sent to an individual is also tracked.

Offer History Details

This table provides a synopsis of a specific fulfillment activity's execution (useful for reporting purposes).

Offer History Web

This table stores details of applicable web content following execution of a web offer.

Sandbox Tables

The "sandbox" tables are copies of the Offer History and Offer History Meta tables that are used for testing interactions.

Field Descriptions

The fields referenced in the diagram above are as follows:

RPContactID	Unique ID used to track an individual's receipt of an offer
DataflowID	Binds an individual to the instance of the audience used to select them
OutputID	Audience output (offer cell) ID
Output name	Audience output (segment) name

WorkflowID	Tracks the workflow instance that ran the activity
ChannelExecutionID	Tracks the instance of the activity/channel that fulfills the offer
OfferTemplateInstanceID	The version of offer content sent via the offer activity
OfferActivityName	Name of the offer activity
ChannelName	Name of the fulfillment channel
OfferChannelSubName	Name of the channel-specific activity that executed an element of an offer activity
OfferName	Name of the offer
FulfillmentCode	The user-supplied fulfillment code used for planning and reporting
Timestamp	The time the record was committed to offer history
AddressKey	Any channel specific address key (e.g., Email Address/Cell number)

Selected	Indicates if the record was sent to a third party channel (accounts for dedupes etc.)

FulfillmentState	The name of the fulfillment state that happened postexecution
EventName	Additional event information (e.g., URL for Email click through)
MetricValue	Any associated value linked to the fulfillment state event
OfferCode	The instance of dynamic content sent to an individual
FirstExecutionDate	Date and time at which activity started execution
LastExecutionDate	Date and time at which activity finished execution
ExportFileName	Name of export file created during activity execution
DeliveryMethod	Delivery method of channel through which fulfilled
Activity Name	Name of activity
ActivitySubName	Name given to the offer/channel list item when configuring offer activity's set of channels
InteractionName	Name of interaction in which activity executed
IsRolledBack	Flag indicating whether rolled back

RolledBackDate	Date and time of rollback
TargetCount	Number of records targeted by fulfillment activity
SeedCount	Number of seeds targeted by fulfillment activity
ContentID	Web content's Published folder ID
Slot	ID representing content for particular slot to be received by specific individual

Offer History Queues

The offer history queuing mechanism is designed to minimize the potential for insert contention resulting in the occurrence of blocking at an RPI cluster's offer history tables.

Two tables in each client's Interaction_[clientname] database facilitate the queuing mechanism:

- op_OfferHistoryInsertQueue: this table is used by RPI to temporarily store details of jobs containing records to be inserted into the offer history or offer history sandbox table.
- op_OfferHistoryInsertConfiguration: this table is used to store details of queues configured in the current RPI installation.

  If no records are present, the queuing mechanism is not used. A single row in the table represents a single queue. The table contains the following columns:

  - QueueID: the key column, the value for which is auto-generated.
  - TimeoutMinutes: the time for which records are to be maintained in a queue, after which an insert is performed. This timeout defines the maximum length of time that an insert is permitted to hold up a queue before the next job in line is permitted to start inserting records.
  - MaximumInsertSize: defines the limit as to the maximum number of records for a job that can be executed via the queue. This column allows certain queues to be reserved for smaller inserts, minimizing the potential for blocking by larger inserts. If the number of records to be inserted in a job exceeds all available queue limits, the system chooses the queue with the largest MaximumInsertSize. If this column is Null, all jobs may use it.

When performing offer history or offer history sandbox inserts, if more than one queue exists, one insert job at each can be executed concurrently. The next job uses the queue with the lowest permissible MaximumInsertSize value. A queue with an appropriate MaximumInsertSize value that has been set will be selected in preference to one where he value has not been set. By adopting this approach, the system retains larger-capacity queues for larger inserts, minimizing the chance that a smaller insert will be blocked by a larger one.

The queuing mechanism is used when executing interaction workflows in both Test and Production mode. It is used when executing control, export and offer activities. A given activity's results log describes the queue that was used.

Outputting Logs To File

By default, the RPI server application will write the error logs and SQL trace into the Pulse_Logging and InteractionAudit_xxx databases, respectively. It is also possible to log the information to a text file. Note, the text file will not contain the Client (Tenant ID) and in the case of the SQL trace, will not contain any additional information about the source or the result of executing the SQL.

The file output can be configured in the following .config files

\ServicesCommon\ExecutionService.exe.config (Log and SQL trace)

\NodeManager\NodeManagerService.exe.config (Log and SQL trace)

\Interaction\Web.config (Log only)

The log message destination is controlled by the <categorySources> section of the config

The 'Business Error' section covers error messages. A new flat file trace listener would be required to direct messages to a file. See the SQL Trace section for an example

```
<add switchValue="All" name="Business Error">
 <listeners>
  <add name="Database Trace Listener" />
 </listeners>
</add>
```

The 'SQL Trace' section is already configured with a flat file trace listener. Change the 'switchValue' to 'All' to output all messages

```
<add switchValue="Critical" name="SQL Trace">
 <listeners>
  <add name="SQL FlatFile TraceListener" />
 </listeners>
</add>
```

The 'General' section allows you gain verbose visibility of all generated log messages

```
<add switchValue="All" name="General">
 <listeners>
  <add name="Database Trace Listener" />
 </listeners>
</add>
```

Query Trace Log Retention

The period for which query trace logs are retained can be controlled by updating the Configuration field in Pulse.op_Housekeepers, for the record with Name 'Audit housekeeper'. The value is set using this format:

```
<config noOfDays="90" noOfDaysRetentionSQLTrace="28" />
```

Outputting Logs To New Relic

RPI optionally supports the output of logging information to the New Relic monitoring system. Further information can be found in the separate “Enabling New Relic Logging” document.

RPI Realtime

RPI Realtime consists of the following:

- RPI Realtime ASP.NET application.
- RPI Realtime Agent ASP.NET application: used by RPI Realtime to communicate with the RPI data warehouse and operational databases.

RPI's relationship with a client's website is discussed in the Redpoint Interaction Realtime section of the RPI User Guide. To summarize briefly, an integration between RPI and a public website can take place in the following contexts:

- Landing pages can be created in RPI and published to a website.
- Web forms can be included in landing pages, which, when completed and submitted by a site visitor, can serve as a useful data capture tool.
- An outbound RPI communication (e.g. an email offer) can contain a link to a landing or other web page, and results can be collated in respect of visitors' behavior at the same and presented in the context of the outbound offer activity within an interaction.
- In the same context, a target web page can contain dynamic content, which can be varied in accordance with the known visitor's characteristics.
- Realtime decisions can be used to personalize content in a dynamic asset hosted in a landing or other web page. Such personalization can be effected on several bases – e.g. the visitor's browser, location, time of day, current weather conditions or, if a known visitor, her or his attribute values.
- Goals, such as the clicking of a link or the submission of a web form, can be defined within a landing page, and visitor's attainment of the same can be monitored as conversion rates.
- Goal driven assets can be used to effect A/B/n testing within a website, with the results of the initial presentation to visitors of random pieces of content being analyzed statistically to determine the most efficacious piece of content in encouraging goal attainment. The winning content can then be served to subsequent visitors.

The RPI Realtime application is available in the following subfolders in the RPI DeploymentFiles folder:

- InteractionRealtimeAPI: the files within this folder should be used to install RPI Realtime in a Microsoft .NET Framework environment.

  Realtime web service configuration settings are managed in the appsettings.json.config file.

  Note: The .NET Core 2.2 version of the Realtime API is no longer shipped or supported as of version 6.2

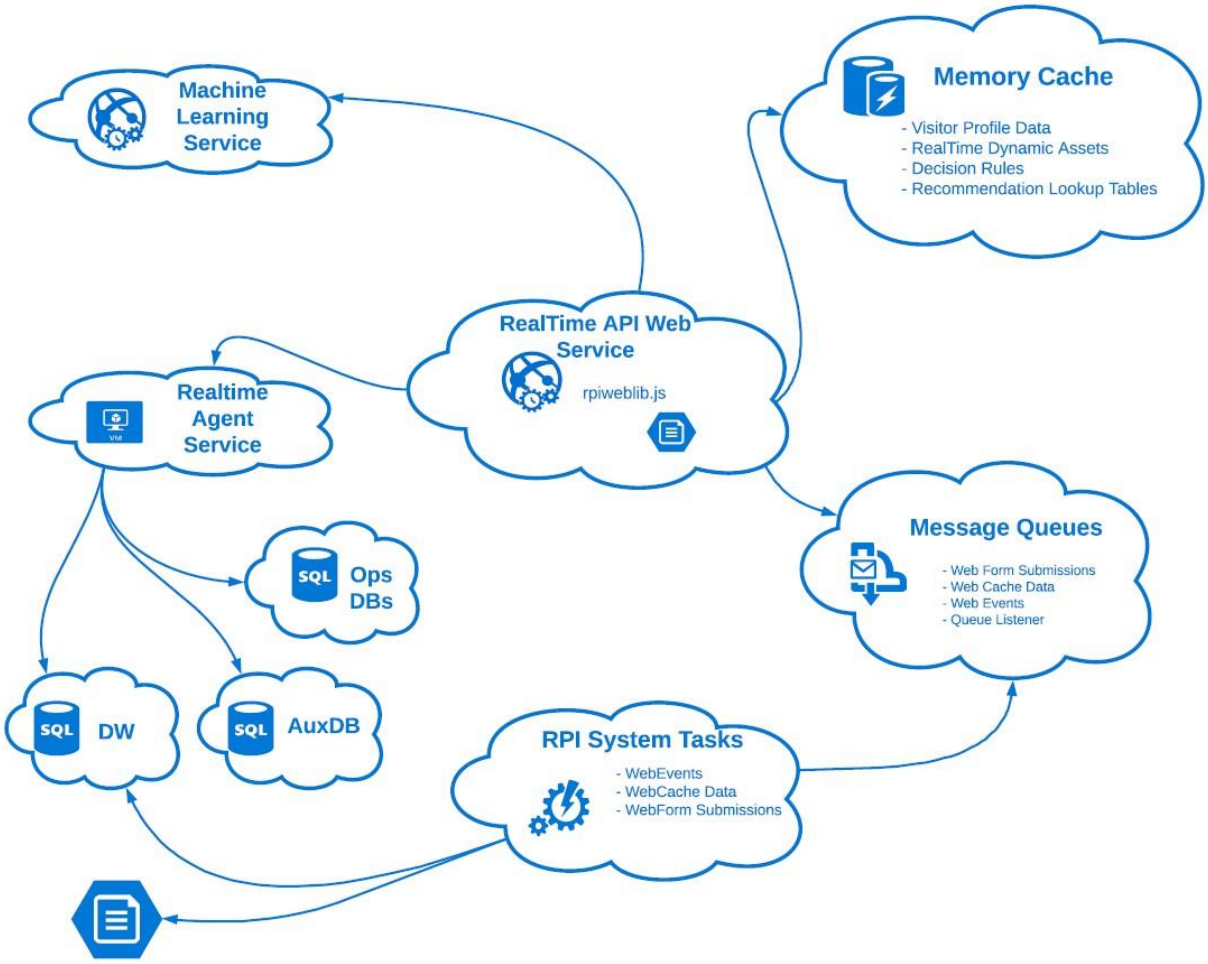
- InteractionRealtimeAgent

To set up RPI Realtime, the following pre-requisites must exist for each client that intends to use RPI Realtime:

- A public-facing web server.
- An RPI Realtime website.

Note that the public-facing website that hosts RPI landing pages, and the RPI Realtime website, which facilitates e.g. realtime decisions, web events, web form processing, must be hosted separately, and not combined into a single website.

The following illustrates the logical relationship between the components of RPI Realtime:



Installing RPI Realtime – Microsoft .NET Framework

To install RPI Realtime under the .NET Framework, please follow these steps for each client:

1. Copy the contents of DeploymentFiles\InteractionRealtimeAPI into the RPI Realtime website's root folder.

2. In RPI's Configuration Workbench, open the System Configuration interface, and set configuration setting RealtimeAPIAddress set to the RPI Realtime website's full URL.
3. Follow instructions in the sections below to configure the following. Note that it is only necessary to undertake configuration in the areas that support relevant functionality; for example, if you wish to use landing pages, but not realtime decisions, you only need to follow the instructions in the Web events section.
    - Web events: used the following contexts:
        - RPI landing pages.
        - Collation of state and metric information relating to visitors' behavior at a website, to which directed in an outbound communication
        - Tracking of goals.
    - Web forms
    - Realtime decisions
4. For added security, at the RPI Realtime website's web.config file, locate setting Access_Control_Allow_Origin. Change the "*" value to the landing page website address.

Configuration of the RPI Realtime website is carried out using settings defined in the appsettings.json.config file. Please see **here** for more information.

If required, you can generate an appsettings.json.config file from the Server Workbench's Realtime Profiles tab. Please see **here** for more information.

appsettings.json Configuration File

The appsettings.json.config file is used to manage RPI Realtime website settings.

Note that, on saving changes made within the file, an IIS restart is not required unless made within the Logging section.

Properties are exposed in sections within the file as follows:

AppSettings

- RPIAuthToken: authorization token used when connecting to the RPI Realtime API. **Note that this setting's default value should be changed to a fresh GUID at RPI Realtime installation.**
- RPIClientID: the ID (GUID) of the RPI client the Realtime API engine is associated with. The client ID can be obtained from Server Workbench.
- RealtimeAgentAddress: the address of the RPI Realtime Agent service, which is used by RPI Realtime to communicate with the RPI data warehouse and operational database.

- RealtimeAgentAuthToken: authorization token to be used by the RPI Realtime API when connecting to the Realtime Agent. **Note that this setting's default value should be changed to a fresh GUID at RPI Realtime installation.**

- ThresholdBetweenPageVisitsMinutes: defines the time, in minutes, after which a visitor's visiting a previously-visited web page will be deemed a new visit.

- MessageDaysExpiry: the number of days after which entries in a visitor's message history will expire.

- CacheWebFormData: defines whether to persist submitted web form data in a visitor's profile.

- CachedAttributeLoadTimeoutMS: the time, in milliseconds, after which RPI Realtime will time out when attempting to load data from the data warehouse into the realtime cache. If a call times out, the retrieval will be performed asynchronously. A value of 0 indicates that no timeout will be applied.

- MessageHistoryRefreshIntervalMinutes: defines the period, in minutes, after which a visitor's cached message history will be refreshed from the data warehouse.

- EnableHelpPages: controls whether documentation will be made available at the RPI Realtime API website.

- NoDaysPersistWebEvents: the length of time, in days, for which web events data will be persisted in a visitor's profile.

- NoMinsCacheSelectionResults: the time period, in minutes, for which the results of selection rules, executed in the context of selection rule realtime decisions, are persisted.

- RedpointMLServiceAddress: the address of the Redpoint Machine Learning service.

- OverrideMinThreadPoolSize: facilitates specification of a minimum thread pool size different to the default setting.

- MinThreadPoolSizeOverride: used in conjunction with the setting above; specifies the overridden minimum thread pool size.

- ParameterToDataMappings: facilitates the mapping of querystring parameters to cached attribute list lookup keys, as per the following example:

```
"ParameterToDataMappings": [
    {"ParameterName": "Email", "CALName": "CAL01"},
    {"ParameterName": "Employee", "CALName": "CachedTable01"},
    {"ParameterName": "CustomerKey", "CALName": "Cassandra"}]
```

- HashVisitorID: if set to True, visitor IDs as returned from the Realtime API are hashed and therefore obfuscated.

- CacheOutputExclusions: allows for the specification of a series of parameters that are not output when cache data is written to the database or file.
- CheckForProfileUpdatesOnSave: if True, a check will be performed to ensure that no intervening changes have occurred when a visitor profile is updated.
- MessageHistoryProfileRecordLimit: allows the maximum number of message history records per visitor that can be stored in the visitor profile to be defined.
- MessageHistoryDaysInProfile: controls the length of time for which message history data can be stored in the visitor profile.
- RealtimeServerCookieEnabled: specifies whether or not server-side cookies are enabled for RPI Realtime. Defaults to False.
- RealtimeServerCookieHttpOnly: specifies whether or not the server-side visitor cookie will be set using the HTTPOnly parameter. When set to true, the cookie will be inaccessible to web browsers through JavaScript. Defaults to False.
- RealtimeServerCookieName: specifies the name of the server-side visitor cookie. Defaults to 'rg-visitor'.
- RealtimeServerCookieExpires: specifies the number of days that the cookie expiration will be set for. Defaults to 60
- RealtimeServerCookieDomain: specifies the domain that the cookie will be set for. Default is blank. When blank, the cookie domain will be set using the FQDN that the Realtime request was made on.
- CORSOrigins: specifies the valid domain names that can make requests to Realtime. Default is an empty list. When empty, RPI Realtime will uses the value sent in the origin header of the request in its CORS response header.
- CacheOutputCollectIPAddress: controls whether visitor IP addresses are stored in the RPI_WebDevices table.
- RealtimeAgentInProcessEnabled: when set to True, the RPI Realtime Agent will run inprocess. For more information, please see elsewhere in the Admin Guide.
- SaveProfilePostDecisionResponse: this setting defaults to False. When false, visitor profile changes are saved using the main RPI Realtime execution thread, prior to return of results. When true, visitor profile changes are saved using a separate execution thread.
- ThresholdBetweenSiteVisitsMinutes: defines period after which a visit to a previouslyvisited site by a given visitor is considered a new visit.
- NoMinsCacheRecommendResults: defines the length of time, in minutes, for which recommendation results are stored in a visitor’s profile.

- EnableProfileMergeEvents: if this setting is set to true, a ProfileMerge event is raised on the merging of two visitor profiles.
- TrackProfileMergeInSourceProfile: if this setting is set to true, when two profiles are merged, the profile from which merged is updated with the merge details.
- MaxNoEventMetadataInstances
- EnableAuditMetricsInHeaders: this setting enables or disables the output of audit trace information in endpoints' response headers. By default, it is set to False. When enabled, the following audit headers are made available:
    - RPI-SmartAssets: the number of smart assets evaluated during the call ○ RPI-Rules: the number of rules evaluated during the call
    - RPI-DatabaseRules: the number of rules executed involving database execution via the Realtime Agent during the call ○ RPI-Cals: the number of cached attribute list lookups performed during the call ○ RPI-Models: the number of AML predictions made during the call ○ RPI-MergeProfile: the number of visitor profile merges made during the call
    - RPI-RecLookup: the number of recommender field lookups made to directly fetch fields from a database table or value list during the call
    - RPI-RecTableLoad: the number of recommender field table loads made to cache recommender fields in the realtime cache during the call
    - RPI-Context: any layout or area realtime context path used for smart asset evaluation during the call ○ RPI-View: any view name requested during the call
    - RPI-StartTime ○ RPI-EndTime
- DecisionCacheDuration: this setting controls the number of seconds for which smart asset logic is stored in memory. It is set to 5 seconds by default. If set to 0, logic is not cached.
- WebVisitorCacheDuration: controls the number of seconds for which visitor profiles are cached in memory. This feature is designed to increase the performance of operations upon visitor profiles, as it reduces the number of calls to the realtime cache. It should only be used in environments where only a single system is interacting with a particular visitor profile at a time, and, in the case of a multi-node RPI Realtime setup, the load balancer should have sticky sessions enabled.
- EventDaysToPersistOverride: facilitates the setting of a separate expiry date duration for a visitor's web events key.

When server-side cookies are enabled, RPI will set a server-side cookie that contains visitor and device IDs for a request. For every incoming request, if the visitor and/or device IDs specified in the request do not match the server side values, the service will override the values sent in with the request with the values from the server-side cookie. The response will include the updated visitor and device IDs and the consuming application will need to handle that appropriately. The Realtime Web Client (JavaScript SDK) will automatically pick up this change and set the client side visitor cookie appropriately.

Queues

- FormQueuePath: the path of the queue used to manage web form submissions.
- EventsQueuePath: the path of the queue used to manage the processing of web events
- CacheOutputQueueEnabled: controls whether visitor profile changes are sent to the queue defined in WebCacheQueuePath.
- CacheOutputQueuePath: the path of the queue used to process visitor profile data.
- RecommendationsQueuePath: the path of the queue used to manage RPI Realtime recommendations.
- ClientQueueSettings: facilitates definition of the queue technologies to be used by RPI Realtime. In this example, queues are hosted by the Microsoft Azure Service Bus provider:

```
"ClientQueueSettings": {
    "Assembly": "Redpoint.Resonance.AzureQueueAccess",
    "Type": "Redpoint.Resonance.AzureQueueAccess.AzureServiceBusQueueFactory",
    "Settings": [
        {
        "Key": "xxx",
        "Value":
"Endpoint=sb://xxx.servicebus.windows.net/;SharedAccessKeyName=RootManageShared
AccessKey;SharedAccessKey=xxx"
        }]
```

- ListenerQueuePath: path of the Listener queue. For more information, please see the Queue Listener section within the Interaction Designer documentation.
- ListenerQueueSettings: facilitates definition of the queue technology to be used to host the Listener queue.

IdentitySettings

- MasterKeyName: the name of the primary attribute used to key entries in a known visitor's profile.
- MasterKeyAliases: aliases by which the master key can be referenced.
- AlternativeKeys: alternative keys by which a known visitor's profile can be indexed.
- IdentityParameters: facilitates the mapping of a visitor profile data parameter to a key in the data warehouse.

The following settings control whether profile parametervalues/database values are to be merged on the occurrence of a ProfileMerge event. They provide for more granular control of the data that is carried over when a realtime visitor moves from an anonymous to a known state

- MergeParameters
- MergeCALAttributes
- ParameterMergeExclusions (array)
- CALMergeExclusions (array)

CacheSettings

- Caches: facilitates definition of the cache technologies to be used by RPI Realtime. In this example, caches could be hosted using the Azure Redis and Cassandra providers:

```
"Caches": [
    {
    "Name": "AzureRedis",
    "Assembly": "Redpoint.Resonance.AzureRedisCache",
    "Class": "Redpoint.Resonance.AzureRedisCache.AzureRedisCacheHandler",
    "Settings": [
        {
        "Key": "ConnectionString",
        "Value":
"rpidevjh.redis.cache.windows.net:6380,password=QbFkhPcomFWoeOGOvsJihUAW/eguwAh
ZVX1jffeU04c=,ssl=True,abortConnect=False"
        }
        ]
    },
    {
    "Name": "Cassandra",
    "Assembly": "Redpoint.Resonance.Web.Shared",
    "Class": "Redpoint.Resonance.Web.Shared.Cache.DatabaseCache",
    "Settings": [
        {
        "Key": "ClientID",
        "Value": "D2A8B022-E87E-45E4-B9CD-3FC5809FD9C8"
        },
        {
        "Key": "DatabaseID",
        "Value": "45E3C456-B8D4-4EA0-9496-BD24AA4FD842"
        }
    ]
```

- DataMaps: allows you to specify which cache providers are to be used in which contexts by RPI Realtime. Data of the following types can be stored in separate caches:

- Visitor profile (profile, parameter value and database values) ○ Visitor history ○ Non-visitor data (e.g. published realtime decisions and landing pages) ○ Product Recommendations

- Visitor Backup: allows you to define a secondary backup cache, to which visitor profile data will be written automatically. When a visitor record is removed from the main cache and a parallel visitor cache has been configured, on the visitor profile's being loaded, the missing record is repopulated in the main cache from the parallel cache.

- Message History: this cache type is supported at the MongoDB, Couchbase and Database cache providers only. It allows for the configuration of a separate cache to persist visitors' message history (as served by advanced smart assets).

- Visitor Readonly Parameters: this cache is typically populated by external processes. When the visitor profile is loaded into memory, the visitor profile and visitor readonly parameters are loaded in parallel and merged. Visitor readonly parameters can be leveraged in realtime decisions (the use of attribute list decisions is recommended), but you cannot include them within personalized content.

  If a parameter is present in both, the visitor readonly parameter value takes precedence. If the visitor profile and visitor readonly parameters are hosted in the same physical cache, a datamap KeySetting setting is used to avoid key clashes; the key is set to '[Key value]+[KeySetting]'.

The DataMaps section allows you to specify which of the configured cache providers is to be used to support each type of cache. In addition, it allows you to specify the number of days for which data is to be persisted at each cache, as well as to whether data should be compressed when stored in a cache (note that a Cache Decompression utility is available to facilitate viewing compressed data; details can be found **here**). Caching data will reduce the size of the profile in the cache and improve performance persisting to remote caches. There is a performance impact compressing and decompressing profiles; if the cache is local then performance may be improved by switching off compression.

The ability to use separate caches and disparate technologies facilitates attainment of an optimal balance between data durability and performance.

In this example, the Cassandra database cache configured above is used to persist visitor history data; the aforementioned Azure Redis cache is used to hold all other types of cache data:

```
"DataMaps": [
    {
    "Type": "Visitor Profile",
    "Cache": "AzureRedis",
    "DaysToPersist": 28,
    "CompressData": "False"
    },
    {
    "Type": "Visitor History",
    "Cache": "Cassandra",
    "DaysToPersist": 28,
    "CompressData": "False"
```

```
        },
        {
        "Type": "Non Visitor Data",
        "Cache": "AzureRedis",
        "DaysToPersist": 365,
        "CompressData": "False"
        },
        {
        "Type": "Product Recommendations",
        "Cache": "AzureRedis",
        "DaysToPersist": 365,
        "CompressData": "False"
        }
        ]
```

ProductRecommentations

This section is used to configure one or more Product Recommendation endpoints at the current RPI Realtime installation.

The product recommendation endpoint allows you to perform lookups from the realtime cache or data warehouse. This provides third party web pages or applications with the ability to perform personalization (e.g. to provide product recommendations).

The productRecommendations section can contain one or more productRecommendation nodes, as per the following example:

```
"ProductRecommendations": [
    {
    "Name": "Foo",
    "ClientID": "xxx8b022-e87e-45e4-b9cd-3fc5809fdxxx",
    "PersistLookupHash": "True",
    "SendResponseToDatabase": "True",
    "ResponseTableName": "ProductRecommendations",
    "Lookup": [
        {
        "Type": "Table",
        "ResponseName": "Response1",
        "DatabaseID": "",
        "SchemaName": "dbo",
        "TableName": "ProdRecLookup",
        "RefreshTriggerMinutes": 1,
        "DefaultResponse": "",
        "AdditionalFields": [
        {
        "Name": "Additional1",
        "DefaultValue": ""
        },
       {
        "Name": "Additional2",
        "DefaultValue": ""
```

```
}]},
{
"Type": "Profile",
"ResponseName": "PersonalizedResponse",
"DBParameter": "MyResponse",
"DefaultResponse": "="[{'ProductID': 'xxx'},{'ProductID': 'yyy'}]"
}]}
]
```

The following documents the above:

- productRecommendations: multiple product recommendations can be supported per RPI Realtime instance.
    - name: the name by which the productRecommendation will be uniquely identified. ○ clientID:

      GUID value unique to the current RPI client.

    - persistLookupHash: this attribute defaults to False. If true, the lookup hash passed to the product recommendation endpoint will be persisted for a given visitor and used as a key for performing database lookups.
    - sendResponseToDatabase: this attribute controls whether a recommendation will be persisted at the data warehouse. If true, data will be written to the table defined by responseTableName, at execution of the Web events importer system task.
    - responseTableName: used in conjunction with sendResponseToDatabase. If that attribute is True, responseTableName is mandatory.
- lookup: RPI Realtime supports the provision of multiple lookups within a product recommendation.
    - type: this attribute defines whether the lookup will be made from a database table ('Table'), or a visitor's profile ('Profile').
    - responseName: this attribute represents the JSON attribute that contains the product recommendation response. If the lookup's type is Table, it also represents the field in the database table that holds the response
    - databaseID: this attribute can be used to define whether the lookup should be made from the data warehouse (if set to blank), or from an auxiliary database (if set to a database's ID GUID).
    - refreshTriggerMinutes: for performance, the database lookup table is cached in the realtime cache. This attribute represents the period, in minutes, after which a refresh of the cached values will be triggered.
    - defaultResponse: in the case of the Table lookup, if a matching lookup hash is not supplied to the product recommendation service, the defaultResponse is returned. For a Profile lookup, if the database parameter is missing, the defaultResponse is returned

    - dbParameter: the name of the visitor profile database parameter that is used to persist the recommendation.

- additionalFields: this node is used to persist details of additional JSON packet data as returned from a database 'Table' lookup. A default can be provided for each.

The product recommendation endpoint can be invoked using the following HTTP GET call: `http://[server]/api/Recommendations/[visitorID]?LookupHash=[value][&ReturnAll=true/false][&recommendationName=name][&LogEvent=true/false]` Where the above reference the following:

- visitorID: the visitor's unique realtime cache key.
- LookupHash: a value used as the key when performing the lookup.
- ReturnAll: if set to True, all of the product recommendation's lookups will be returned.
- recommendationName: if the name of the specific product recommendation to be called. If blank, the first is invoked.
- LogEvent: facilitates overriding of productRecommendation setting sendResponseToDatabase

Plugins

The Plugins section is used to record details of the pre- and post-processing plugins configured at the current RPI installation.

- Pre-processing plugins can be used to manipulate visitor profiles prior to a call being made for a realtime decision content.

  A pre-processing plugin is defined as per the following example:

```
{
"Name": "PreProcessing",
"Factory":
    {
    "Assembly": "Redpoint.Resonance.Web.Shared",
    "Type": "Redpoint.Resonance.Web.Shared.Plugins.PreDecisionExampleFactory"
    }, "Type":
    {
    "Name": "Predecision",
    "ApiContentFilterOperator": "Include",
    "ApiContextFilters": [ "PrePluginTest" ]
    },
"Settings": [
    {
    "Key": "Param",
    "Value": "Foo"
    }
    ]
}
```

  The value passed in an API call's apiContext property is used to determine whether a pre-processing plugin is to be applied. You can specify multiple comma-delimited values in a list. apiContentFilterOperator is used to Include or Exclude whether a preprocessing plugin is to be invoked.

  For example, if a value passed in apiContext when an API call is made matches a value in the web config setting 'apiContextFilter', and apiContentFilterOperator is set to Include, pre-processing plugin functionality will be called. The functionality would not be called if apiContentFilterOperator was set to Exclude.

- Post-processing plugins are defined as per the following example:

```
{
"Name": "PostProcessingExample",
"Factory":
    {
    "Assembly": "Redpoint.Resonance.Web.Shared",
    "Type": "Redpoint.Resonance.Web.Shared.Plugins.PostDecisionExampleFactory"
},
"Settings": [
    {
```

```
    "Key": "Prefix",
    "Value": "pre "
    },
    {
    "Key": "Suffix",
    "Value": "post"
    }
    ]
}
```

For more information, please see the RealtimeContentPlugins configuration setting, and realtime dynamic asset/goal driven asset/inbound message list documentation elsewhere in the User Guide.

- Web form processing plugins can be used to manipulate data received during web form submission.

```
{
"Name": "FormProcessing",
"Factory":
    {
    "Assembly": "Redpoint.Resonance.Web.Shared",
    "Type":
"Redpoint.Resonance.Web.Shared.Plugins.FormProcessingExampleFactory"
    },
"Settings": [
    {
    "Key": "PluginField1",
    "Value": "Rhubarb"
    },
    {
    "Key": "PluginField2",
    "Value": "Custard"
    }
    ]
    }
```

- Geolocation functionality can be provided by geolocation plugins:

```
"GeolocationSettings": {
"Provider": "My geo provider",
"APIKey": "",
"WeatherUnits" : "Metric",
"CoordsCacheTimespanMinutes": 10,
"PluginAssembly": "RedPoint.Resonance.Web.Shared",
"PluginType": "RedPoint.Resonance.Web.Shared.Plugins.GeolocationExamplePlugin",
"PluginSettings": []
}
```

VisitorViews

Visitor views can be used to define the attributes of a site visitor that can be retrieved using the Visitor Views RPI Realtime API endpoint. An example VisitorViews section is provided below:

```
"VisitorViews": [
  {
    "Name": "ViewA",
```

```
        "ParameterNames": [
          "Stringie",
          "DMTitle",
          "TestParam"
        ],
        "DatabaseFieldNames": [
          "First Name",
          "Yearly Income"
        ]
      }
],
```

QueueListenerConfiguration

This section defines the parameters to be sent to a queue listener. More detail is provided in the RPI Reference Guide. An example QueueListenerConfiguration section is provided below:

```
    "QueueListenerConfiguration": [
      {
        "Name": "Default",
        "SendAddressParameter": "EmailAddress",
        "Parameters": [
          "FirstName",
          "LastName"
        ]
      },
```

IDValidationSettings

This section allows you to specify rules around the type of data values that are acceptable to be used as visitor and device identifiers. An example IDValidationSettings section is provided below:

```
IDValidationSettings
    "IDValidationSettings": {
      "EnableVisitorIDValidation": true,
      "VisitorID": {
        "MinimumLength": 1,
        "MaximumLength": 36,
        "EnableLetters": true,
        "EnableNumbers": true,
        "PermittedCharacters": [
          "-",
          "_",
          "/",
          ".",
          "@",
          "#",
          "&",
          "?"
        ]
      },
      "EnableDeviceIDValidation": true,
      "DeviceID": {
        "MinimumLength": 1,
        "MaximumLength": 36,
        "EnableLetters": true,
```

```
        "EnableNumbers": true,
        "PermittedCharacters": [
          "-",
          "_",
          "/",
          ".",
          "@",
          "#",
          "&",
          "?"
        ]
      }
    }
  },
```

Logging

Logs can be written to a daily log file, defined in appsettings.json's FileLogging section. E.g.:

```
  "FileLogging": {
    "LogDirectory": "C:\\Redpoint\\Web Processing\\Logs",
"FileName": "EntryLog_"
```

Trace information can also be sent to other destinations as defined in the Logging section.

The level of logging detail can be set for each log destination. The options are:

- None
- Trace
- Information
- Warning
- Error

You can reference environment variables at the LogDirectory and FileName settings; e.g.:

`"LogDirectory": "C:\\RedPoint\\Web Processing\\Logs_[EnvironmentVariableName]"` The '[MachineName]' hard-coded variable is also supported in the same context.

For a .NET Framework deployment, logging to Azure and other sources requires configuration in both the appsettings.json and the web.config

First, configure which level of information to set in the appsettings.json.config using the "SystemDiagnostics" element.

```
"SystemDiagnostics": {
      "LogLevel": {
        "Default": "Trace"
      }
}
```

Then set any destinations in the web.config "system.diagnostics" section.

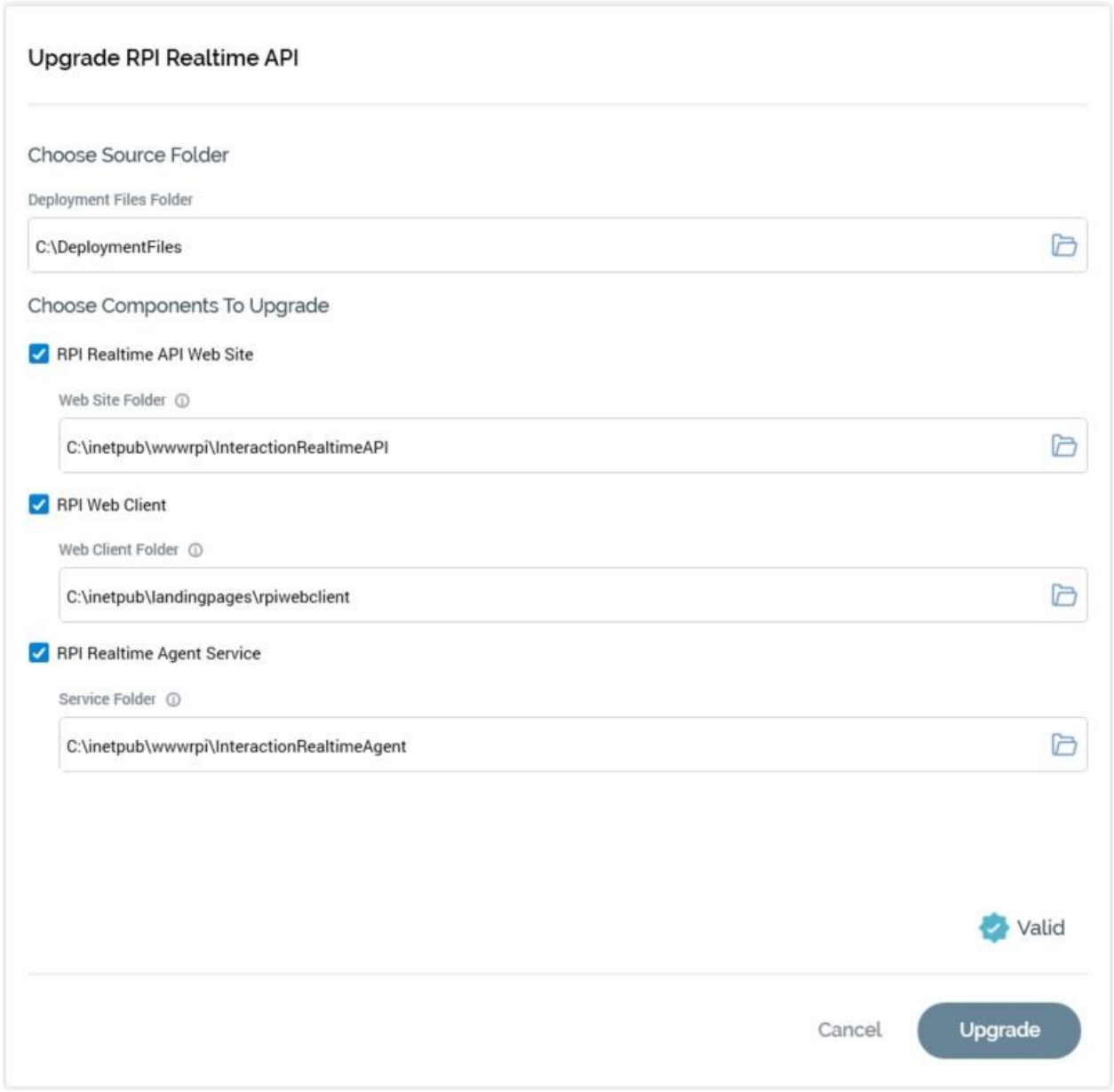
```
<system.diagnostics>
  <switches>
    <!--( 0=Off, 1=Error, 2=Warning, 3=Info, 4=Verbose )-->
    <add name="TraceLevel" value="2" />
    <!--( 0=Off, 1=On )-->
  </switches>
  <trace autoflush="true" >
    <listeners>
      <!--Use this to put trace messages to the event log-->
      <!--<add name="eventLogListener"
type="System.Diagnostics.EventLogTraceListener"
initializeData="InteractionRealtime" />-->
      <!--To use, you will need to add a trace element to System.Web.  Log can be
viewed at address /trace.axd-->
      <!--<add name="WebPageTraceListener" type="System.Web.WebPageTraceListener,
        System.Web,
        Version=4.0.0.0,
        Culture=neutral,
        PublicKeyToken=b03f5f7f11d50a3a" />-->
    </listeners>
  </trace>
  <sources>
  </sources>
</system.diagnostics>
```

Upgrading RPI Realtime

Upgrading RPI Realtime files is carried out using Server Workbench.

In the initial Server Workbench interface's Welcome overlay, select the Upgrade RPI Realtime tab:

Click the Upgrade button to display the Upgrade RPI Realtime API overlay:



The overlay contains the following:

- Choose Source Folder section:
  - Deployment Files Folder: mandatory.
- Choose Components to Upgrade section:
  - Checkboxes:
    - RPI Realtime API Web Site: checking displays the Web Site Folder property.
    - RPI Web Client: checking displays the Web Client Folder property.
    - RPI Realtime Agent Service: checking displays the Service Folder property.

All checkboxes are unchecked by default, and it is mandatory to check at least one checkbox.

- Web Site Folder property: this is displayed and enabled when the RPI Realtime website checkbox is checked. It is mandatory when displayed. It represents the physical address of the RPI Realtime website.
- Web Client Folder property: this is displayed and enabled when the RPI Web Client checkbox is checked. It is mandatory when displayed. It represents the full path of the rpiwebclient folder.
- Service Folder property: this is displayed and enabled when the RPI Realtime Agent Service checkbox is checked. It is mandatory when displayed. It represents the folder hosting the Realtime Agent service.
- Error Details section: this section is displayed when errors occur during upgrade, and documents the issues found. ○ Buttons:
  - Upgrade: if you click this button when one or more validation errors are present, a warning message is displayed. Invocation displays the Copy Log To Clipboard button and performs the upgrade. A message is displayed upon its successful completion.
  - Copy Log To Clipboard: clicking this button copies log entries generated during upgrade to the clipboard.
  - Cancel: invocation of this option removes the overlay from display

Please note the following:

- Errors will occur if you attempt to upgrade the RPI Realtime application prior to its having been installed.
- RPI Realtime will need to be upgraded separately at each cluster node exposing Realtime functionality.
- Note that, if there are any custom plugins that you wish to retain in the RPI Realtime service's bin folder following upgrade, please include within the same a text file called 'plugins.txt', in which are listed the name(s) of the .DLL(s) in question. Filenames should include the extension, and each should be placed on a separate line within the file. Failure to do so will see the plugin(s) deleted by the upgrade process.

Note About the Redpoint JavaScript Realtime Web Client

Documentation around the configuration and use of the JavaScript realtime web client to enable client-side integration between a web site and the Redpoint Realtime API can be found in Deployment Files in the following directory:

DeploymentFiles\InteractionRealtimeAPI\RPIFormValidation\shared\js\rpiwebclient\documentation\index.html

Realtime Status Endpoints

The following realtime methods can be used to determine the current status of the RPI Realtime installation:

- GET api/Status: returns the overall status of the RPI Realtime installation. The payload is returned as follows:


```
{
  "OverallStatus": true,
  "RealtimeAgent": {
    "InProcessAgentEnabled": false,
    "ServiceStatus": {
      "Status": true,
      "Message": null
    },
    "Databases": {
      "PulseDatabase": {
        "Status": true,
        "Message": null
      },
      "LoggingDatabase": {
        "Status": true,
        "Message": null
      },
      "ClientOperationalDatabase": null,
      "DataWarehouse": null,
      "AuxDatabases": null
    },
    "ClientAvailability": null
  },
  "Caches": {
    "NonVisitorData": {
      "Status": true,
      "Message": null
    },
    "VisitorProfile": {
      "Status": true,
      "Message": null
    },
    "VisitorHistory": {
      "Status": true,
      "Message": null
    },
    "ProductRecommendations": {
      "Status": true,
      "Message": null
```

```
    },
    "VisitorBackup": null
  },
  "Queues": {
    "ClientQueues": {
      "Status": true,
      "Message": null
    },
    "ListenerQueue": {
      "Status": true,
      "Message": null
    }
  }
}
```


- GET api/Status/{cacheName}: returns status details of a specific cache. The cacheName parameter should be set to the name of the cache as defined at the appsettings.json configuration file. The payload returned is contingent on the cache type; for example, 'database' returns the SQL required to create the cache.

Queue Setup

RPI uses queues in the following contexts:

- Collating realtime states and metrics
- Web form submission
- Managing data from the realtime cache
- Queue listeners
- Recommendation endpoint result collation

Queueing functionality can be provided by a range of Queue Providers, which are managed in the RPI application in a dedicated configuration interface. Please see the User Guide for more information. Note that a default MSMQ queue provider is automatically created at RPI installation.

A Copy JSON configuration to clipboard button is available at each queue provider. Clicking it copies to the clipboard a JSON snippet, structured as per the following example:

```
{"Assembly":"Redpoint.Resonance.AzureQueueAccess","Type":"Redpoint.Resonance.AzureQueu
eAccess.AzureServiceBusQueueFactory","Settings":[{"Key":"ConnectionString","Value":"En
dpoint=sb://Redpointinteractiondev.servicebus.windows.net/;SharedAccessKeyName=RootMan
ageSharedAccessKey;SharedAccessKey=EcWsxCsX+xOLQCdt0H6kScI1bR252V+WGljpou5MjwQ=","Valu
es":null}]}
```

This text must be copied into the ClientQueueSettings setting at the RPI Realtime appsettings.json file. Queue Listener queue details are copied into ListenerQueueSettings. To determine if configured correctly, browse to the RPI Realtime status page:

```
[RPI Realtime website]/status
```

Details of the specific queues required to support each of the above are provided separately.

Please note the following:

- When setting up MSMQ queue config settings, the full path of the queue must be provided (e.g. '.\Private$\RPIWebCacheData').
- If using Azure queue, set its lock duration to the maximum permissible value (5 minutes).

Web Events Setup

Web events are a mechanism used by RPI to effect communication between the RPI Realtime application and a website. They are used:

- To facilitate communication between RPI landing and other web pages and RPI.
- To capture states and metrics generated through visitors' website behaviors.
- To monitor goals and visitors' attainment of the same.

Web events adapters, which provide web events capabilities when associated with a landing page, dynamic or goal driven asset or a channel, are configured within a dedicated RPI configuration interface, details of which can be found in the RPI User Guide.

To configure web events for RPI, please follow these steps:

1. Create a new 'rpiwebevents' queue, using your queue provider of choice.
2. Ensure that the RPI Windows services user and web events website's application pool's identity are configured to have read/write access to the queue.
3. At the appsettings.json file, ensure the EventsQueuePath setting references the aforementioned queue
4. In the RPI System Configuration configuration interface, ensure WebEventQueuePath is set to the path of the queue.
5. In the RPI Web Adapters configuration interface, create and configure a Web Events adapter. Optionally describe any metrics and/or states to be tracked (see the RPI User Guide for more information).
6. In the Operations.System Tasks interface, ensure that the Web events importer system task is enabled.

Please refer to the following RPI User Guide sections for detailed information about the operational usage of web events.

1. Redpoint Interaction Realtime: covers JavaScript functions that can be called at web pages to pass back metric and state information.
2. Configuration Workbench: covers the setup of web events adapters, and attachment to channels to obtain state information at outbound activity execution.
3. Digital Asset Designer: covers use of web events adapters at the dynamic and goal driven assets.
4. Landing Page Designer: covers use of web events in landing pages.
5. Interaction Designer: describes collation the of metric and state information, using a web events adapter, at outbound channel execution.

Web Forms Setup

Web forms can be hosted in RPI landing pages and allow you to identify and elicit responses from site visitors. Form field data values are persisted and can be used in due course as the basis for further targeting.

To set up web forms, please follow these steps:

1. Create a new ‘rpiwebformsubmission’ queue, using your queue provider of choice.
2. Ensure that the RPI Windows services user and web events website’s application pool’s identity are configured to have read/write access to the queue.
3. At the appsettings.json file, ensure the FormQueuePath setting references the aforementioned queue.
4. In the RPI System Configuration configuration interface, ensure that FormSubmissionQueuePath is set to the path of the queue.
5. In the Operations.System Tasks interface, ensure that the Web form processor task is enabled.

Realtime Cache Setup

In order for RPI realtime decisions to be used, a caching mechanism must be made available, and configuration performed to ensure that the RPI Realtime application can make use of the same. The following caching technologies are supported:

- Memcached
- Amazon Elasticache
- Redis
- Windows Azure Redis Cache
- SAP HANA
- Couchbase
- Cassandra
- MongoDB
- SQL Server
- Database
- Ncache
- CosmosDB
- MemoryDB Redis

The following sections describe how to configure RPI Realtime to use each of these. Note that setup of each cache technology is beyond the scope of this documentation.

website

Memcached Configuration

This section describes how to configure the RPI Realtime to utilize the Memcached caching mechanism.

1. At the RPI Realtime appsettings.json file, ensure that a Memcached entry is added to the CacheSettings section, as per the following example:

```
{
"Name": "Memcached",
"Assembly": "Redpoint.Resonance.Memcached",
"Class": "Redpoint.Resonance.Memcached.MemcachedHandler",
"Settings": [
    {
    "Key": "SectionName",
    "Value": "memcached"
    }
    ]
}
```

2. In the web.config memcached element, configure the server addresses and ports.

3. If you wish to turn on Memcached logging, please uncomment the following (separate) sections in the same web.config file:

```
<!--Turn on for memcache logging
<sectionGroup name="enyim.com">
<section name="log" type="Enyim.Caching.Configuration.LoggerSection,
Enyim.Caching" />
</sectionGroup>-->

<!--Turn on for memcache logging
<appSettings>
<add key="Enyim.Caching.Diagnostics.LogPath" value="C:\Redpoint\Web
Processing\Logs\MemcacheLog.txt" />
</appSettings>
<enyim.com>
<log factory="Enyim.Caching.DiagnosticsLogFactory, Enyim.Caching" />
</enyim.com>-->
```

4. For further information on configuring Memcached, please see **https://github.com/enyim/EnyimMemcached/wiki**.

Amazon Elasticache Configuration

Configuration of Amazon Elasticache is carried out exactly as per Memcached configuration.

Redis Configuration

This section describes how to configure the RPI Realtime website to utilize the Redis caching mechanism.

1. In the RPI Realtime appsettings. json file's CacheSettings section, add the following:

```
{
"Name": "Redis",
"Assembly": "Redpoint.Resonance.RedisCache",
"Class": "Redpoint.Resonance.RedisCache.RedisCacheHandler",
"Settings": [
    {
    "Key": "IPAddress",
    "Value": "127.0.0.1"
    },
    {
    "Key": "DisableMGET",
    "Value": true
    }
    ]
}
```

Note the DisableMGET setting; this is used to toggle whether the cache provider will use multikey operations.

Windows Azure Redis Cache Configuration

This section describes how to configure the RPI Realtime Redis caching mechanism.

website to utilize the Windows Azure

1. In the RPI Realtime appsettings. json file's CacheSettings section, add the following:

```
{
"Name": "AzureRedis",
"Assembly": "Redpoint.Resonance.AzureRedisCache",
"Class": "Redpoint.Resonance.AzureRedisCache.AzureRedisCacheHandler",
"Settings": [
  {
    "Key": "ConnectionString",
    "Value": "_azure connection string_"

  },
  {
  "Key": "DisableMGET",
  "Value": true
  }
  ]
}
```

Note the DisableMGET setting; this is used to toggle whether the cache provider will use multikey operations.

SAP HANA Cache Configuration

This section describes how to configure the Realtime Agent website to utilize the SAP HANA caching mechanism to support RPI realtime decisions. Please follow the steps below.

1. Ensure the following assembly is present within the Realtime Agent bin directory:

a. Redpoint.Resonance.SAPHANACache.dll

2. In the Realtime Agent's web.config file, add the following entries:

```
<section name="sapHANACacheClient" type="
Redpoint.Resonance.SAPHANACache.Configuration.SAPHANACacheClientSection,
Redpoint.Resonance.SAPHANACache"/>
```

This entry must be added within the <configSections> section.

```
<setting name="WebCacheAssembly" serializeAs="String">
    <value>Redpoint.Resonance.SAPHANACache</value>
 </setting>
 <setting name="WebCacheClass" serializeAs="String">
    <value> Redpoint.Resonance.SAPHANACache.SAPHANACacheHandler</value>
</setting>
```

These entries must be added within the <Redpoint.Resonance.Web.Shared.Properties.Settings> section.

```
<sapHANACacheClient>
    <host server="[server]:[port]" schema="SYSTEM" username="SYSTEM"
password="Redpoint2016"/> </sapHANACacheClient>
```

This entry must be added within the <configuration> section.

Couchbase Cache Configuration

This section describes how to configure the RPI Realtime to utilize the Couchbase caching mechanism.

1. In the RPI Realtime appsettings. json file's CacheSettings section, add the following:

```
{
  "Name": "Default",
  "Assembly": "Redpoint.Resonance.CouchbaseCache",
  "Class": "Redpoint.Resonance.CouchbaseCache.CouchbaseCacheHandler",
  "Settings": [
    {
      "Key": "Server",
      "Value": "Server",
```

```
      "Values": null
    },
    {
      "Key": "Username",
      "Value": "Username",
      "Values": null
    },
    {
      "Key": "Password",
      "Value": "Password",
      "Values": null
    },
    {
      "Key": "BucketName",
      "Value": "BucketName",
      "Values": null
    }
  ]
}
```

Cassandra Cache Configuration

This section describes how to configure the RPI Realtime website to utilize the Cassandra caching mechanism.

1. In the RPI Realtime appsettings. json file's CacheSettings section, add the following:

```
{
  "Name": "Default",
  "Assembly": "Redpoint.Resonance.CassandraCachehe",
  "Class": "Redpoint.CassandraCache.CassandraCacheHandler",
  "Settings": [
    {
      "Key": "Keyspace",
      "Value": "rpi",
      "Values": null
    },
    {
      "Key": "ContactPoints",
      "Value": "server1,server2",
      "Values": null
    },
```

```
    {
      "Key": "Username",
      "Value": "CassandraUser",
      "Values": null
    },
    {
      "Key": "Password",
      "Value": "Password",
      "Values": null
    }
  ]
}
```

MongoDB Cache Configuration

This section describes how to configure the RPI Realtime website to utilize the MongoDB caching mechanism.

1. In the RPI Realtime appsettings. json file's CacheSettings section, add the following:

```
{
  "Name": "Default",
  "Assembly": "Redpoint.Resonance.MongoDBCache",
  "Class": "Redpoint.Resonance.MongoDBCache.MongoDBCacheHandler",
  "Settings": [
    {
      "Key": "Server",
      "Value": "Server"
    },
    {
      "Key": "Port",
      "Value": "Port"
    },
    {
      "Key": "Database",
      "Value": "Database"
    },
    {
      "Key": "CollectionName",
      "Value": "CollectionName"
    },
    {
    "Key": "IDasKeyEnabled",
    "Value": true
    }
  ]
}
```

Note the `IDasKeyEnabled` parameter, which, when enabled (the recommended setting), ensures records are written and retrieved from MongoDB using the native Mongo _id field instead of "Key" field.

2. The following Settings can be used to configure Partition Key functionality:
    - PartitionKeyEnabled (default value = false)
    - PartitionKeys (default values 'FieldName1' and 'FieldName2')
3. The following additional parameters are also supported at MongoDB cache:
    - Ssl

- ReplicaSet
- ConnectionString: e.g.:

```
{
"Key": "ConnectionString",
"Value": "mongodb://localhost:27017"
}
```

If a ConnectionString is provided, the Server, Port, Username, Password, ssl and ReplicaSet parameters are ignored. Provision of the Database parameter is mandatory, CollectionName and MessageHistoryCollectionName optional.

SQL Server Cache Configuration

When using the RPI SQL Server native cache provider, scripts to set up SQL Server in-memory cache tables are available in the following deployment files folder:

\DeploymentFiles\SQL Server\Useful Scripts\Realtime In Memory Cache Setup.sql

Notes are provided therein, which should be read before executing the scripts.

An example SQL Server cache configuration section from appsettings.json is provided below:

```
{
"Name": "InMemorySQL",
"Assembly": "RedPoint.Resonance.Web.Shared",
"Class": "RedPoint.Resonance.Web.Shared.Cache.SQLServerCache",
"Settings": [
{
"Key": "ConnectionString",
"Value": "Data Source=localhost;Initial Catalog=AdventureWorksDW;Integrated
Security=True"
}
]
}
```

NCache Cache Configuration

An example NCache cache configuration section from appsettings.json is provided below:

```
{
    "Name": "NCache",
    "Assembly":"RedPoint.Resonance.NCache",
    "Class":"RedPoint.Resonance.NCache.NCacheHandler",
    "Settings": [
        {
        "Key":"CacheID",
        "Value":"<cache-name>"
        },
        {
        "Key":"ServerList",
        "Value":"<server1-ip>,<server2-ip>,..."
        }
```

```
        ]
}
```

CosmosDB Cache Configuration

An example CosmosDB cache configuration section from appsettings.json is provided below:

```
{
  "Name": "AzureCosmosDB",
  "Assembly": "RedPoint.Resonance.DocumentDBCache",
  "Class": "RedPoint.Resonance.DocumentDBCache.DocumentDBCacheHandler",
  "Settings": [
    {
      "Key": "DatabaseId",
      "Value": "<DATABASE ID HERE>"
    },
    {
      "Key": "ConnectionString",
      "Value": "<CONNECTIONSTRING HERE>"
    },
    {
      "Key": "ContainerName",
      "Value": "<NAME>"
    }
  ]
}
```

The ContainerName setting will be used as the table name to be created in CosmosDB. Its 'id' column will be used as the PartitionKey (indicated by a prefix '/' in the Azure portal).

MemoryDB Redis Cache Configuration

An example MemoryDB Redis cache configuration section from appsettings.json is provided below:

```
{
  "Name": "MemoryDB",
  "Assembly": "RedPoint.Resonance.MemoryDBRedisCache",
  "Class": "RedPoint.Resonance.MemoryDBRedisCache.MemoryDBRedisCacheHandler",
  "Settings": [
    {
      "Key": "Host",
      "Value": ""
    },
    {
      "Key": "Port",
      "Value": ""
    },
    {
      "Key": "ssl",
      "Value": ""
    },
    {
      "Key": "UserName",
      "Value": ""
    },
```

```
    {
      "Key": "Password",
      "Value": ""
    }
  ]
}
```

Using a Database as the Cache

Rather than using an in-memory caching technology, you can also use a database to serve as the RPI Realtime cache. An obvious advantage of this approach is the inherent resilience provided by a database technology; a disadvantage is that performance will likely be compromised when compared to an in-memory cache. When data resiliency is not an absolute requirement, an in-memory cache is the recommended option.

To use a database as a cache, it must be defined in RPI's appSettings.json configuration file. The following snippet defines two database caches – 'DataWarehouse' and 'Cassandra':

```
    "Caches": [
      {
        "Name": "DataWarehouse",
        "Assembly": "Redpoint.Resonance.Web.Shared",
        "Class": "Redpoint.Resonance.Web.Shared.Cache.DatabaseCache",
        "Settings": [
          {
            "Key": "ClientID",
            "Value": "D2A8B022-E87E-45E4-B9CD-3FC5809FD9C8"
          }
        ]
      },
      {
        "Name": "Cassandra",
        "Assembly": "Redpoint.Resonance.Web.Shared",
        "Class": "Redpoint.Resonance.Web.Shared.Cache.DatabaseCache",
        "Settings": [
          {
            "Key": "ClientID",
            "Value": "D2A8B022-E87E-45E4-B9CD-3FC5809FD9C8"
          },
          {
            "Key": "DatabaseID",
            "Value": "45E3C456-B8D4-4EA0-9496-BD24AA4FD842"
          }
        ]
      }
    ]
```

The database caches thus configured can subsequently be leveraged at the same file's DataMaps section:

```
  "DataMaps": [
      {
        "Type": "Visitor Profile",
        "Cache": "Cassandra",
```

```
          "DaysToPersist": 28,
          "CompressData": "True"
        },
        {
          "Type": "Visitor History",
          "Cache": " Cassandra ",
"DaysToPersist": 28,
          "CompressData": " True"
        },
        {
          "Type": "Non Visitor Data",
          "Cache": "DataWarehouse",
          "DaysToPersist": 365,
          "CompressData": " True"
        },
        {
          "Type": "Product Recommendations",
          "Cache": "DataWarehouse",
          "DaysToPersist": 365,
          "CompressData": " True"
        }
      ]
```

In this example, the Cassandra cache is being used to persist visitor profile and history data; the Data Warehouse cache holds non-visitor data (such as published realtime decisions) and product recommendation data.

The following tables are required at a database that is intended to be used as a cache:

- rpi_cacheobjects: this table stores details of all non-visitor data (e.g. landing pages, realtime decisions). It contains the following columns: ○ cachekey ○ cachevalue ○ expirydate
- rpi_cachevisitors: this table stores details of realtime visitors. It contains the following columns: ○ cachekey: key column ○ profile ○ parametervalues ○ databasevalues ○ history ○ expirydate
- rpi_cacheincrements: stores counters; used at content display ramp-up, inbound message lists (capping the maximum number of messages served). Note that, when using database providers there is a risk of contention when using the database to manage cache increments. It is therefore not recommended to use database providers for non-visitor data.

Table creation scripts are available at the following URL:

```
[RPI Realtime website]/status/{CacheName}
```

(where {CacheName} corresponds to the Name property for the relevant entry defined in the “CacheSettings” section of the appsettings.json.config file)

Note that scripts vary by database provider.

Realtime Helper Utility

The following command line application is provided in the RPI deployment files:

`DeploymentFiles\Utilities\RedpointRealtime\Redpoint.RealtimeHelper.exe`

It can be used to perform a number of useful tasks to assist in the ongoing management of an RPI Realtime installation:

- Decompress a visitor profile:

  `redpoint.realtimehelper profile <profile to decompress>`

  `redpoint.realtimehelper profile -db <e.g. mongodb,cosmosdb>  <profile to decompress>`

- Encrypt an appsettings password setting: `redpoint.realtimehelper`

  `appsetting <setting to hash>`

- Encrypt a visitor ID: `redpoint.realtimehelper visitorid encrypt`

  `<visitor id to encrypt>` • Decrypt a visitor ID:

  `redpoint.realtimehelper visitorid decrypt <visitor id to encrypt>`

Realtime Decisions Setup

RPI's realtime decisions allow you vary web page content in real time, in accordance with, e.g.:

- Site visitor characteristics such as region or browser type.
- A visitor's prior receipt of an RPI outbound offer.
- An identified visitor's cached attribute values, as sourced from the RPI data warehouse.

Realtime decisions utilize a (typically in-memory) realtime cache (supported caching technologies are Memcached, Redis, Windows Azure Redis Cache, SAP HANA, Amazon Elasticache, Database, Cassandra, Couchbase, MongoDB, Ncache, CosmosDB and MemoryDB Redis).

The following setup steps need to be undertaken to configure an RPI server to use realtime decisions:

1. Ensure that the caching service is running. Please refer to the appropriate section below.

2. Set Boolean client system configuration setting EnableRPIRealtimeServices to True.

Note that it is strongly recommended that a failover mechanism be implemented to support the realtime cache. In its absence, should the caching service stop, all cached visitor profile data will be lost. On the service's restart, any cached decision logic will be reloaded automatically, but any profile data will need to be built up again. Setup of a failover mechanism is beyond the scope of this documentation.

Realtime Agent Setup

The RPI Realtime Agent service allows RPI Realtime to communicate with the Data Warehouse and Operational databases.

Please follow these steps to set up the Realtime Agent.

1. Copy the InteractionRealtimeAgent folder, which can be found in the RPI DeploymentFiles folder.
2. Paste it into the RPI web application folder (typically C:\inetpub\wwwrpi).
3. Launch Internet Information Services (IIS).
4. In IIS, select the RPIWebSite website, and Add Application.
5. Configure the properties as follows:
    - Alias: InteractionRealtimeAgent
    - Application pool: RPIAppPool
    - Select the Physical path of the newly-pasted folder
6. Having configured the application's properties, browse to the newly-added application's Realtime Agent service, and ensure that a page describing the service is shown.
7. Copy the service's URL to the clipboard.
8. Open web.config at the RPI Realtime website.
9. Locate the RealtimeAgentAddress setting and set its value to the URL copied previously.
10. Save the web.config changes.
11. Open web.config for the InteractionRealtimeAgent folder.
12. Ensure that the following database connection strings are configured correctly:

```
<connectionStrings>
    <add name="OperationalDatabase"
connectionString="Server=localhost;Database=Pulse;Integrated Security=true"
providerName="Microsoft.Data.SqlClient" />
    <add name="LoggingDatabase" connectionString="Data Source=localhost;Initial
Catalog=Pulse_Logging;Integrated                                    Security=True"
providerName="Microsoft.Data.SqlClient" /> </connectionStrings>
```

If you wish to take advantage of .NET's automatic connection retries feature, please see the following article: **https://msdn.microsoft.com/enus/library/system.data.sqlclient.sqlconnectionstringbuilder.connectretrycount(v=vs.11 0).aspx**

13. Save the web.config changes.

The Realtime Agent will now be ready for use.

The Realtime Agent can be used in-process. To configure it for use in this way, follow these steps:

- Set the RealtimeAgentInProcessEnabled setting in appsettings.json to True.
- Copy the Realtime Agent deployment files' bin folder into the RPI Realtime bin folder.
- Set the following RPI Realtime web.config setting as follows:

```
<probing privatePath="bin\[RT Agent bin folder name]" />
```

- Add the following connection strings to the existing RPI Realtime web config `<connectionStrings>` node:

```
<add name="OperationalDatabase"
connectionString="Server=localhost;Database=Pulse;Integrated Security=true"
providerName="Microsoft.Data.SqlClient" />

    <add name="LoggingDatabase" connectionString="Data Source=localhost;Initial
Catalog=Pulse_Logging;Integrated Security=True"
providerName="Microsoft.Data.SqlClient" />
```

Note that Realtime Agent web.config settings are also available in the RPI Realtime web.config file under RedPoint.Interaction.RealtimeAgent.Properties.Settings.

Geolocation Setup

Three Geolocation providers are supported by RPI:

- Bing
- Azure
- Google Maps Services

To use Geolocation functionality in RPI realtime decisions with Bing or Azure, you will need a Map API key. This will need to be obtained from the respective provider, and will need to be referenced at the RPI Realtime appsettings file's MapAPIKey setting.

In addition, the Geolocation plugins section in the AppSettings file will need to reference the provider of choice, as per this example:

```
"GeolocationSettings": {
      "Provider": "Google Maps",
      "APIKey": "<API KEY>",
      "PluginAssembly": "RedPoint.Resonance.RealtimePlugins",
      "PluginType": "RedPoint.Resonance.RealtimePlugins.GoogleMapsPlugin.GoogleMapsPlugin",
      "WeatherUnits": null,
      "CoordsCacheTimespanMinutes": 1,
      "PluginSettings": []
    }
```

Load Web Cache Data Task Setup

The Load web cache system task (disabled by default) is responsible for loading cached visitor data into a series of dedicated data warehouse tables, and/or to a log file. To set the task up, please follow these steps:

1. Create a new 'RPIWebCacheData' queue, using your queue provider of choice.
2. Ensure that the RPI Windows services user and web events website's application pool's identity are configured to have read/write access to the queue.
3. At the RPI Realtime appsettings.json file, ensure the CacheOutputQueuePath setting references the aforementioned queue.
4. Ensure that at least one of the following system configuration settings is set to True:
    . WebCacheOutputToDatabase
    . WebCacheOutputToFile
5. In the RPI System Configuration configuration interface, ensure RPIWebCacheData is set to the path of the queue.
6. Ensure that appsettings.json setting CacheOutputQueueEnabled is set to True.

Web Publish Sites in a Load Balanced Environment

When publishing landing pages to IIS in a load balanced environment, it is recommended that Centralized Web Farm Management is used. This allows content that is published to a given share to be mirrored across a number of web servers. For more information, please see **http://www.iis.net/overview/control/centralizedwebfarmmanagement**.

It is recommended that the share used in this context not be hosted at one of the public-facing web servers.

Security Concerns Using Sequential Keys

If the Interaction Realtime API is publicly accessible and landing pages or dynamic content make use of personalization based on attributes, care should be taken when choosing a Master Key or alternative key that is sequential or easy to guess. In this case, it would be possible for an attacker to pull the same attribute value data value back for other visitors whose profile data is stored in the cache by iterating through possible keys.

Deploying the RPI Realtime API to Azure App Service

This section details a recommended approach to deploying the Redpoint Realtime API to Azure App Service.

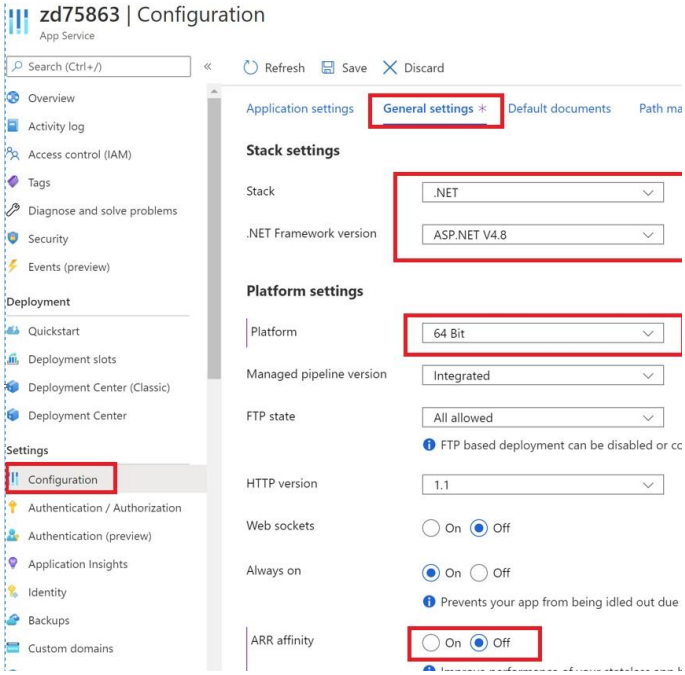
The procedure assumes that the App Service base resources have already been provisioned and that networking requirements for Realtime running on App Service to access the Realtime message bus, Realtime cache, and Realtime Agent are in place. As of the current Release of RPI at the time of writing this article the App Service resource must be the Windows Server option targeted to .NET Framework.

Please follow the steps below to deploy the RPI Realtime API to Azure App Service.

1. Validate stack and settings. At Configuration > General settings, ensure the App Service instance:

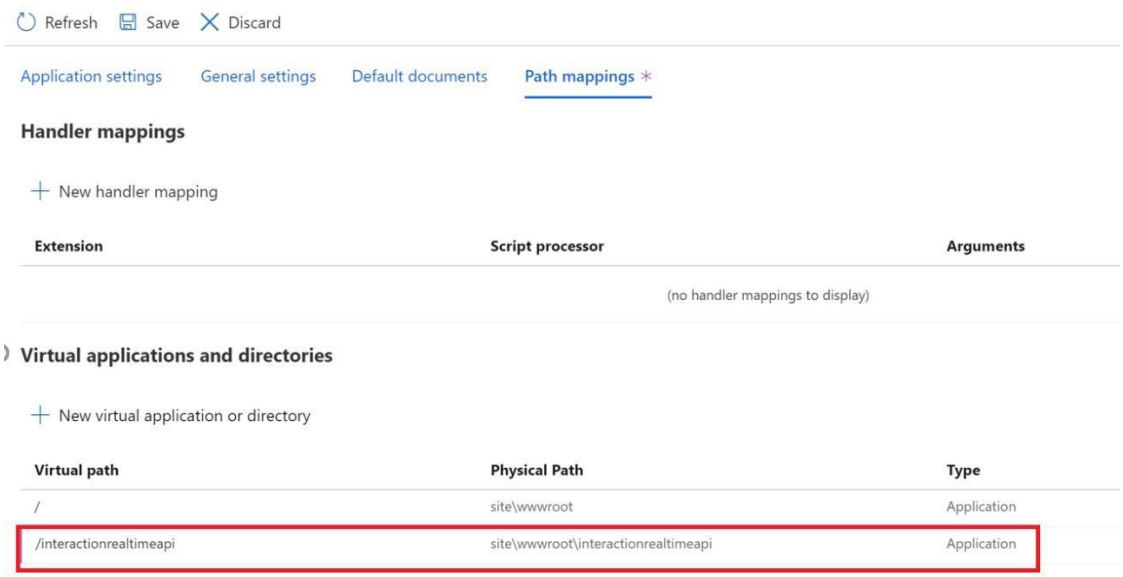
   Is targeted to .NET Framework v4.8

   Is set to 64bit
   Has AAR affinity turned off

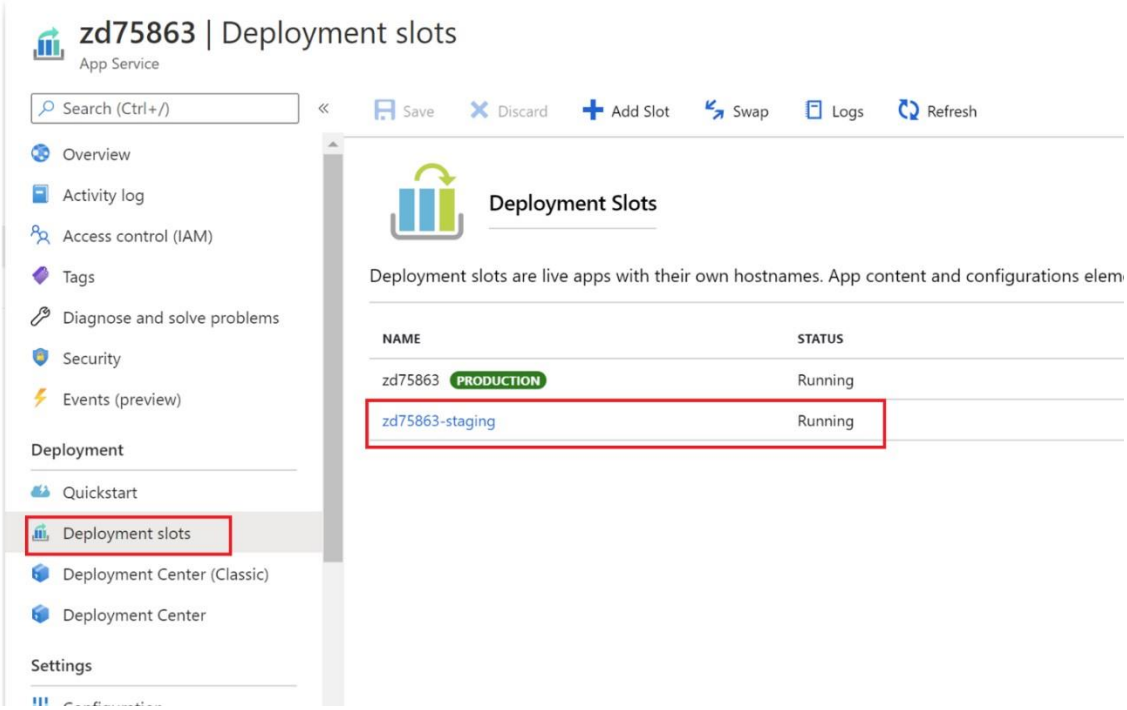


2. Set up virtual directory. If the realtime service is to be deployed in a virtual directory, then from Configuration > Path mappings, create a virtual to physical path mapping and make sure it is an application (i.e. the Directory option should be un-checked). It's

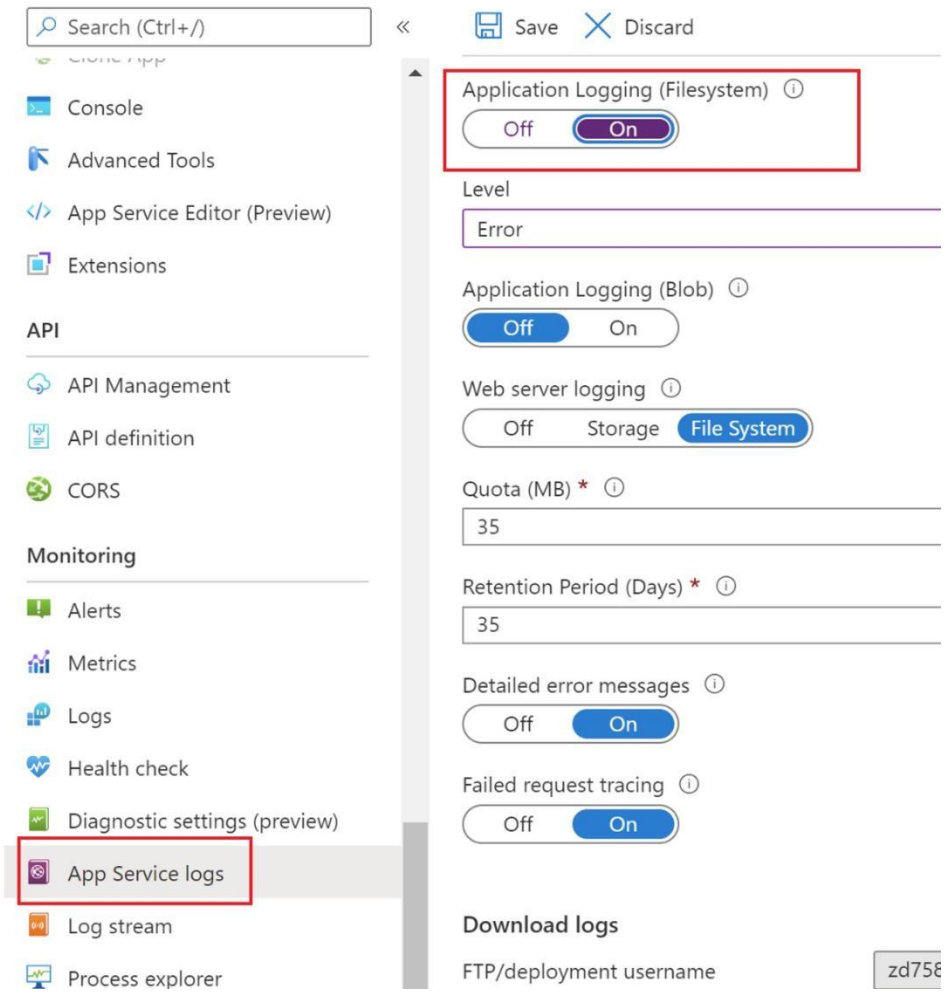
   strongly recommended to use interactionRealtimeAPI as the virtual directory name to match the folder in Deployment Files.



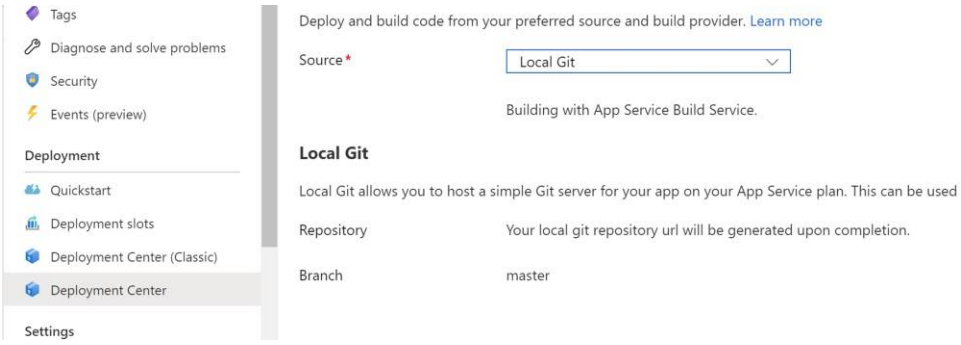
3. Create a staging deployment slot. It is recommended to create a staging deployment slot where changes can be applied and tested prior to production deployment. This can be added from Deployment > Deployment Slots



4. Turn on application logging. At Monitoring > App Service Logs set Application Logging (Filesystem) on and the level to Error



5. Set deployment source. At Deployment Center > Settings, set the source of your staging deployment slot to Local Git or Azure Repo:



6. Clone the repository, copy in Realtime API files, set logging directory. Using the Git client of your choice, clone the repository you just created to a local repository and copy in the Realtime API deployment files. Assuming the virtual directory created above was named InteractionRealtimeAPI, simply copy in the InteractionRealtimeAPI folder from deployment. If you named the virtual directory and path mapping differently, then rename the folder accordingly.

Make any necessary configurations in appsettings.json.config. Configuration settings of queue providers, Realtime caches and the Realtime agent within the appsettings.json.config file is out of scope of this document. Please refer to the RPI System Administration Guide for instructions on how to update this file. The one setting that we'll cover here specific to App Service is the logging directory. This should be modified and set to path "D:\\Home\\LogFiles\\Application". As of RPI v6.2, you can specify a [MachineName] variable in the file name to capture the hostname of the machine RPI is running on. v6.1:

```
},
"FileLogging": {
  "LogDirectory": "D:\\Home\\LogFiles\\Application",
  "FileName": "EntryLog_RPIRealtime_v61_"
},
```

v6.2 and above:

```
"FileLogging": {
  "LogDirectory": "D:\\Home\\LogFiles\\Application",
  "FileName": "EntryLog_RPIRealtime_[MachineName]_ver62_"
},
```

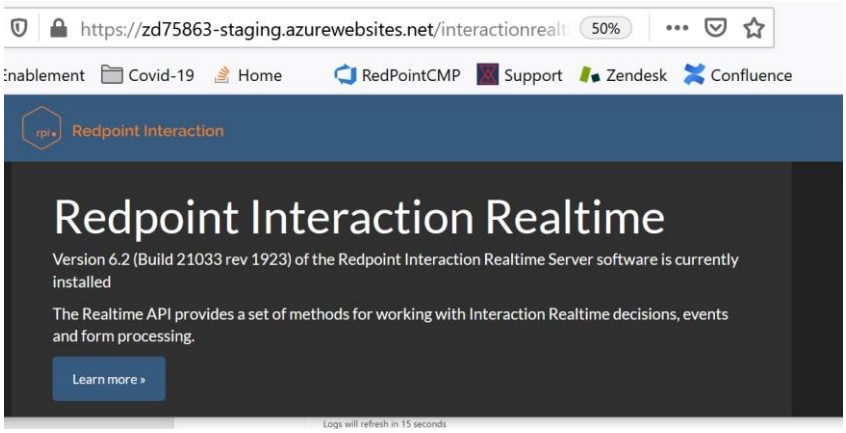
Links to Git Resources:

Git for windows
https://git-scm.com/download/win

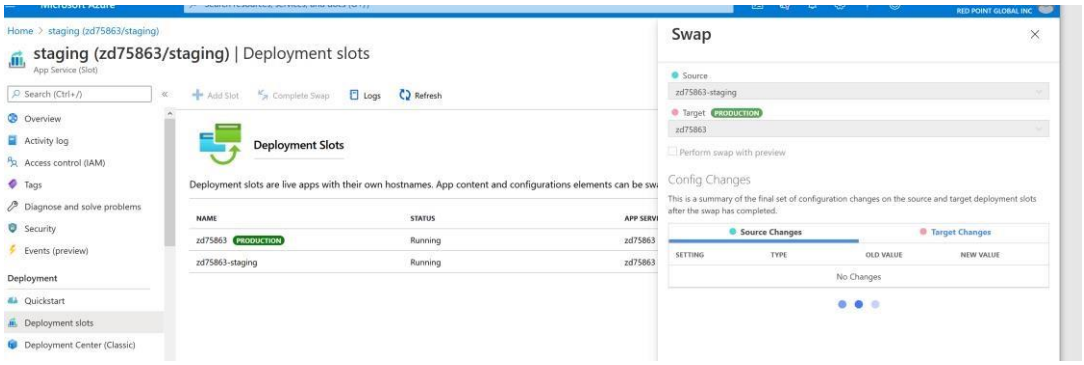
Source tree - git UI for Win & Mac https://www.sourcetreeapp.com/

7. Perform initial commit and validate app loads. Ensure your Git client is not set to ignore DLLs and then make the initial commit. Once the commit is complete, you can confirm the operation from Azure portal Deployment Center logs:

8. Validate staging deployment. Load the staging URL into your browser and confirm the RPI Realtime splash page loads as expected and perform any additional validation as needed



9. Swap staging into production. Once validations are complete, initiate a swap from the staging to the production slot.



10. Validate new production deployment. Load the production URL into a browser and confirm the RPI Realtime splash page loads as expected and perform any additional validation as needed.

RPI to RPDM Integration

Integration between RPI and Redpoint Data Management (RPDM) is required for several RPDMdependent features in RPI:

- Data Intake (Data Projects): functionality that supports importing flat files into the RPI Data Warehouse
- Data Process Projects: functionality that supports running Data Management projects during the execution of an interaction workflow and/or interaction audience to augment RPI capabilities (e.g. couponing).
- Goal Driven Assets with Machine Learning: functionality that supports using RPDM's Machine Learning capabilities to optimize Goal Driven Assets.
- Channel States Import: functionality that leverages RPDM to facilitate import of channel execution fulfillment states results; mostly used in environments where high volume of results are generated.

A number of RPI client system configuration settings (within the Environment group) are used to manage the integration between the products. The following examples detail how these should be configured to achieve three different integration options.

1. Integration using RPDM RPBase.dll: to connect to RPDM using the RPBase.dll libraries and any available Execution Server that the Site Server provides, the following system configuration parameters should be set:
   - DataManagementCredentials: configure with the credentials for connecting to RPDM.
   - DataManagementExecutionServer: leave blank.
   - DataManagementIntegrationType: set to 0.
   - DataManagementServerName: set to the RPDM Site Server hostname.
   - DataManagementServerPort: set to the Site Server port (normally 20400 for RPDM7, 20410 for RPDM8).
2. Integration using RPDM RPBase.dll and a named Execution server RPI: assuming integration with a multi-node DM environment, this option allows the system to designate RPI-initiated DM jobs to run on a specific execution server The DM system parameters should be configured as above, then additionally configure the name of the desired execution server at the DataManagementExecutionServer system configuration setting.
3. Integration using Data Management Operational API RPI: supported at RPDM v7+. To configure RPI to run DM jobs using the DM Operational API web service, begin with the configuration from the second option above, and additionally:
   - DataManagementIntegrationType: change from 0 to 1.
   - DataManagementOAPIWebServiceAddress: configure with point to the base URL for the OAPI web service.
   - All other settings are still required.

Data Intake Setup

Please follow these steps to set up RPI's Data Intake functionality, which allows you to load files into your data warehouse using the Data Project Designer.

Before commencing setup of Data Intake, please ensure that the following pre-requisites are in place:

1. RPI is installed.
2. Redpoint Data Management v6 or later is installed.

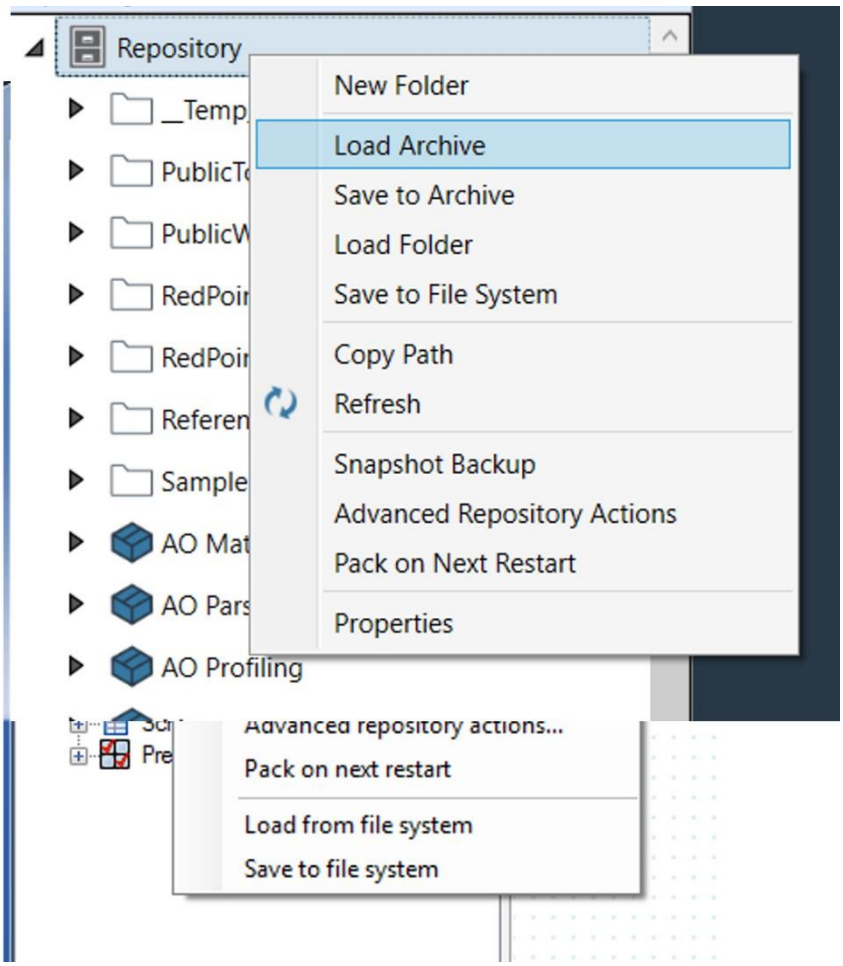
Follow these steps to install Data Intake:

Data Intake Share Folder

1. Create a folder named DataIntake where files will be loaded and processed. 'C:\Program Files\Redpoint Global\RPI\' is a reasonable place to create this, but it may be created anywhere.
2. Create four sub-folders inside the newly created folder
   . Processing
   . Archive
   . Upload
   . Reference
3. Open the DataIntake folder's Properties dialog, and navigate to the Sharing tab. Ensure that the RPI Execution service user has read and write access to this folder and all of its sub-folders.
4. If the above folder and share do not reside on the same physical or virtual as the RPI server, update the following configuration values in RPI: a. DataIntakeArchiveDirectory
   . DataIntakeLandingDirectory
   . DataIntakeProcessingDirectory
   . DataIntakeProcessingDirectoryDataManagement

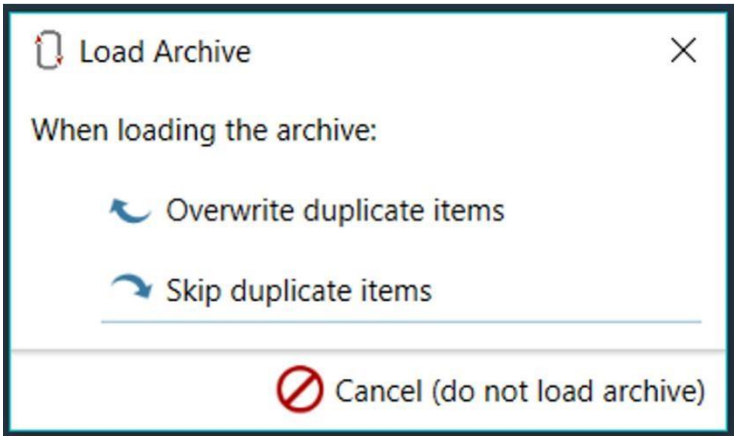
Data Management Configuration

1. Macros
   a. Connect to Redpoint DataManagement via your RedpointDM.exe client.
   b. On the left side are the Palette and Repository tabs, select the Repository tab. Right click on repository and select load from archive.



c. Import Redpoint_macros_object.dlb from the source folder (Example: C:\source\Resonance\Database Scripts\DataManagement Macros).

i. If the following dialog is displayed, please select Skip duplicate items.



d. Validate that a folder named “RedpointMacros” now exists in the repository.

2. Database Connections
   . Browse in the repository to data connections.
   . Right click on data connections – select “New connection”.
   . Create 2 new data connections:
     i. RPI_AuditDB – this connection will connect to the interaction audit database. ii. RPI_MainDB – this connection will connect to the main data warehouse

     Note that the data connections should be in sync with the respective connections in the Server Workbench Clients interface.

   . Configure the above data connections per the specifics of your environment, with users who have DB owner level permissions.
   . If you are configuring a multi-tenant environment on the same data management instance, an additional pair of connections will be required for each tenant. The names of the connections must start with “RPI_”. We recommend RPI_AuditDB_[Tenant Name] and RPI_MainDB_[Tenant Name]. In this case, the AuditDatabaseSourceName and MainDatabaseSourceName configuration values in RPI must also be configured.
3. Please set the following system configuration settings:
   . AuditDatabaseSchema: the schema to use for the processing engine to write back logging information.
   . AuditDataSourceName: the connection string to use for the processing engine to write back logging information.
   . MainDatabaseSchema: the Data Intake data warehouse schema.
   . MainDatabaseSourceName: the Data Intake data warehouse connection string.
   . DataManagementCredentials: the credentials used to sign in to Data Management.
   . DataManagementExecutionServer: Optional setting to force RPI triggered data management jobs to run on a specific execution server.
   . DataManagementIntegrationType: the method to be used in communicating with Data Management. Set 0 to designate legacy method, set 1 to use the Operational API web service.
   . DataManagementOAPIWebServiceAddress: the configured URL to connect to the OAPI web service.
   . DataManagementServerName: the RPDM Site server hostname to be used.
   . DataManagementServerPort: the Port number used to connect in Data Management (normally 20400 for DM7, 20410 for DM8).

RPI - AML Setup

RPI uses Redpoint Automated Machine Learning (AML) to execute predictive analytic functionality (used in contexts such as Model Projects and Clustered Audiences).

To use AML with RPI, the following client configuration settings much be configured:

- AutomatedMLAPIAddress: the AML web service address
- AutomatedMLCredentials: the credentials used to access AML

Active Directory Setup

RPI can be configured to facilitate login using both Azure Active Directory (AD) and Active Directory Federation Services (ADFS).

If planning to use Azure AD or ADFS functionality, please be aware of the following:

- RPI links AD users to their RPI equivalents via email address.
- In both contexts, the email address is sourced from the following claim: 'http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress'
- The integration is only for authentication purposes. Individual users, with appropriate, linking, email addresses, need to be created in RPI.
- User groups and permissions must similarly be configured manually in RPI.

To use Azure Active Directory, the following cluster-level settings need to have be provided:

- EnableAzureAD
- AzureADAADInstance
- AzureADAudience
- AzureADClientID
- AzureADInteractionResourceID
- AzureADRedirectUri
- AzureADTenant

To use ADFS, the following cluster-level settings need to have be provided:

- EnableADFS
- AFDSAADInstance
- ADFSAudience
- ADFSClientID
- ADFSInteractionResourceID
- ADFSMetadataEndpoint
- ADFSRealm
- ADFSRedirectUri
- ADFSTenant
- ADFSValidateAudience
- ADFSValidateIssuerSigningKey

On changing any of the above settings:

- All users currently connected to the RPI server using ADFS or Azure AD need to log out.
- The RPI website must be stopped and started in IIS.

- To access the changed settings, users must click the Retrieve settings… button in the RPI Login dialog.

OpenID Connect

RPI facilitates use of an OpenID Connect provider to provide user authentication. Verified providers are:

- Auth0
- KeyCloak
- Okta
- Ping Identity
- Gigya

Full details on setting up RPI to use OpenID Connect can be found in a separate OpenID Configuration document.

Queue Listener Setup

Queue listeners facilitate the monitoring of 'listener queue' for the arrival of data. Data arrives in the form of JSON packages – placed on the queue either by an external system, or at submission of a web form. Downstream queue activities can then use this data to execute offers. Queue listeners might typically be used for the sending of emails e.g. after a customer makes a purchase, or when a web form is submitted in a landing page.

More details on queue listeners can be found in the RPI User Guide.

Please follow these steps to configure RPI to use queue listeners:

1. Configure the listener queue provider in the Server Workbench Queue Providers interface.
2. Ensure the following cluster configuration settings are set:
    . ListenerQueueIsEnabled: set to True
    . ListenerQueuePath: set to path of queue to be used as listener queue.
3. At the RPI Realtime appsettings.json file, set the ListenerQueuePath setting value to the same value as per the previous step.
4. In the Server Workbench Queue Providers interface, copy the listener queue's JSON configuration to the clipboard.
5. Paste the same into the RPI Realtime appsettings.json file's ListenerQueueSettings section.

Azure KeyVault

Microsoft Azure KeyVault can be used to securely persist credentials used by RPI. KeyVault can be enabled by specifying a value in the KeyVaultConfig field in Pulse database table RPI_Cluster. As an example:

```
{
    "assembly": "RedPoint.Resonance.AzureKeyVault",
    "class": "RedPoint.Resonance.AzureKeyVault.AzureKeyVaultClient",
    "settings": {
        "uri": "https://xxx.vault.azure.net/"
    }
}
```

Load Balancer Setup

When installing an RPI cluster behind a load balancer (which is required), please note the following:

- A machine key must be generated and added to the Interaction API's web.config:

```
<system.web>    <machineKey
    validationKey="....."
decryptionKey="....."
  />
```

- This value will need to be applied at the Interaction API web.config of each machine behind the load balancer. The value will need to be updated manually at each upgrade.
- The load balancer needs sticky sessions enabled.
- A wildcard certificate or SAN certificate needs to be installed on all RPI nodes and the load balancer.

Scheduler Rules

Scheduler rules can be applied to optionally limit the types of task that a cluster node can execute. Scheduler rules are applied by updating the SchedulerRules database field, which is located in the rpi_ClusterNodes table in the Pulse operational database. They are configured by updating the field value with a JSON document, as per the following example:

```
{
"JobRules" :
[
  {
  "Type": "Task",
  "JobNames": ["Web events importer", "Web cache data importer"]
  }
],
"Clients": ["A31519BE-48F0-452C-9A8B-64E76E199BAA"]
}
```

The document's properties are as follows:

- Type: as per rpi_ExecutionSchedule.Type
- JobNames: this property can be used to further qualify Type. If Type is set to 'Task', the property can be used to specify specific task name(s), as per rpi_Tasks.TaskName. Else the value specified should be set as per rpi_ExecutionSchedule.Name.
- Clients: as per rpi_Clients.ClientID

If scheduler rules are applied, a node will only be able to execute work in accordance with the specified restrictions. If a node in a multi-node cluster is configured with scheduler rules, it will be given priority to execute the work to which limited.

Client Migration Scripts

Scripts that facilitate the migration of an RPI client to another, new client, are provided in the DeploymentFiles\SQL Server\Useful Scripts\Client Migration Scripts folder. These facilitate the migration of an RPI client – e.g. ‘Test’, into a new, e.g. ‘Production’ client.

Please see the Migration_Script_Steps text file in the same place, which describes the steps to be undertaken to migrate a client.

Database-Specific Configuration Settings

The following configuration settings need to be considered when running RPI against specific database providers. Some settings pertain to specific providers, as evidenced in their naming; others apply across providers.

- AWSPerformCOPYFromS3
- AWSRedshiftDistributionStyle
- AWSRetainCOPYGeneratedFilesInS3
- AWSSourceS3BucketName: set as per the following: [AWS S3 ECP Name]\[Bucket name]
- AzureSQLDWDistributionType
- AzureSQLDWDropStatisticsOnTempTables
- DatabaseInsertBatchLimit: applies at the following database providers.
    - Azure SQL ○ DB2 ○ Google Spanner ○ MariaDB ○ SQL Server
- DatabaseSpecificRetryableErrorEnabled
- DB2RunStatsEnabled
- DisableDatabaseObjectQualification: if set to True, when SQL is generated, database objects will not be qualified. Applies at the following database providers: ○ AWS RedShift ○ AWS RedShift Spectrum ○ Azure PostgreSQL ○ DB2 ○ Google Spanner ○ Greenplum ○ Netezza ○ Oracle ○ PostgreSQL
- GoogleBigQueryUseInformationSchema
- OracleAudienceTempTablesIndexDisabled
- OracleDegreeOfParallelism
- OracleDMLSessionParallelismEnabled
- OracleOptimizerDynamicSamplingLevel
- OracleRunTableStats
- OracleTempTablesParallelismEnabled
- OracleTempTablesPKConstraintDisabled
- OracleUseDirectPathInsert
- OracleUseHashEnabled
- SQLServerDisableRowLockHint
- SQLServerIndexTempTables
- SQLServerUseColumnStoreIndexTempTables: if set to true, a CLUSTERED COLUMNSTORE index will be created at RPI-generated temporary audience tables. The index will not be created if a temporary table contains one or more columns with the following data types:
    - text ○ ntext ○ xml ○ image ○ rowversion ○ sql_variant ○ CLR types (hierarchyid and spatial types) ○ nvarchar(max), varchar(max) and varbinary(max) (SQL Server versions not later than 13.x)
    - uniqueidentifier (SQL Server versions not later than 11.x)

- SQLServerUseCountBig
- SQLServerUseCreateTableInsert
- SQLServerUseIdentityTempTables
- SQLServerUseSelectIntoOverUpdate
- SQLServerUseTempFlagTables
- TeradataIndexTempTables
- TeradataUsePrimaryIndex
- TeradataUseTableCharacterSet
- YellowbrickDistributionType

Troubleshooting

All server side issues are logged in the Log table in the "Pulse_Logging" database. This can be accessed via the RPI Operations Interface, which should always be the first place to look in the event of an issue occurring.

The "ProcessName" field is useful in determining whether a reported error is related to the issue you are investigating. For example, if an interaction is failing, the process to look for would be Execution service. Any errors from other processes may hint at related issues but will not be the direct cause of the problem.

If the error is interaction- or audience-related, then the logs of those activities in the Interaction or Audience designer interface will also give useful debugging information.

If the error is related to unexpected counts from audiences or offer activities, the SQL trace will provide useful information to explain the cause. From v1.5, the SQL trace is available within the Operations Interface.

When an exception is reported in the client user interface, there will most likely be a log entry in the client local log file—usually in "C:\temp\RPI\trace.log".

Always study the whole stack trace for an issue since often an error is reported at several layers within the code and may be reported differently on the server and the client.

More troubleshooting tips, guides and other resources can be found online on Redpoint's support portal at **https://support.Redpointglobal.com**.

You may also log a service request with our support team for assistance from the same portal.

In addition to this, you may be able to obtain troubleshooting guidance and suggestions from Redpoint power users on the community forums.

How to…

The following section provides useful guidance on how to carry out a range of common operations tasks.

Change the Database Connection Timeout

The DatawarehouseTimeoutMinutes field, in the rpi_Clients table in the Pulse operational database, can be used to control the data warehouse connection timeout. It defaults to the value 0. The field must be changed through use of a SQL script - its update via Server Workbench is not supported.

Change the Database Connection Details

Pulse and Pulse_Logging operational database connection details are configured in the server .config files.

These are:

- Interaction API: web.config
- Resonance.NodeManagerService.exe.config
- Resonance.ExecutionService.exe.config

In all cases, the connection strings will be listed as per the example:

```
<connectionStrings>
  <add
    name="LoggingDatabase"      connectionString="Server=localhost;Database=Pulse_Logging;Integrated Security=true"
    providerName="System.Data.SqlClient" />
  <add
    name="OperationalDatabase"
    connectionString="Server=localhost;Database=Pulse;Integrated  Security=true"                providerName=""  />
</connectionStrings>
```

If you wish to take advantage of .NET's automatic connection retries feature, please see the following article:

https://msdn.microsoft.com/en-us/library/system.data.sqlclient.sqlconnectionstringbuilder.connectretrycount(v=vs.110).aspx

Window services will require a restart once a configuration file has been changed.

System Maintenance

This section describes system maintenance tasks to be undertaken to ensure an RPI's optimal health and performance.

Backup

To prevent data loss in the broadest range of failure scenarios, regularly back up the most important files/folders/databases

1. In SQL Server, for all databases, it is recommended that you utilize the Full Recovery Model: http://msdn.microsoft.com/en-us/library/ms190217(v=sql.105).aspx:

    . <ClientDataWarehouse>

- Pulse
- Pulse_Logging
- Interaction_xxx databases
- InteractionAudit_xxx databases

2. The important files and folders are as follows:
    - C:\inetpub\wwwrpi*
    - C:\Program Files\Redpoint Global*

Creating a Recovery Point

To create an RPI recovery point, from which a system can be restored, please follow these steps:

1. Stop the following RPI Windows services:
    - Execution
    - Node Manager
2. In SQL Server back up the following databases:
    - Pulse
    - Pulse_Logging
    - Interaction_xxx
    - InteractionAudit_xxx
3. In Windows File Explorer, back up the following folders:
    - At the RPI application server:
        - C:\inetpub\wwwrpi*
        - C:\Program Files\Redpoint Global*
    - At the RPI Realtime server (if applicable):
        - C:\inetpub*rpi*

Restore Procedure

To restore RPI from a recovery point, please follow these steps:

1. Using Server Workbench, take all cluster nodes offline.
2. Stop all cluster nodes' Redpoint Interaction Node Manager Services.
3. In IIS, stop any RPI-related websites.
4. In SQL Server restore the following databases from the backup:
    - Pulse
    - Pulse_Logging
    - Interaction_xxx
    - InteractionAudit_xxx

5. In Windows File Explorer, restore the following folders from the backup:
    - At the RPI application server:
        - C:\inetpub\wwwrpi*
        - C:\Program Files\Redpoint Global*
    - At the RPI Realtime server (if applicable):
        - C:\inetpub*rpi*
4. Start all cluster nodes' Redpoint Interaction Node Manager Services.
5. In IIS, start any RPI-related sites
6. Using Server Workbench, put all cluster nodes back online.

Note that, if you restore from a recovery point, all activities undertaken and artifacts created within RPI since the original recovery point backup was created will be lost.

Monitoring

1. Using a Network Monitoring Tool, monitor the following Windows services (automatic restart where appropriate):
    - RPI Node Manager Service
    - RPI Execution Service
2. Using a Network Monitoring Tool, monitor the status of following URL (if inaccessible, client applications will be unable to connect to Interaction):
    - https://<ExternalServerName>/Interaction
3. SSL Certificate Expiration Monitoring

SQL Server

1. In SQL Server, periodically identify and create essential indexes missing on database tables: http://msdn.microsoft.com/en-us/library/ms345417(v=sql.105).aspx
    - This is recommended only for the <ClientDataWarehouse> and NOT the three Interaction databases
2. In SQL Server, for all four databases, periodically remedy index fragmentation by either reorganizing or rebuilding indexes: http://msdn.microsoft.com/enus/library/ms189858(v=sql.105).aspx
3. In SQL Server, for all four databases, periodically update the distribution of statistics that are used by SQL Server to optimize navigation through tables during the processing of Transact-SQL statements: http://msdn.microsoft.com/enus/library/ms178678(v=sql.105).aspx

4. In SQL Server, for all four databases, periodically check the allocation and structural integrity of user and system tables as well as indexes in the databases: http://msdn.microsoft.com/en-us/library/ms180226(v=sql.105).aspx

Changing the Data Warehouse Schema

There may arise occasions where a given client's data warehouse's schema needs to change. A SQL Server stored procedure (op_RenameDefaultSchema) is provided in each client's Interaction operational database. This can be executed, using the following syntax, to make the change:

```
exec dbo.op_RenameDefaultSchema 'oldschema', 'newschema'
```

Note that the decision to separate the Offer History schema from the main Data Warehouse schema is one that should be made during the planning stages of your RPI deployment. Configuring a separate Offer History schema after the application is live and assets have been built could result in audience and interaction execution errors and is not recommended.

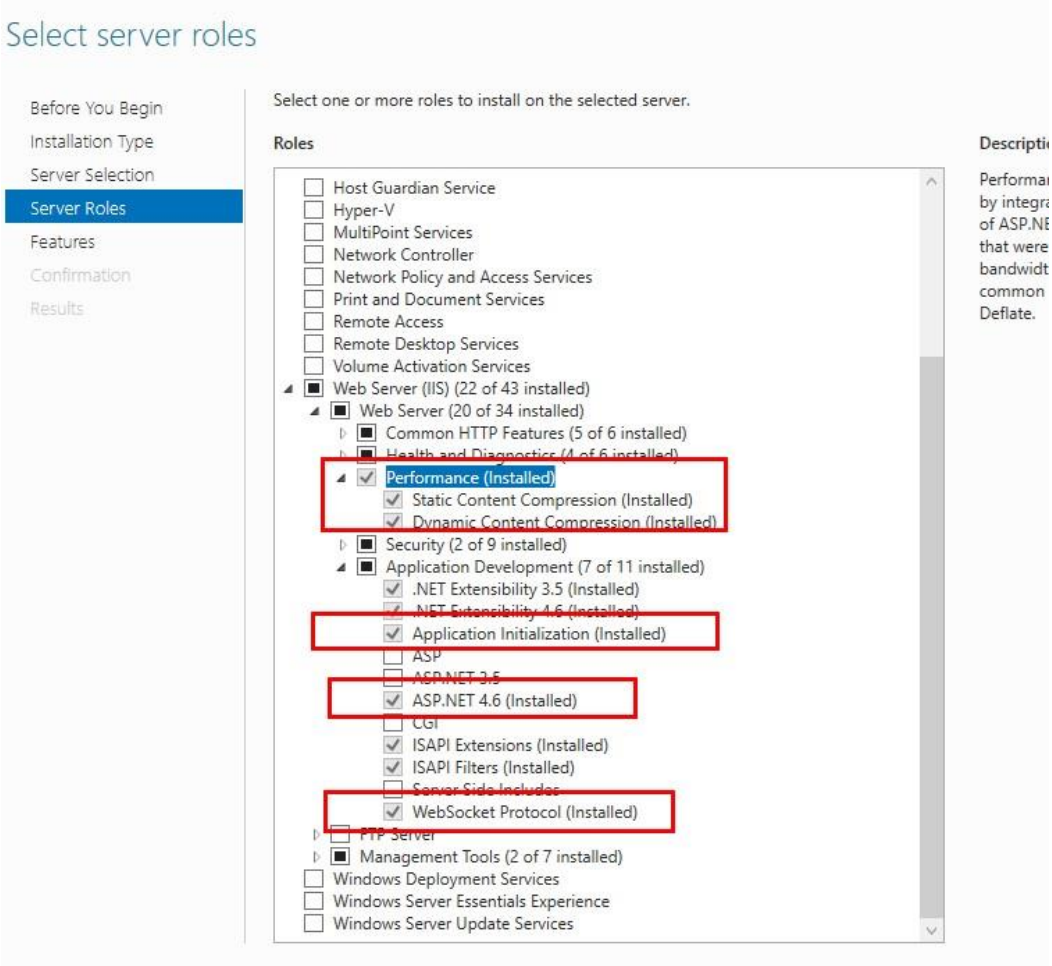
RPI Integration API Installation

The RPI Integration API provides a series of endpoints that allow for a third party development team to invoke RPI functionality. The following section describes installation of the RPI Integration API.

Prerequisites

- Microsoft .NET Framework 4.8
- IIS Configured with the following options ○ Static Content Compression ○ Dynamic Content Compression ○ ASP.NET 4.6
    - ○ Application Initialization

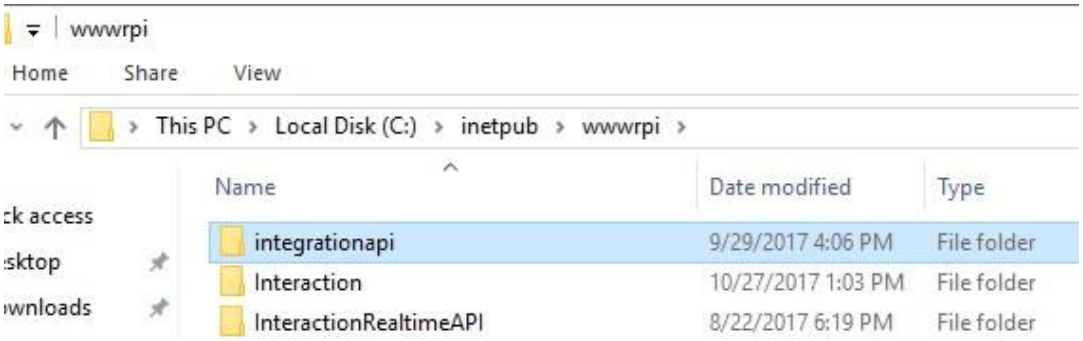
- WebSocket Protocol



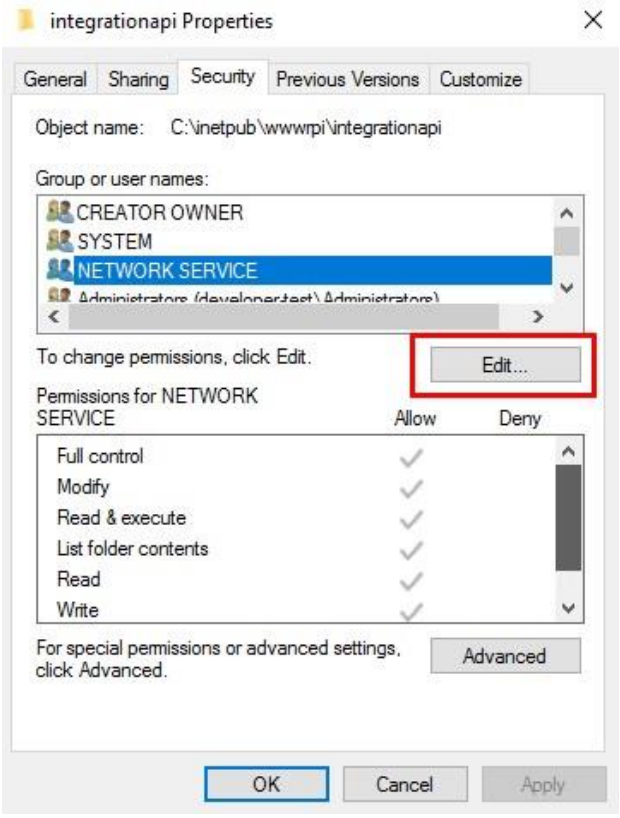
- IIS URL Rewrite 2.1 - https://www.iis.net/downloads/microsoft/url-rewrite

Integration API Installation

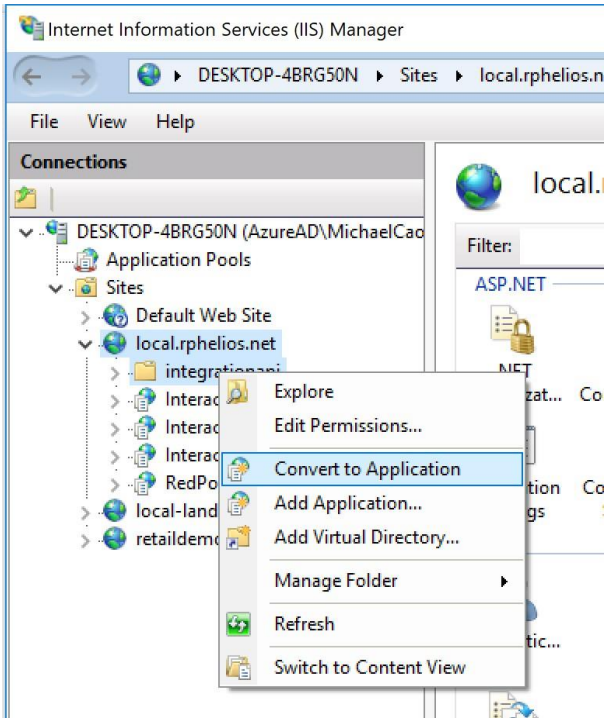
1. Create an 'integrationapi' directory in the root of the RPI web application



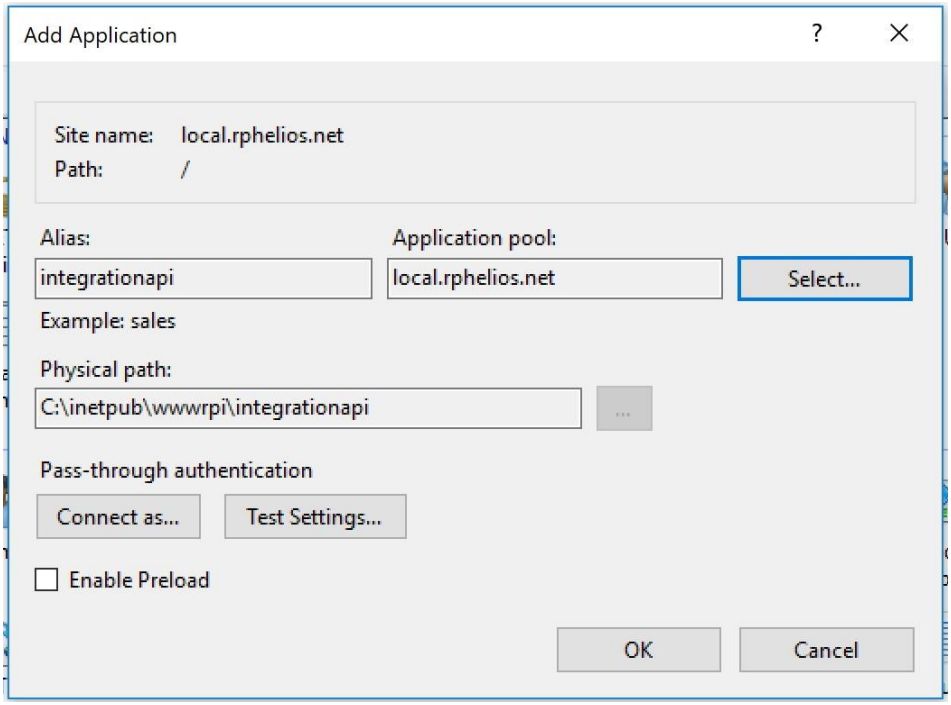
2. Copy the contents of API-Deployment-Files.zip to the integationapi directory
3. Modify the ACL properties for the App_Logs and App_Temp folders to allow “Full Control” for the ”Network Service” user.

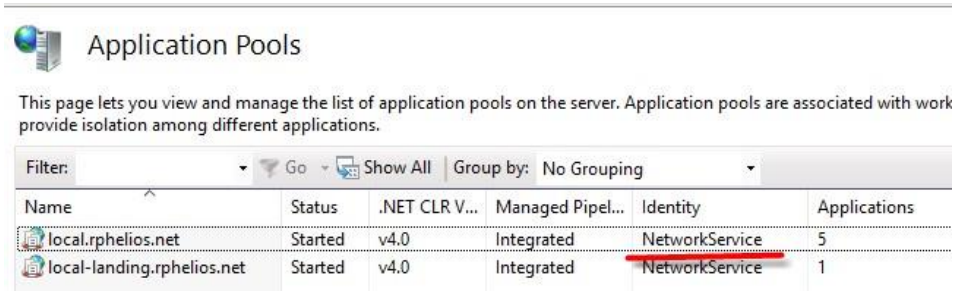


4. Convert the integationapi directory to an IIS Application

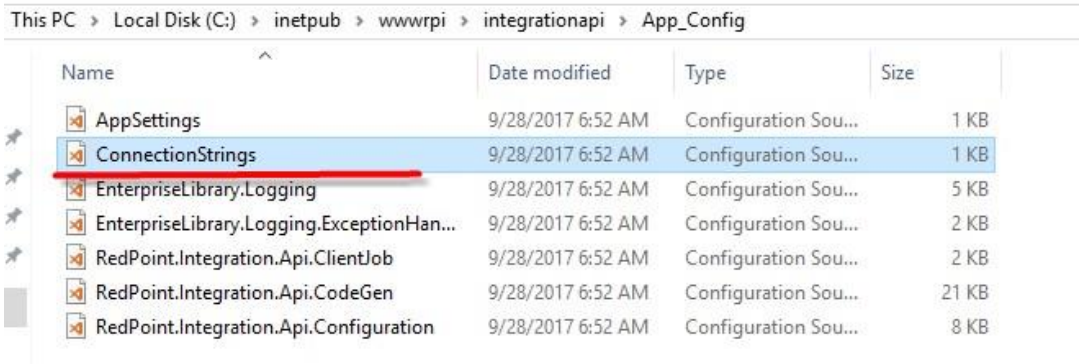


5. Select an Application Pool configured to run under the NetworkService Identity for the Integration API application.

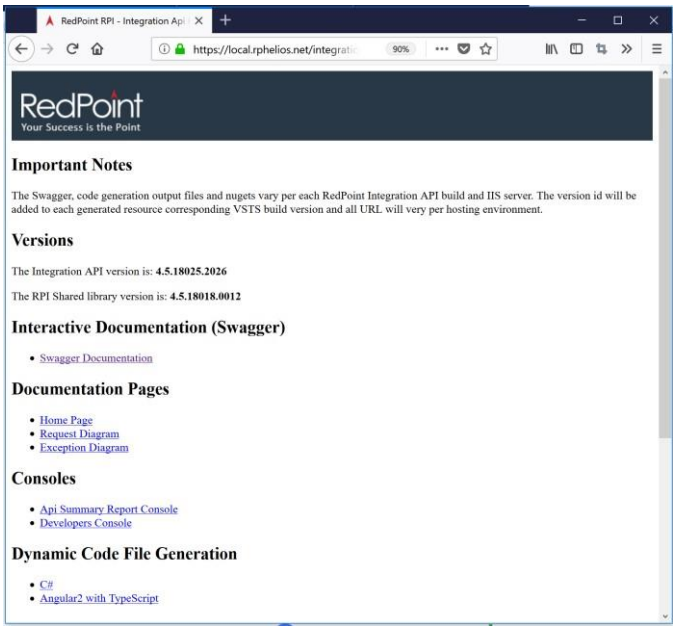




6. The Integration API database connection string is configured to use SQL Server Integrated Security. This connection string must match the Redpoint Interaction connection sting. Connection strings are configured in the ConnectionStrings.config file in the App_Config directory at the root of the integrationapi directory.



7. Navigate to the new Integration API application in a web browser to confirm your installation.



Integration API User Group

User accounts that require access to the Integration API must be assigned to the Integration API user group. User group assignment can be undertaken in the Server Workbench Clients tab.

API users that require access to the Integration API cluster endpoints will require the Cluster administrator permission, which can be managed in the same context.

Appendix A – Database Preparation

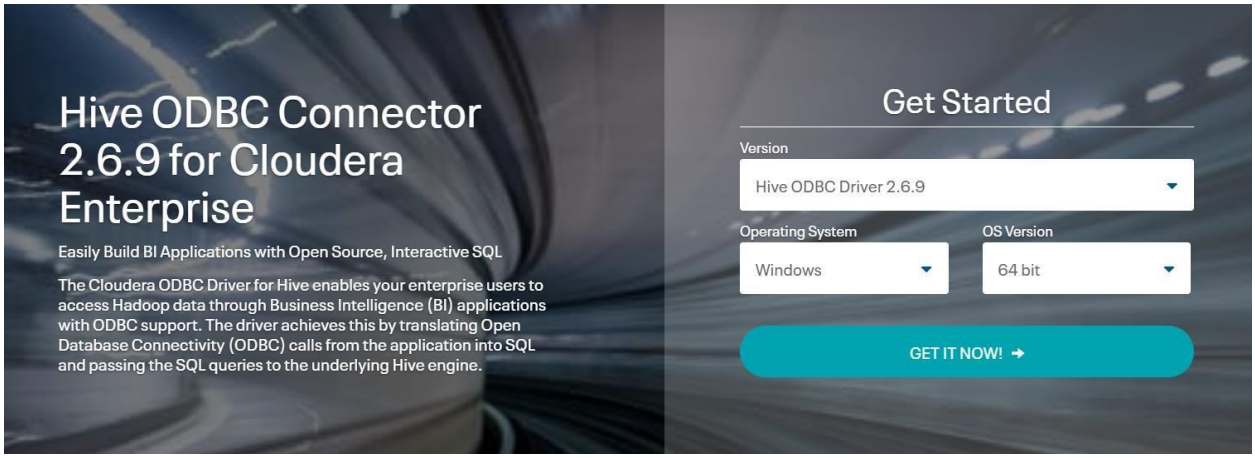
This section describes the steps to be undertaken to ensure that databases are prepared for use by RPI as a data warehouse or auxiliary database.

Apache Hive Configuration

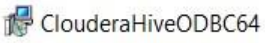
This section describes how to create and configure an Apache Hive Data Source Name (DSN). Please follow the steps below. Please note that you may skip steps 1 – 4 if you have already installed the ODBC driver.

1. In a web browser, navigate to **https://www.cloudera.com/downloads/connectors/hive/odbc/2-6-9.html** to download the driver.

2. Select latest Hive ODBC driver, click Windows Operating System and select 64 bit. Click Get It Now to download the driver:



3. After downloading the file, go to your download folder. Locate and double click the ClouderaHiveODBC64.msi file.

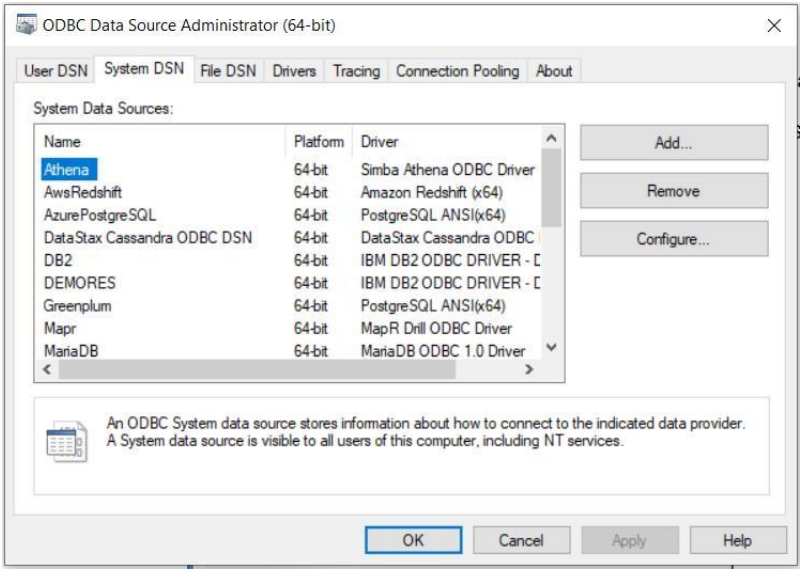


4. In the installation Wizard Window, click Next and follow the required steps to install the driver.

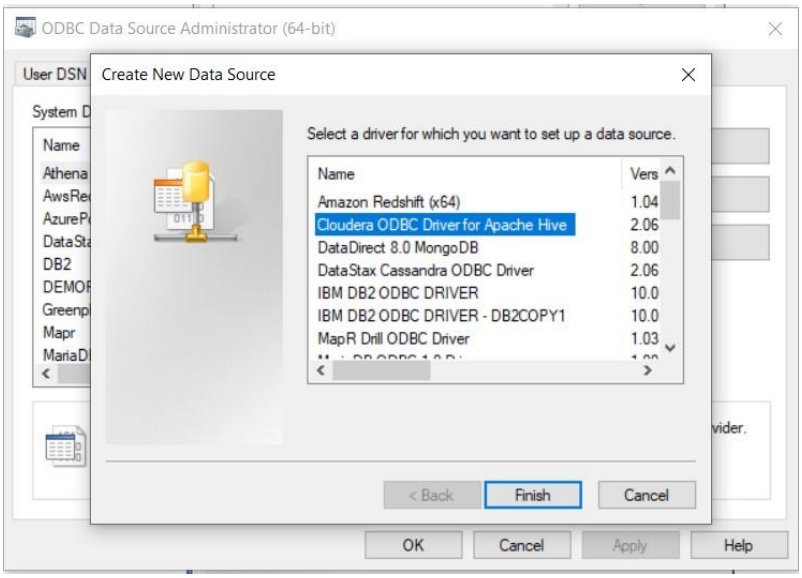
5. Once you have successfully installed the ODBC driver, go to Control Panel\System and Security\Administrative Tools and click ODBC Data Sources (64).

6. In the ODBC Data Sources Administrator Window, click the System DSN tab.

7. Click the Add… button to create a new Data Source.



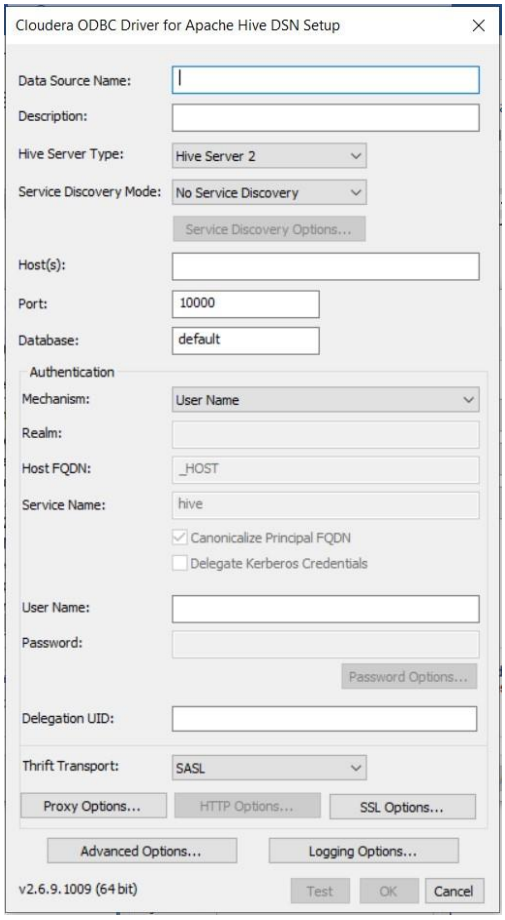
8. Find and select Cloudera ODBC Driver and click Finish.



9. In the Cloudera ODBC Driver DSN Setup Window, configure the following details:

   - Data Source Name

- Host: Hive host server (IP address or server name).
- Port: Hive server port number. Only numerical values are accepted. The default Hive server port is 10000.
- Database: the Hive database name (the default name is ‘default’).
- Hive Server Type: the type of Hive server configuration. Supported values are ‘Hive Server 1’ (the default) and ‘Hive Server 2’. Select ‘Hive Server 2’.
- Mechanism: the type of authentication used. In this case, select ‘User Name’.
- User Name: the username to be used to connect to Hive. Enter the value ‘hive’.



10. Click the Test button. Once the connection has been made successfully, click the OK button to create the DSN.

Apache Spark Configuration

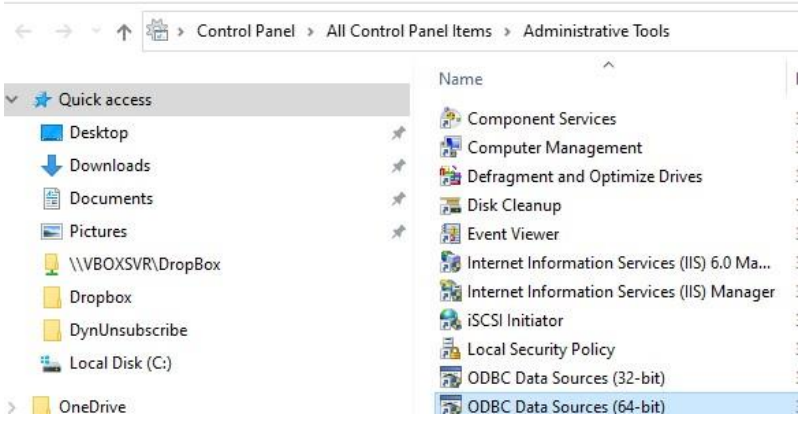
This section describes how to create and configure an Apache Spark Data Source Name (DSN). Please follow the steps below. Please note that you may skip steps 1 – 3 if you have already installed the ODBC driver.

1. In a web browser, navigate to https://www.simba.com/connectors/apache-sparkdriverodbc to download the driver.

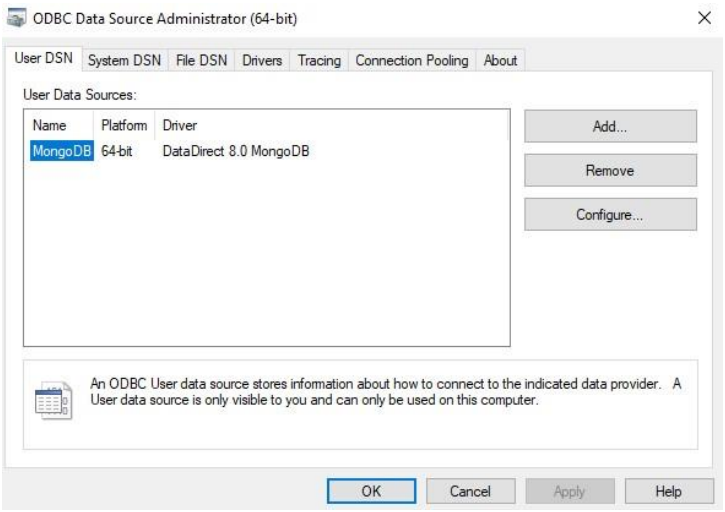
2. In the download folder, double click the SimbaSparkODBC64.msi file.

3. In the Simba Spark ODBC Driver 64-bit Setup Window, click Next and follow the required steps to install the driver.

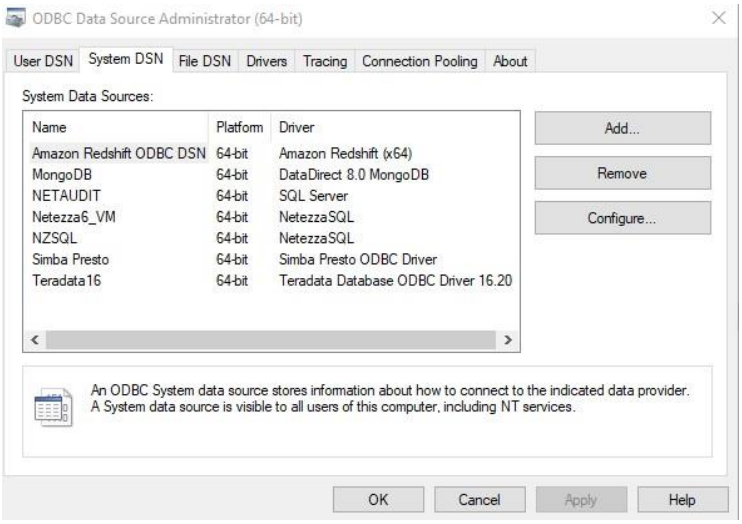
4. Once you have successfully installed the ODBC driver, go to Control Panel\All Control Panel Items\Administrative Tools and click Data Sources (ODBC).



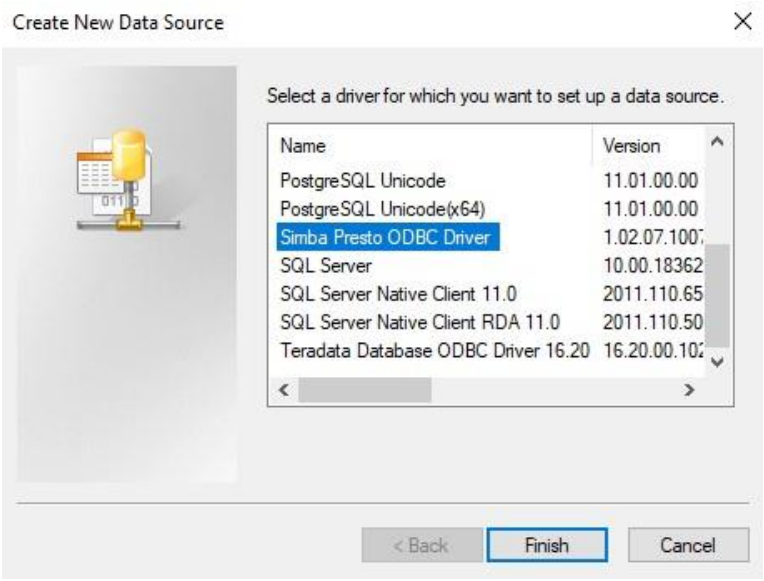
5. In the ODBC Data Source Administrator Window, click the System DSN tab.



6. Click the Add… button to create a new Data Source.

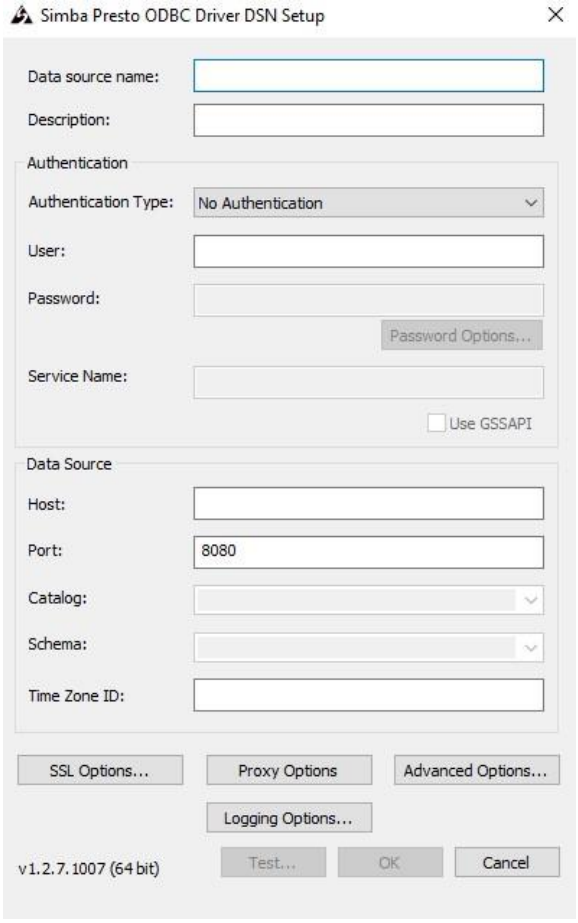


7. Find and select Simba Spark ODBC Driver and click Finish.


Create New Data Source
Select a driver for which you want to set up a data source.
Name
Version
PostgreSQL Unicode
11.01.00.00
PostgreSQL Unicode(x64)
11.01.00.00
Simba Presto ODBC Driver
1.02.07.1007
SQL Server
10.00.18362
SQL Server Native Client 11.0
2011.110.65
SQL Server Native Client RDA 11.0
2011.110.50
Teradata Database ODBC Driver 16.20
16.20.00.102
< Back
Finish
Cancel

8. In the Simba Spark ODBC Driver DSN Setup Window, configure the following details:

    - Data Source Name
    - Host: the Spark host server (IP address or a server name).
    - Port: the Spark server port number. Only numerical values are accepted. The default Spark server port is 10000.
    - Database: the Spark database name (‘default’) by default.
    - Spark Server Type: the type of Spark server configuration. In this case, select ‘SparkThriftServer (Spark 1.1 and later)’.
    - Mechanism: the type of authentication to be used. Select ‘User Name’.
    - User Name: enter ‘spark’.



9. Click the Test button. Once the connection has been made successfully, click the OK button to create the DSN.

MongoDB (SQL) Configuration

This section describes how to create and configure a MongoDB (SQL) Data Source Name (DSN). Please follow the steps below. Please note that you may skip steps 1 – 4 if you have already installed the ODBC driver.

1. In a web browser, navigate tohttps://www.progress.com/products/datasources/mongodb

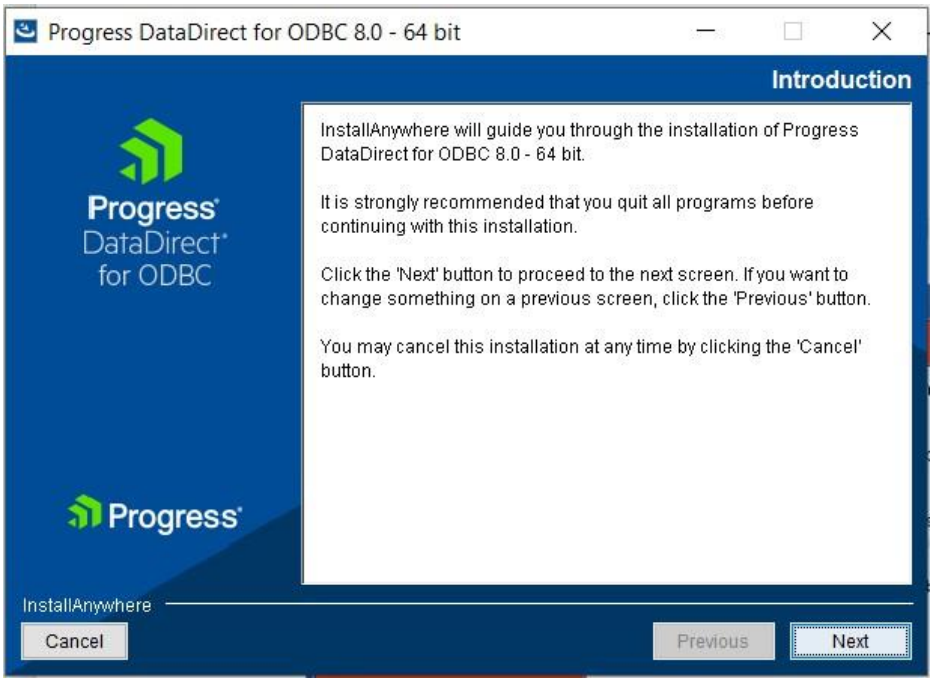
https://www.progress.com/odbc/mongodb to download the driver. 2. In the download folder, locate the

PROGRESS_DATADIRECT_ODBC_MONGODB_WIN_64.zip file and extract its contents.

3. Within the extracted folder, double click the

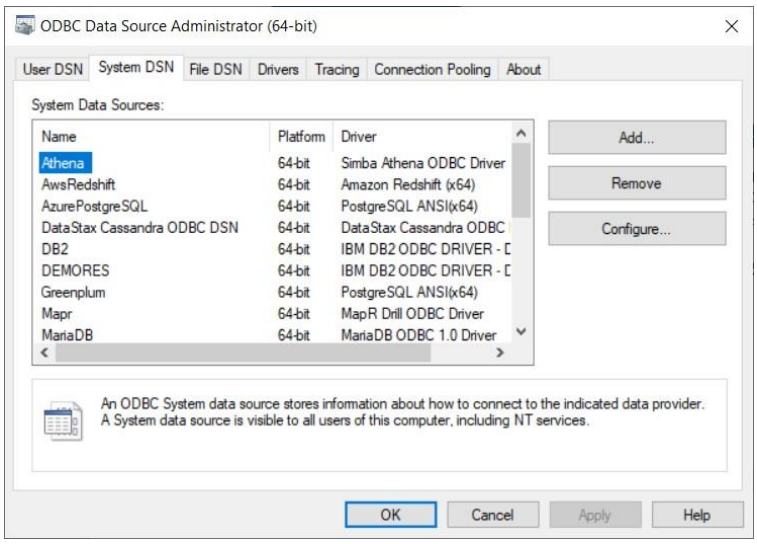
PROGRESS_DATADIRECT_ODBC_8.0_WIN_64_INSTALL.exe file.

4. In the DataDirect for ODBC 8.0 – 64 bit Window, click Next and follow the required steps to install the driver.



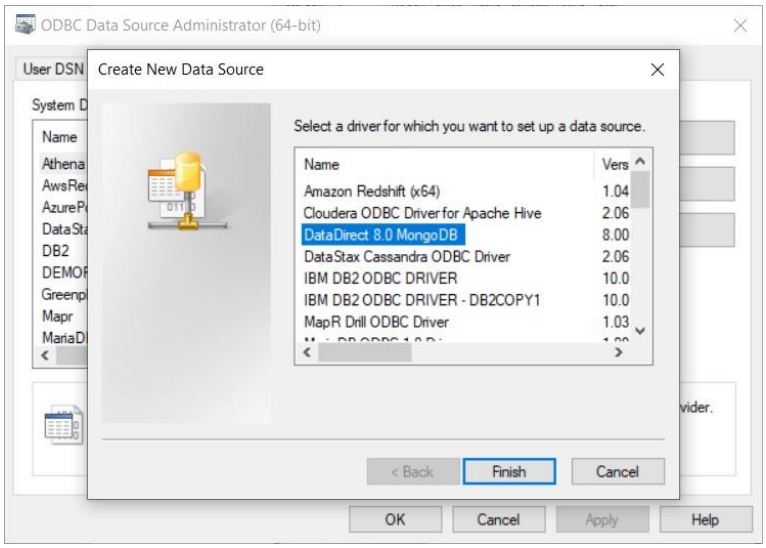
5. Once you have successfully installed the ODBC driver, go to Control Panel\System and Security\Administrative Tools and click ODBC Data Sources (64-bit).

6. In the ODBC Data Source Administrator Window, click the System DSN tab.



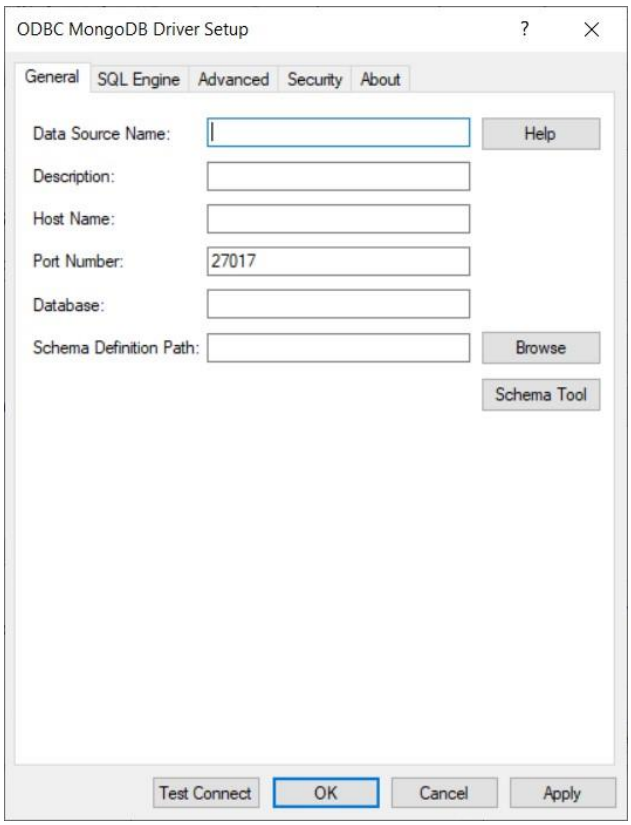
7. Click the Add… button to create a new Data Source.

8. Find and select DataDirect 8.0 MongoDB and click Finish.



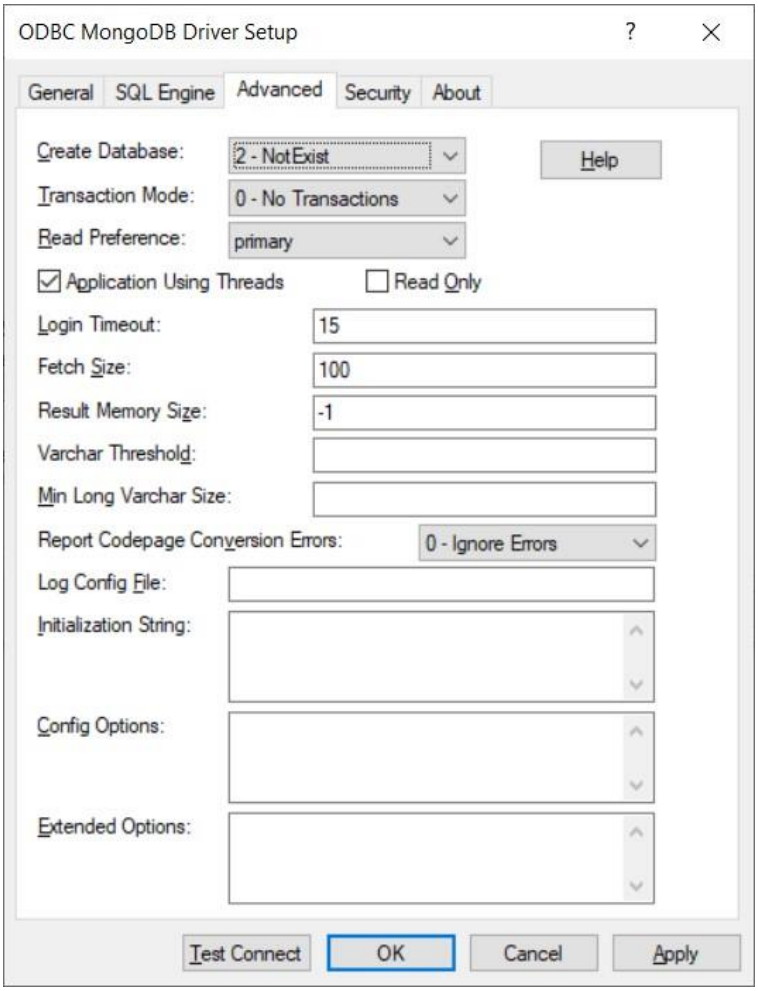
9. In the ODBC MongoDB Driver Setup Window, configure the following details:

   - Data Source Name
   - Host Name: the MongoDB host server (IP address or a server name).
   - Port Number: the MongoDB server port number. Only numerical values are accepted. The default port number is 27017.
   - Database: the MongoDB database name.
   - Schema Definition Path: specifies the name and location of the configuration file where the relational map of native data is written.



10. For more information about the Schema Tool, go to C:\Program Files\Progress\DataDirect\ODBC_80\help\SchemaToolHelp and click schematoolhelp.

11. Click the Advanced tab and make sure the Read Only option is not selected.



12. Click the Test Connect button. Once the connection has been made successfully, click the OK button to create the DSN.

AWS Redshift Configuration

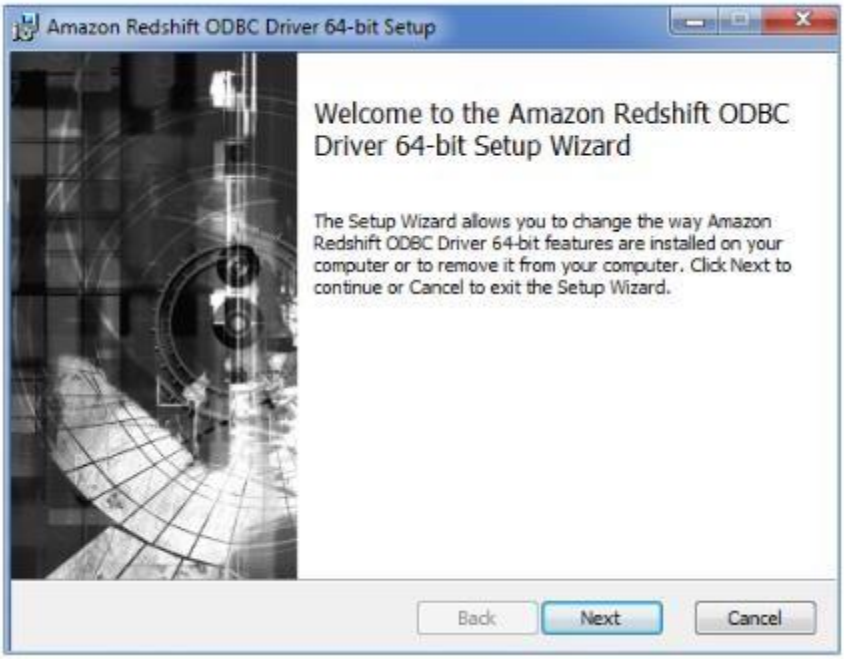
This section describes how to create and configure an AWS Redshift Data Source Name (DSN). Please follow the steps below. Please note that you may skip steps from 1 –4if you have already installed the ODBC driver.

1. In a web browser, navigate to https://docs.aws.amazon.com/redshift/latest/mgmt/install-odbc-driver-windows.html to download the driver.

2. Locate https://s3.amazonaws.com/redshiftdownloads/drivers/odbc/1.4.3.1000/AmazonRedshiftODBC64-1.4.3.1000.msi to download the 64bit Amazon Redshift ODBC installer

3. In the download folder, double click the AmazonRedshiftODBC64-1.4.3.1000.msi file.

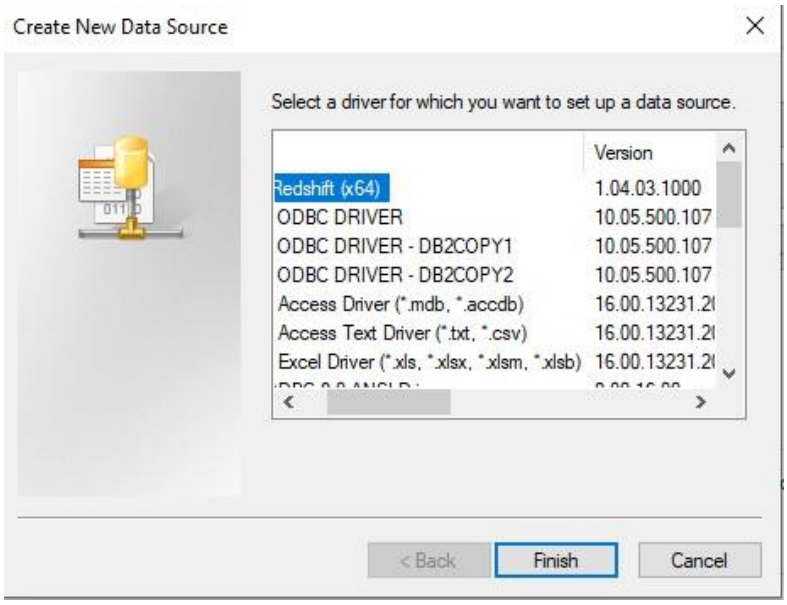
4. In the Amazon Redshift ODBC Driver 64-bit Setup Window, click Next and follow the required steps to install the driver.



5. Once you have successfully installed the ODBC driver, go to Control Panel\All Control Panel Items\Administrative Tools and click Data Sources (ODBC).

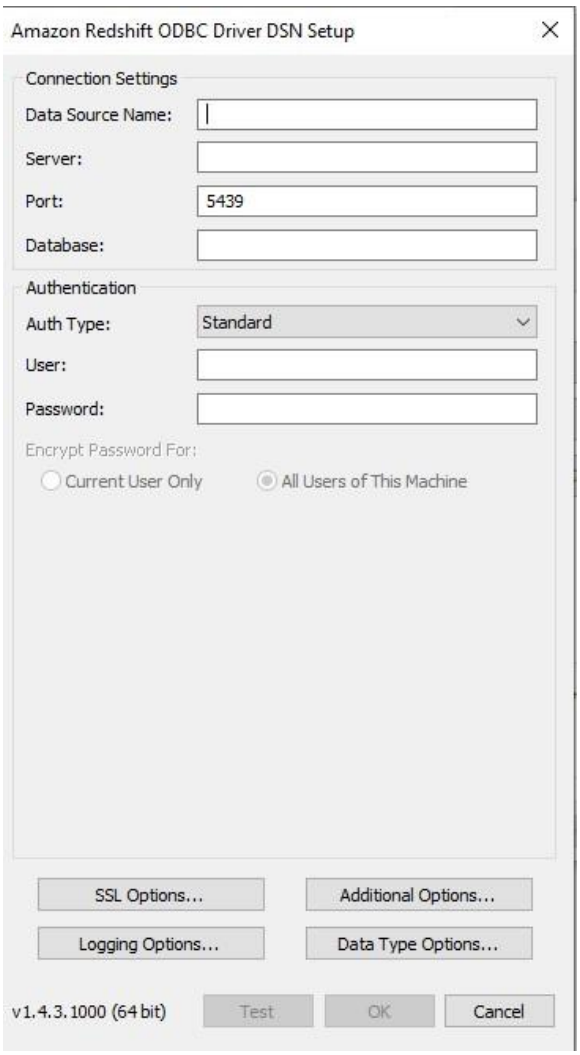
6. In the ODBC Data Source Administrator Window, click the System DSN tab.
7. Click the Add...button to create a new Data Source.

8. Find and select Amazon Redshift (x64) and click Finish.



9. In the Amazon Redshift ODBC Driver DSN Setup Window, configure the following details:

- Data Source Name
- Server: the Amazon Redshift cluster endpoint URL, i.e. ‘xxx.xxx.endpointregion.redshift.amazonaws.com’
- Port: the cluster port number. Only numerical values are supported. The default is 5439.
- Database: the database’s name.
- Auth Type: must be set to ‘Standard’
- User: the database username.
- Password: the database password.
- Encrypt Password For: must be set to ‘All Users of This Machine’
- Additional Options: change the radio button selection to Use ‘Multiple Statements’.



10. Click the Test button. Once the connection has been made successfully, click the OK button to create the DSN.

11. Once completed, launch Server Workbench and login.

12. In the Install Client at Data warehouse pane, select AWS Redshift as the database provider.

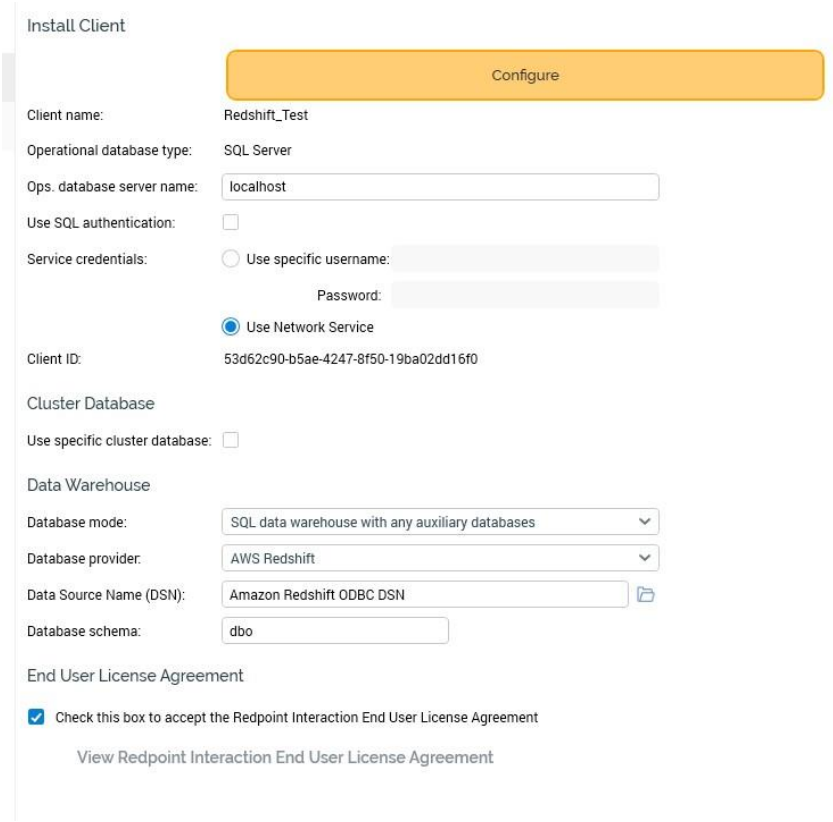
13. Enter the Server name. This can be either the DNS (Domain Name System) or an IP address of the server.

14. Enter the Database provider

15. Enter the Data Source Name

16. Enter the Database schema

17. Check the End User License Agreement checkbox and then click Next



Greenplum/PostgreSQL Configuration

This section describes how to create and configure a Greenplum/PostgreSQL Data Source Name (DSN). Please follow the steps below. Please note that you may skip steps 1 – 3 if you have already installed the ODBC driver.

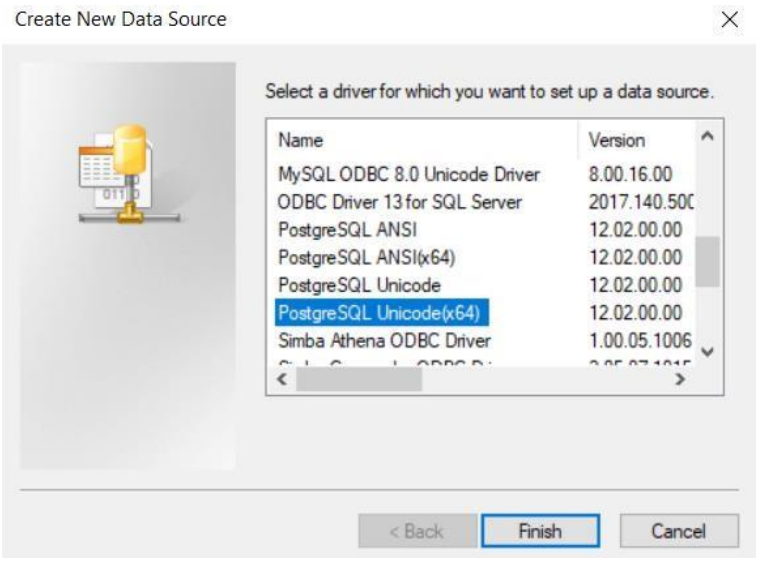
1. In a web browser, navigate to http://www.postgresql.org/ftp/odbc/versions/msi/ to download the driver.
2. Download the latest and stable release version of the driver on the x64 bit platform.
3. In the download folder, locate **psqlodbc_12_02_0000-x64.zip** and extract its contents.
4. Within the extracted folder, double click psqlodbc_x64.msi.
5. In the psqlODBC_X64 Setup Window, click Next and follow the required steps to install the driver.

6. Once you have successfully installed the ODBC driver, go to Control Panel\All Control Panel Items\Administrative Tools and click Data Sources (ODBC).

7. In the ODBC Data Source Administrator Window, click the System DSN tab.

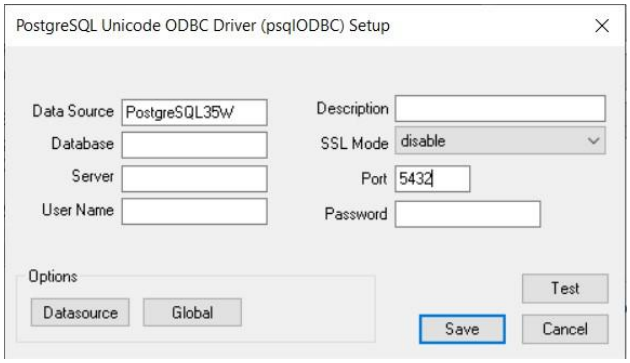
8. Click the Add… button to create a new Data Source.

9. Find and select PostgreSQL Unicode (x64) and click Finish.



10. In the PostgreSQL Unicode ODBC Driver (psqlODBC) Setup Window, configure the following details:

- Data Source Name
- Database: the database name.
- Server: the host server (IP address or server name).
- Port: the server port number. Only numerical values are supported. The default server port is 5432.
- User Name: the database username.
- Password: The database password.



11. Click the Test button. Once the connection has been made successfully, click the Save button to create the Data Source Name (DSN).

Splice Machine Configuration

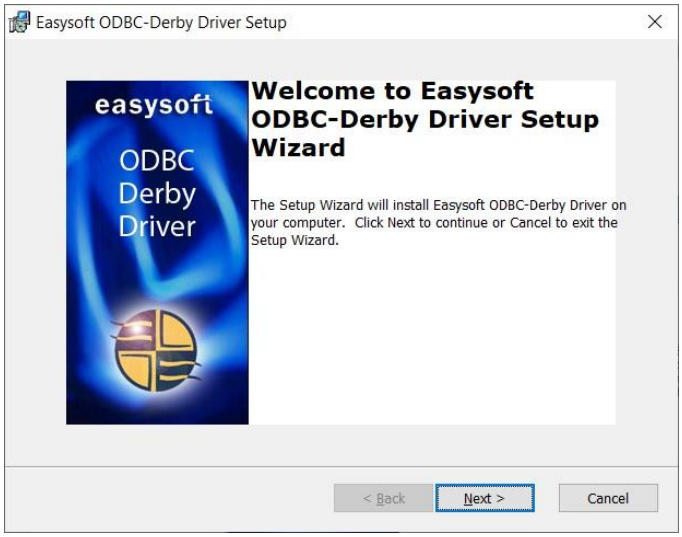
This section describes how to create and configure ac Splice Machine Data Source Name (DSN). Please follow the steps below. Please note that you may skip steps 1 – 3 if you have already installed the ODBC driver.

1. In a web browser, navigate to**http://www.easysoft.com/products/data_access/odbcderby-driver/ - section=tab-1**
   https://www.easysoft.com/cgibin/productdownload.cgi?p=58**http://www.easysoft.com/products/data_access/odbcderby-driver/ - section=tab-1** to download the driver.

2. In the download page, set Platform to 'Windows - x86 - (32 - Bit, 64 - Bit)' and proceed to download the driver.

3. In the download folder, double click odbc-derby-1_1_0-windows.exe.

4. In the Easysoft ODBC-Derby Driver Setup Window, click Next and follow the required steps to install the driver.

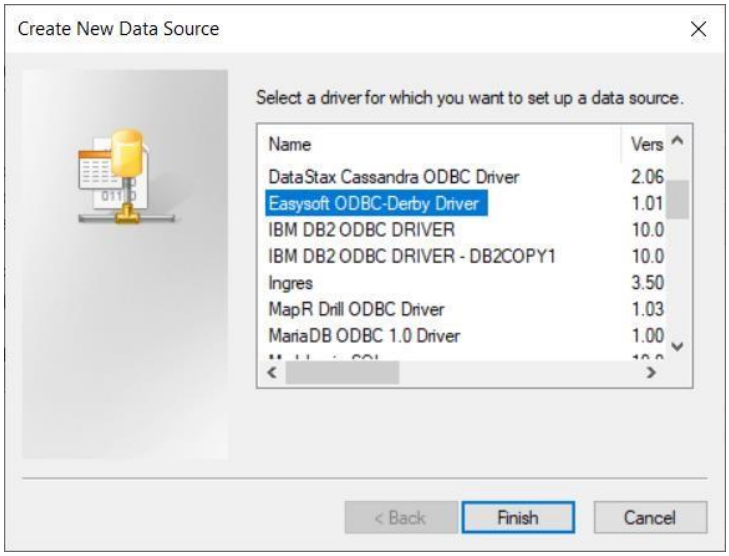


5. Once you have successfully installed the ODBC driver, go to Control Panel\All Control Panel Items\Administrative Tools and click ODBC Data Sources (64-bit).

6. In the ODBC Data Source Administrator Window, click the System DSN tab.

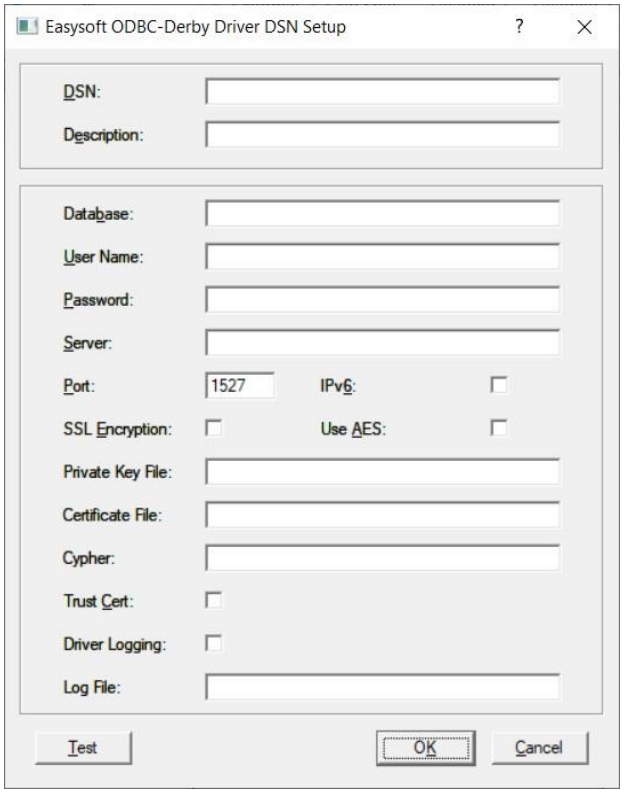
7. Click the Add… button to create a new Data Source.

8. Find Easysoft ODBC-Derby Driver and click Finish.



9. In the Easysoft ODBC-Derby Driver DSN Setup Window, configure the following details:

- DSN: the name of the Data Source.
- Database: the database name.
- User: the database username.
- Password: the database password.
- Server: the host server (IP address or server name). ○ If the server is hosted in Amazon, use the EC2 DNS name, i.e. 'ec2-xx-xxx-xxxxxx.compute-1.amazonaws.com'.
- Port: the server port number. Only numerical values are supported. The default server port is 1527.



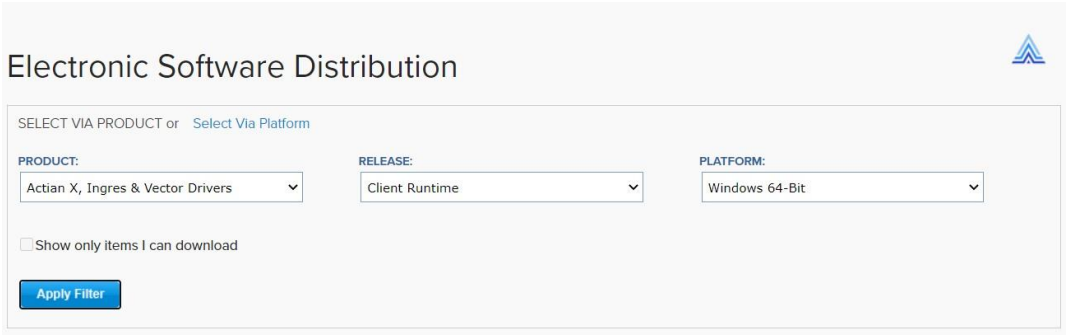
10. Click the Test button. Once the connection has been made successfully, click the OK button to create the DSN.

Actian VectorH Configuration

This section describes how to create and configure an Actian VectorH Data Source Name (DSN). Please follow the steps below. Please note that you may skip steps 1 – 3 if you have already installed the ODBC driver.

1. In a web browser, navigate to http://esd.actian.com/product/drivers/Client_Runtime to download the driver.

2. In the download page, set Product to ‘Actian Vector and Ingres’, Release to ‘Client Runtime’ and Platform to ‘Windows 64-bit’.

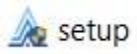


3. Scroll down to locate and click Client Runtime.

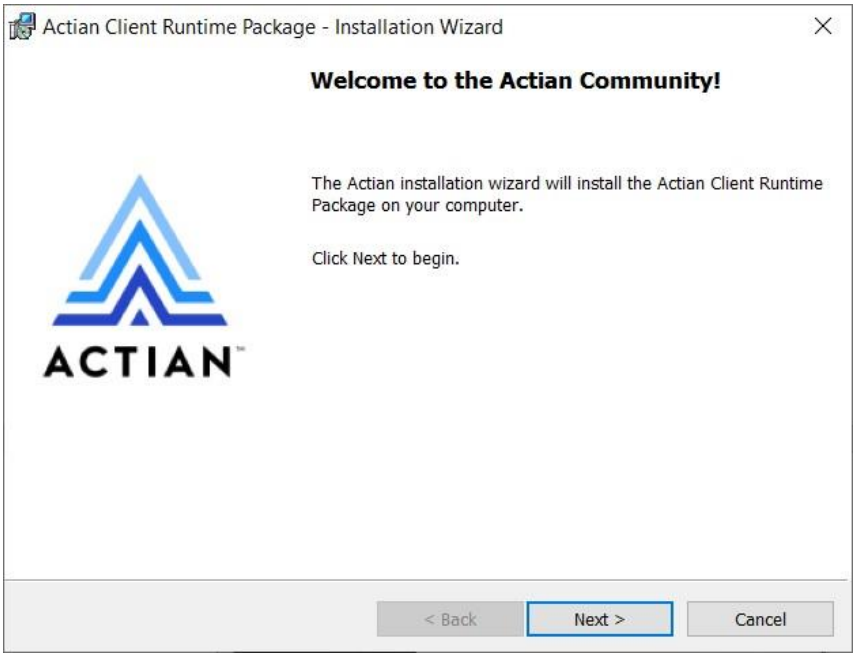
4. Click Download to download the driver.

5. In the download folder, locate clientruntime-vw-6.0.0-129-win-x86_64 file and extract its contents. setup.exe.

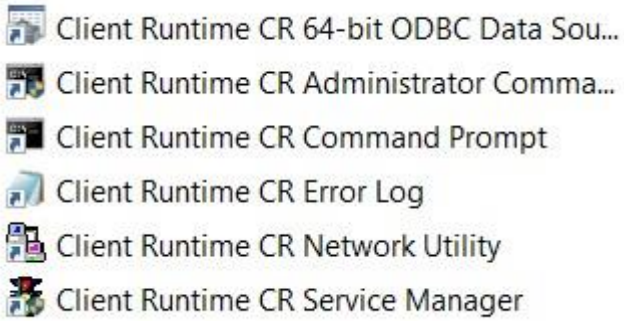
6. Within the extracted folder, double click setup.exe.



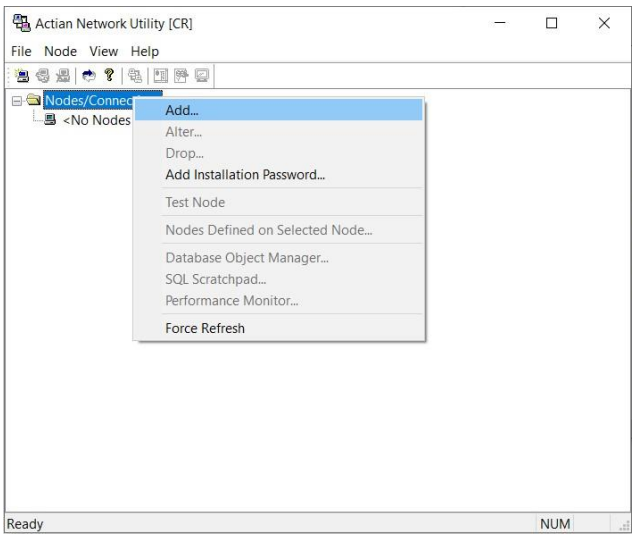
7. In the Actian Client Runtime Package Window, click Next and follow the required steps to install the driver.



8. Once you have successfully installed the client runtime, go to ProgramData\Microsoft\Windows\Start Menu\Programs\Actian Client Runtime CR\Client Runtime CR and click the Client Runtime CR Network Utility.

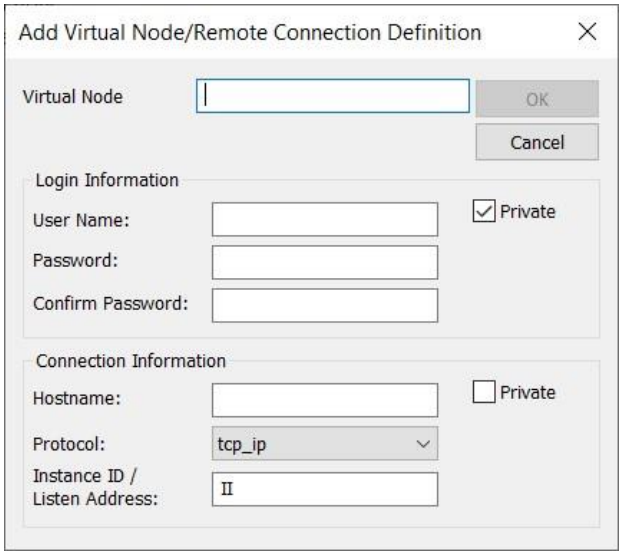


9. Right click Nodes/Connections and click Add… to add a new virtual node.



10. Configure the following details:

- Virtual Node Name: the name of your virtual node.
- User Name: the database username.
- Password: the database password.
- Confirm Password
- Hostname: the host server (IP address or server name). ○ If the server is hosted in Amazon, use the EC2 DNS name, i.e. 'ec2-xx-xxx-xxxxxx.compute-1.amazonaws.com'.
- Protocol: the connection protocol. The default is 'tcp_ip'.
- Instance ID/Listen Address: the connection listen address. The value must be set to 'VH'.



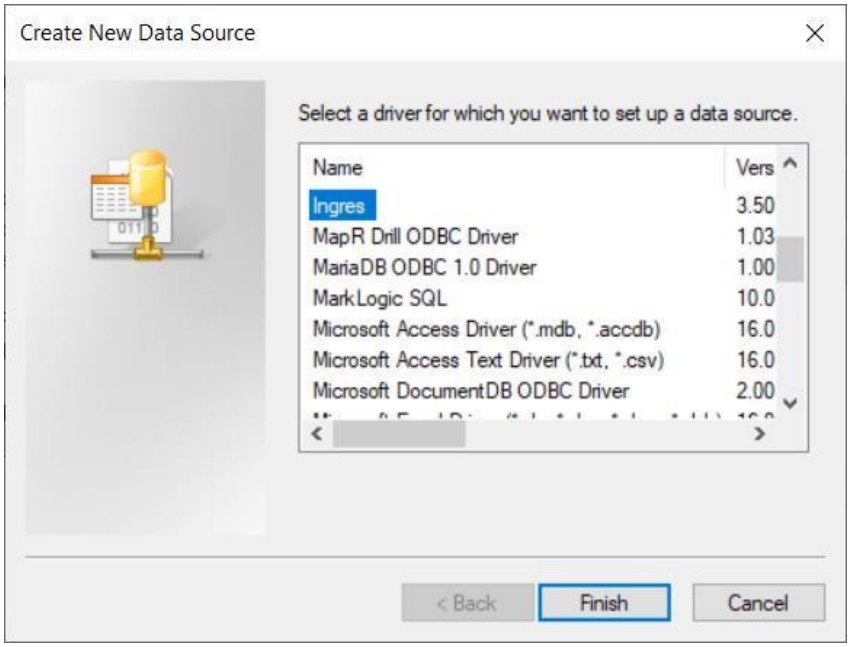
11. Make sure that both Private options are not selected and Click OK.

12. Once you have successfully configured the virtual node, go to Control Panel\All Control Panel Items\Administrative Tools and click ODBC Data Sources (64-bit).

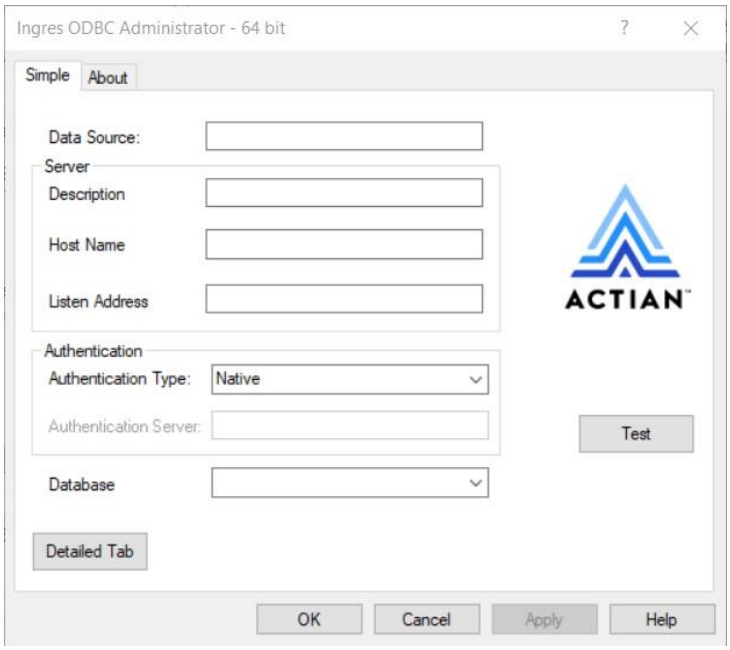
13. In the ODBC Data Source Administrator Window, click the System DSN tab.

14. Click the Add… button to create a new Data Source.

15. Find and select item 'Ingres' and click Finish.



16. In the Ingres ODBC Administrator – 64 bit Window, click the Detailed Tab.



17. In the Detailed tab, configure the following details:

    - Data Source
    - Vnode: the name of the virtual node created earlier.
    - Type: leave the default value ('INGRES').
    - Database: the database name

18. Click the Test button. Once the connection has been made successfully, click the OK button to create the DSN.

DB2 Configuration

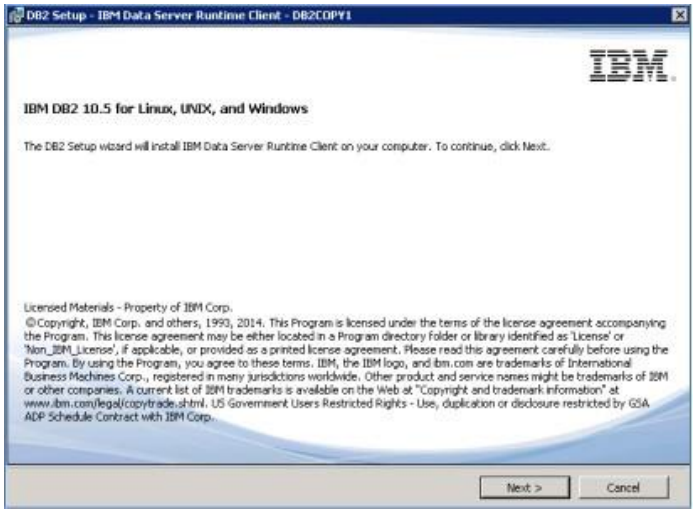
This section describes how to create and configure DB2 Data Source Name (DSN). Please follow the steps below. Please note that you may skip steps 1 – 4 if you have already installed the ODBC driver.

1. In a web browser, navigate to http://www01.ibm.com/support/docview.wss?uid=swg21385217 to download the driver.

2. In the download load page, click IBM Data Server Runtime Client.

3. In the download folder, double click v10.5fp5_ntx64_rtcl.exe.

4. In the DB2 Setup Window, click Next and follow the required steps to install the driver

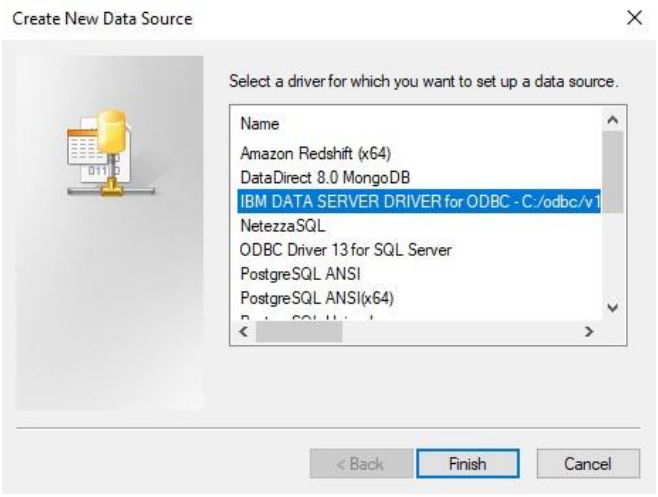


5. Once you have successfully installed the driver, go to Control Panel\All Control Panel Items\Administrative Tools and click Data Sources (ODBC).

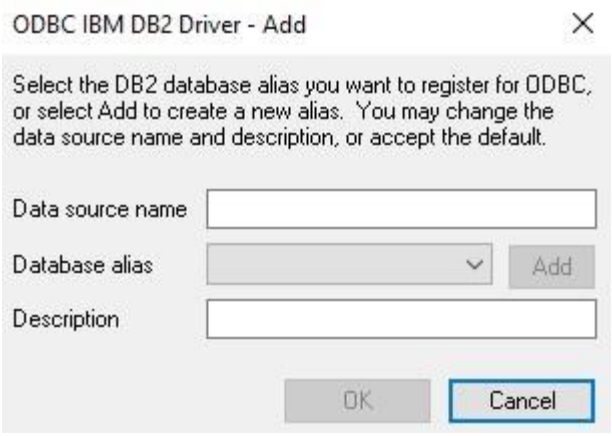
6. In the ODBC Data Source Administrator Window, click the System DSN tab.

7. Click the Add… button to create a new Data Source.

8. Find and select IBM DB2 ODBC DRIVER – BD2COPY1 and click Finish.

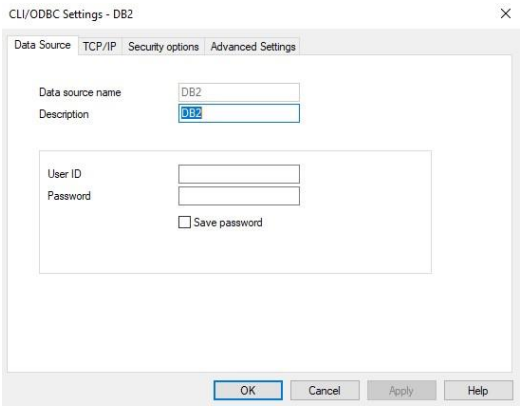


9. Enter the Data source name. Click Add.



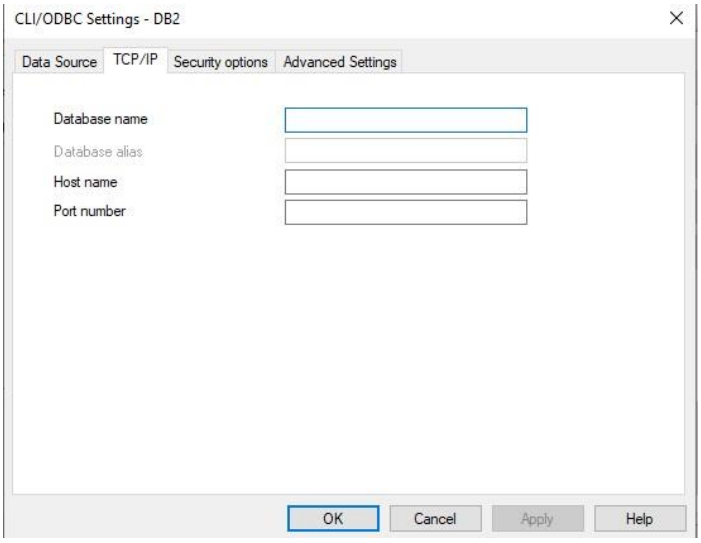
10. In the Data Source Tab, enter the following details:

    - User ID: the database username.
    - Password: the database password.
    - Save password: this option must be selected.



11. Click the TCP/IP tab. Enter the following details:

    - Database name: the name of the DB2 database.
    - Database alias: the DB2 database's alias.
    - Host name: the host server (IP address or server name).
    - Port number: the server port number. Only numerical values are supported. The default server port is 50000.



12. Click the OK button to create the DSN.

13. Once completed, launch Server Workbench and login.

14. In the Install Client at Data warehouse pane, select AWS Redshift as the database provider.

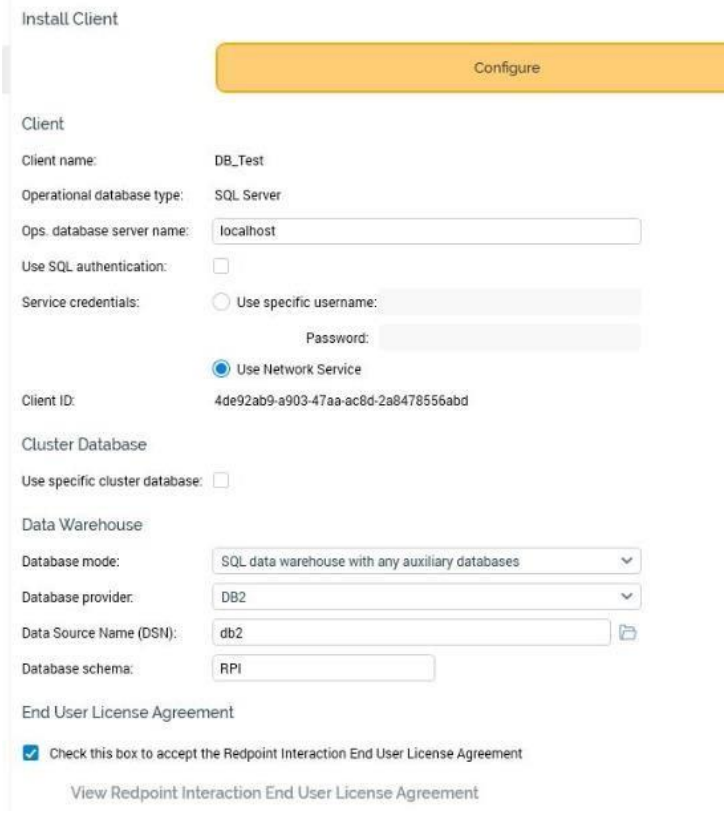
15. Enter the Server name. This can be either the DNS (Domain Name System) or an IP address of the server.

16. Enter the Database provider

17. Enter the Data Source Name

18. Enter the Database schema

19. Check the End User License Agreement checkbox and then click Next



Netezza Configuration

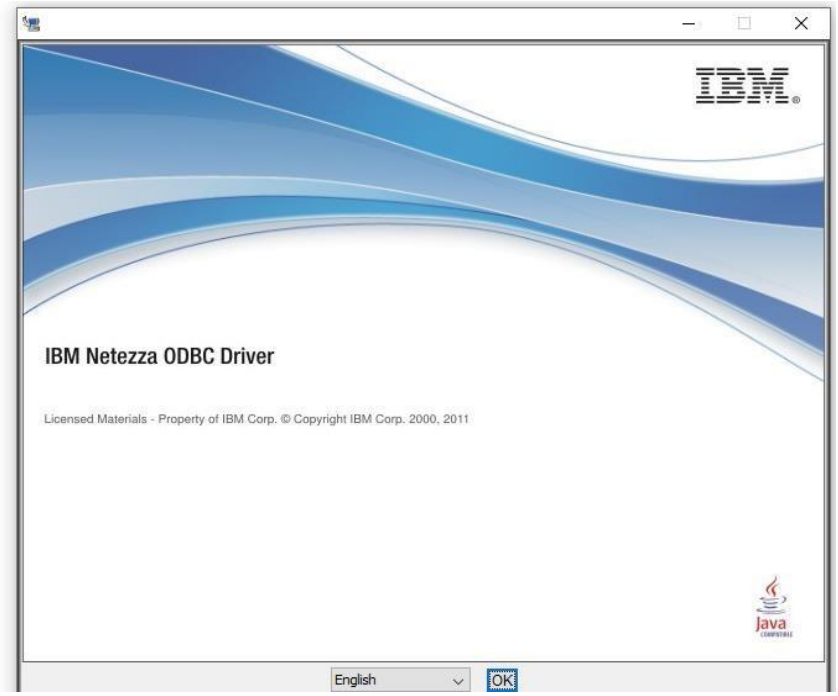
This section describes how to prepare pre-requisites and configure data connections for Netezza databases.

This section describes how to prepare pre-requisites and configure data connections for Netezza databases.

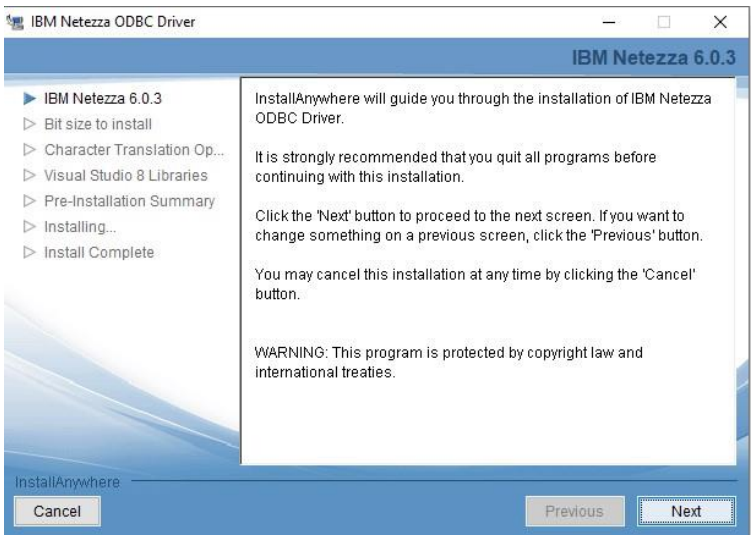
1. If you don’t have the installer, in a web browser, navigate to **https://www-933.ibm.com/support/fixcentral/options**. Access requires an IBM customer account to be able to download the driver.

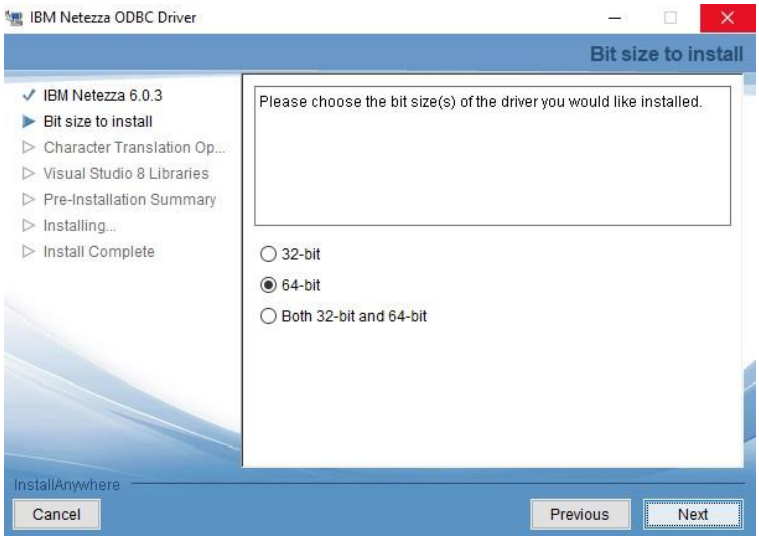
2. Once you have downloaded the driver installer, double click nzodbcsetup to start the installation.

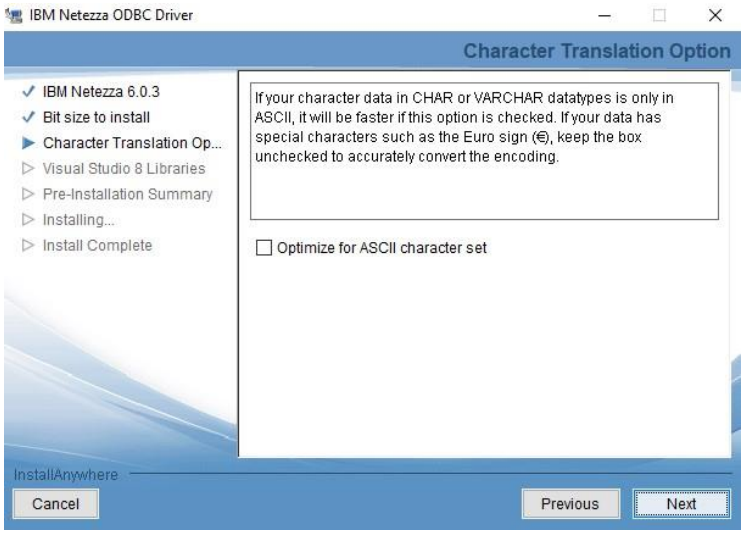
3. When the language options Window opens, accept the English default and click OK. The language options Window controls installation program language.



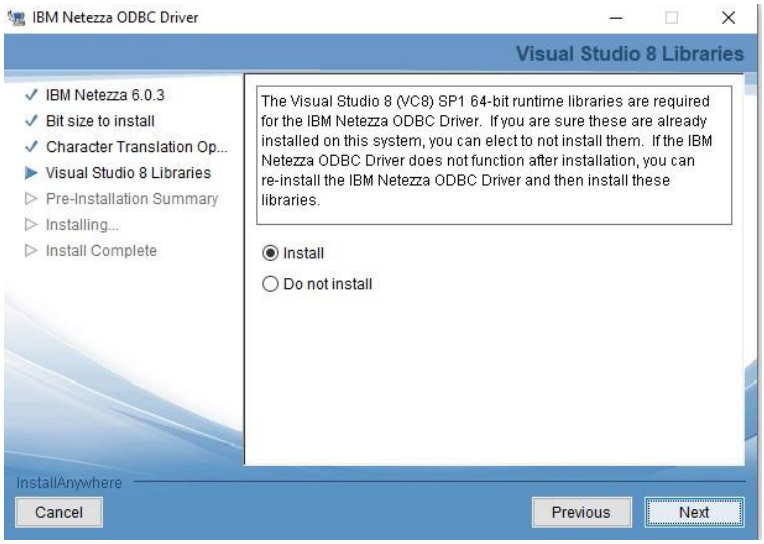
4. When the Introduction Window is displayed, review the information and click Next.


IBM Netezza ODBC Driver
IBM Netezza 6.0.3
IBM Netezza 6.0.3
Bit size to install
Character Translation Op...
Visual Studio 8 Libraries
Pre-Installation Summary
Installing...
Install Complete
InstallAnywhere will guide you through the installation of IBM Netezza ODBC Driver.
It is strongly recommended that you quit all programs before continuing with this installation.
Click the 'Next' button to proceed to the next screen. If you want to change something on a previous screen, click the 'Previous' button.
You may cancel this installation at any time by clicking the 'Cancel' button.
WARNING: This program is protected by copyright law and international treaties.
InstallAnywhere
Cancel
Previous
Next

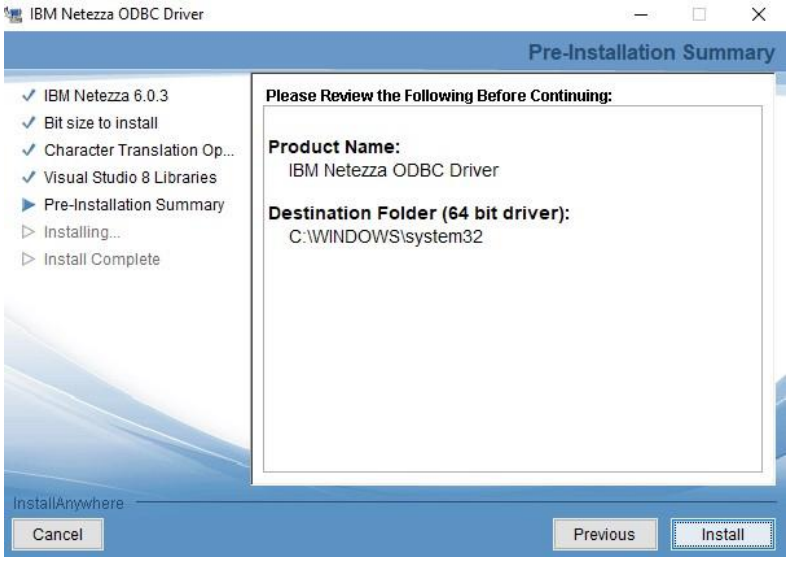

IBM Netezza ODBC Driver
Bit size to install
IBM Netezza 6.0.3
Bit size to install
Character Translation Op...
Visual Studio 8 Libraries
Pre-Installation Summary
Installing...
Install Complete
Please choose the bit size(s) of the driver you would like installed.
32-bit
64-bit
Both 32-bit and 64-bit
InstallAnywhere
Cancel
Previous
Next


IBM Netezza ODBC Driver
Character Translation Option
IBM Netezza 6.0.3
Bit size to install
Character Translation Op...
Visual Studio 8 Libraries
Pre-Installation Summary
Installing...
Install Complete
If your character data in CHAR or VARCHAR datatypes is only in ASCII, it will be faster if this option is checked. If your data has special characters such as the Euro sign (€), keep the box unchecked to accurately convert the encoding.
Optimize for ASCII character set
InstallAnywhere
Cancel
Previous
Next

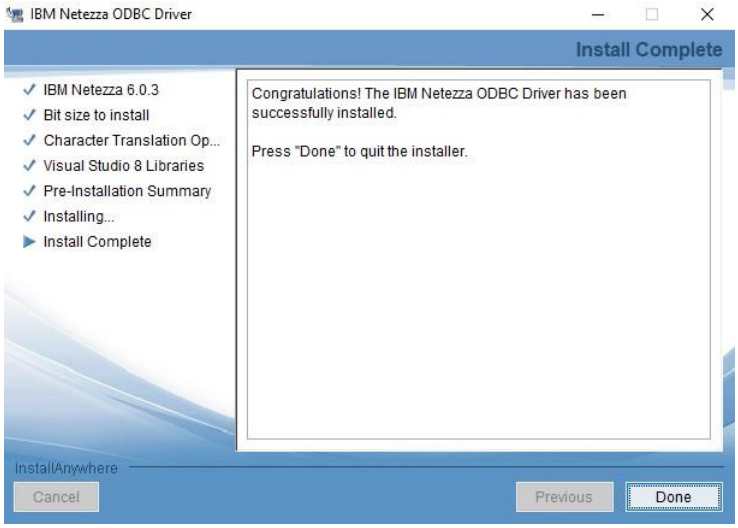
5. Install the Visual Studio 2010 (VS2010) Libraries on the client system.



6. When the Pre-Installation Summary Window opens, use Previous to page through the previous Windows. When finished, click Install to complete the installation process. The Pre-Installation Summary Window summarizes your installation choices.



7. When the Install Complete Window opens, click Done to close the installation program. The Install Complete Window indicates that the installation process completed successfully.



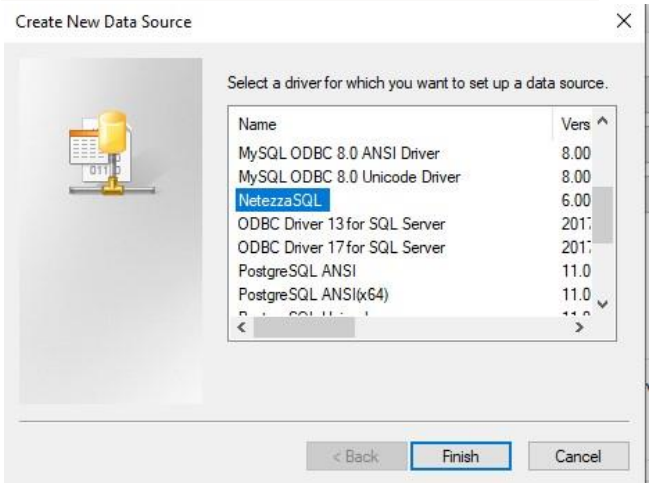
Once you have successfully installed the ODBC driver, go to Control Panel\All Control Panel Items\Administrative Tools and click Data Sources (ODBC).

8.

9. In the ODBC Data Source Administrator Window, click the System DSN tab.

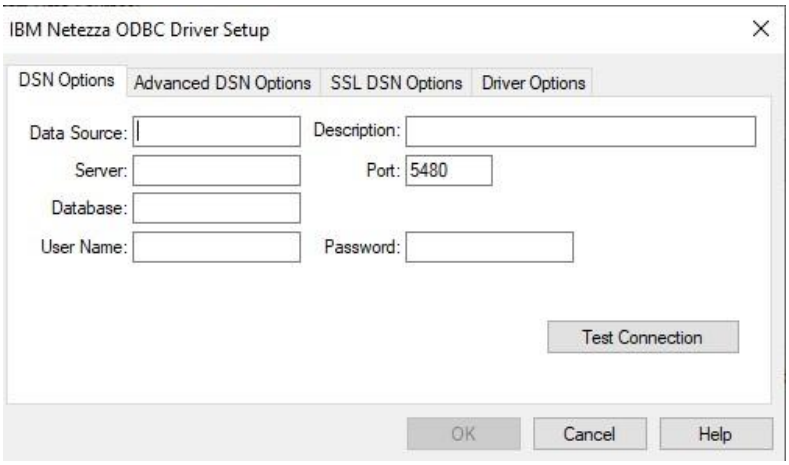
10. Click the Add button to create a new Data Source.

11. Find and select Netezza SQL and click Finish.



12. In ODBC Driver DSN Setup Window, configure the following details:

- Data Source
- Server
- Port
- Database
- User Name
- Password



13. Once completed, launch Server Workbench and login.

14. In the Install Client at Data warehouse pane, select Netezza as the database provider.

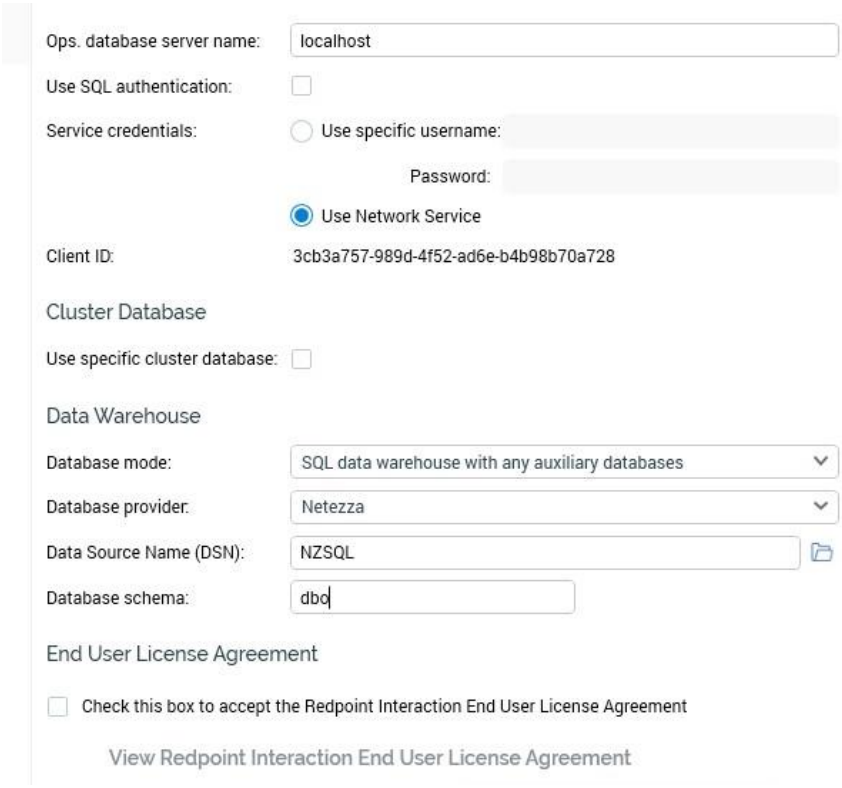
15. Enter the Server name. This can be either the DNS (Domain Name System) or an IP address of the server.

16. Enter the Database provider

17. Enter the Data Source Name

18. Enter the Database schema

19. Check the End User License Agreement checkbox and then click Next

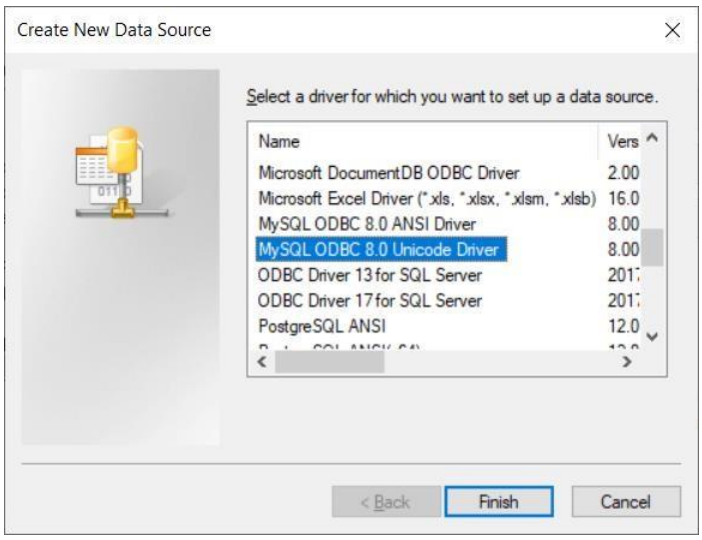

Ops. database server name:
localhost
Use SQL authentication:
Service credentials:
Use specific username:
Password:
Use Network Service
Client ID:
3cb3a757-989d-4f52-ad6e-b4b98b70a728
Cluster Database
Use specific cluster database:
Data Warehouse
Database mode:
SQL data warehouse with any auxiliary databases
Database provider:
Netezza
Data Source Name (DSN):
NZSQL
Database schema:
dbo
End User License Agreement
Check this box to accept the Redpoint Interaction End User License Agreement
View Redpoint Interaction End User License Agreement

MySQL Configuration

This section describes how to create and configure MySQL Data Source Name (DSN). Please follow the steps below. Please note that you may skip steps 1 – 3 if you have already installed the ODBC driver.

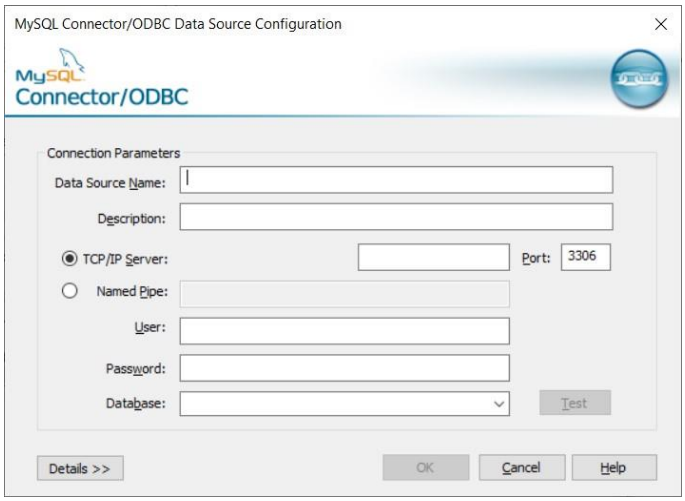
1. In a web browser, navigate to http://www.mysql.com/products/connector/ to download the driver.
2. In the download folder, double click mysql-connector-odbc-8.0.23-winx64.msi.
3. In the MySQL Connector/ODBC 8.0 setup Window, click Next and follow the required steps to install the driver.

4. Once you have successfully installed the driver, go to Control Panel\All Control Panel Items\Administrative Tools and click ODBC Data Sources (64-bit).
5. In the ODBC Data Source Administrator Window, click the System DSN tab.
6. Click the Add… button to create a new Data Source.
7. Find and select MySQL ODBC 8.0 Unicode Driver and click Finish.



8. In the MySQL Connector/ODBC Data Source Configuration Window, configure the following details:

- Data Source Name
- TCP/IP Server: the host server (IP address or a server name).
- Port: the server port number. Only numerical values are accepted. The default server port is 3306.
- User: the database username.
- Password: the database password.
- Database: the database name.



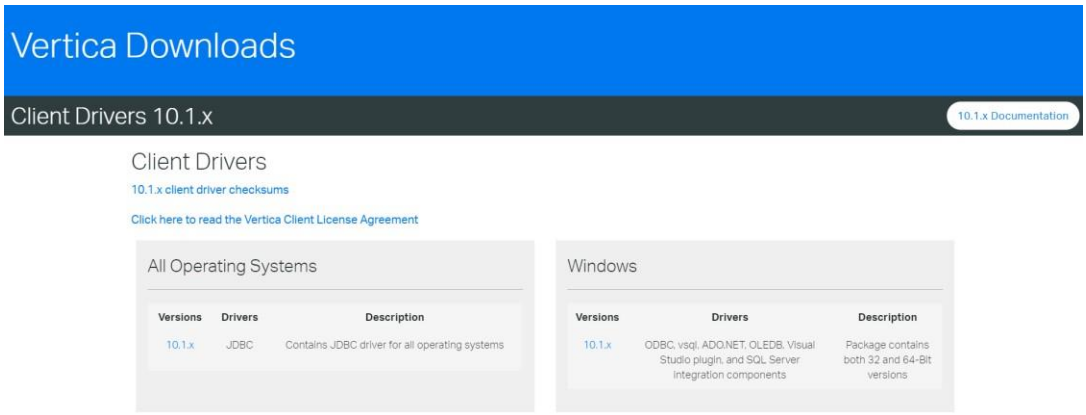
9. Click the Test Connection button. Once the connection has been made successfully, click the OK button to create the DSN.

Vertica Configuration

This section describes how to create and configure the Vertica ADO.Net driver. Please follow the steps below:

1. Go to the official website **https://www.vertica.com/**. Create an account to become a member.

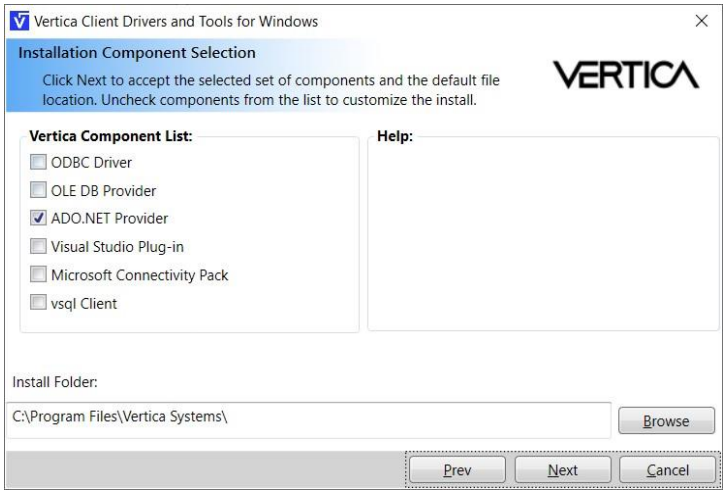
2. Go to **https://www.vertica.com/download/vertica/client-drivers/** and download Client Drivers Package.



3. In the download folder, double click VerticaSetup-10.1.0-0.exe.

4. In the Vertica Client Drivers and Tools setup Window, click Install.

5. Select ADO.Net Provider at component list and follow the installation instructions.



6. Click Finish when completed.

All drivers will now be located in C:\Program Files\Vertica Systems folder.

MarkLogic DB Configuration

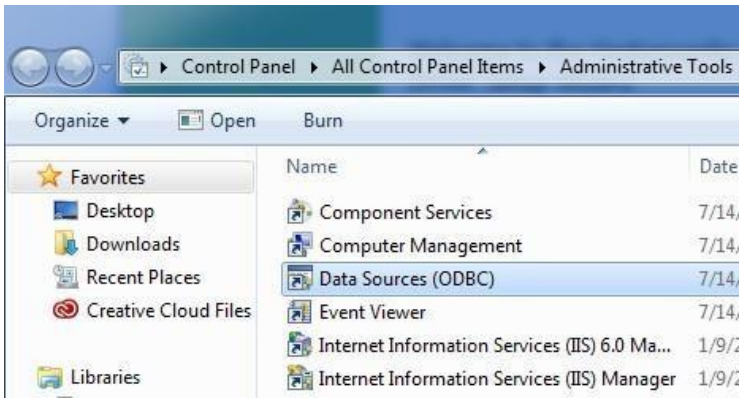
This section describes how to create and configure a MarkLogic DB Data Source Name (DSN). Please follow the steps below:

1. Note: You may skip steps 1-4 if you have already installed the ODBC driver. In a web browser, navigate to **https://developer.marklogic.com/products/odbc/** to download the driver.

2. In the download folder, double click mlsqlodbc-1.4-6-amd64.msi.

3. Follow the steps in the Installation Wizard to install the driver.

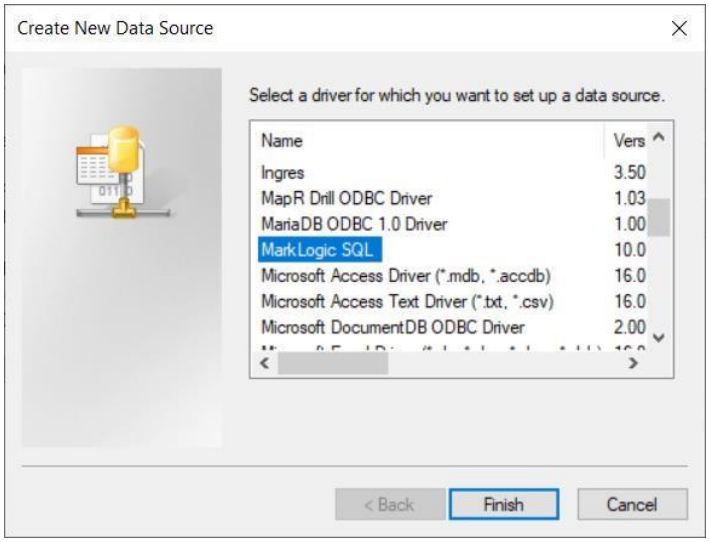
4. Once you have successfully installed the ODBC driver, go to Control Panel\All Control Panel Items\Administrative Tools and click ODBC Data Sources (64-bit).



5. In the ODBC Data Source Administrator window, click the System DSN tab.

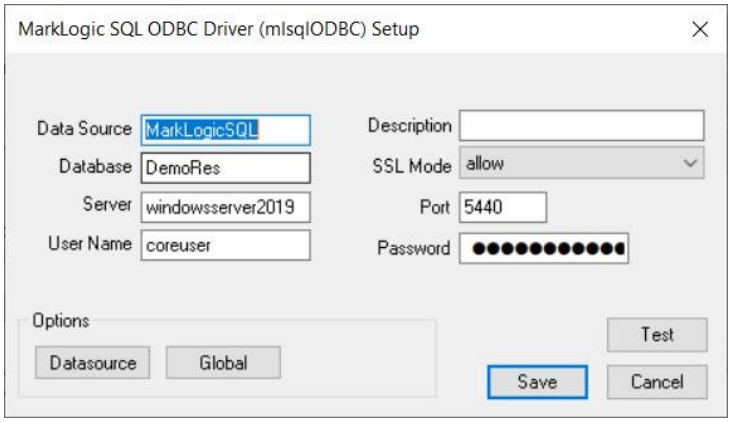
6. Click the Add button to create the new Data Source.

7. Select MarkLogic SQL and click Finish.



8. In the MarkLogic SQL ODBC Data Source Configuration Window, configure the following details:

- Data Source Name
- Database: the database name
- Server: the server name
- Port: the server port number. Only numerical values are accepted. The default server port is 5432
- Set SSL mode to ‘allow'
- Username: the database username
- Password: the database password



9. Click the Test Connection button. Once the connection has been made successfully, click the Save button to create the DSN.

Oracle Configuration

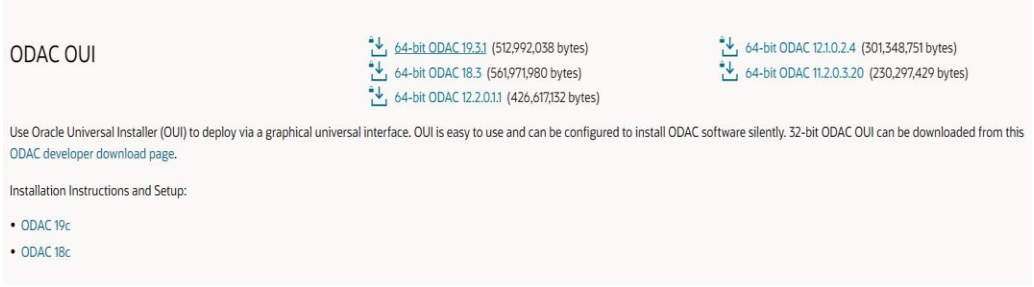
This section describes how to prepare pre-requisites and configure a data connection for an Oracle data warehouse.

1. In a web browser, sign in to **http://www.oracle.com** using your Oracle account. If you don't have an account, please register accordingly.

2. Once you have logged in, click the **Downloads** tab and locate **Drivers**.

3. Find and click **Oracle Data Access Components for Windows**.

4. Click **64-bit ODAC Downloads** and download the appropriate latest version of ODAC.



5. Follow the installation procedures as described.

6. Once installation is complete, launch Interaction Server Workbench and login.

7. In the Data warehouse section in the Install Client overlay, select **Oracle** as the database provider.

8. Enter the server name, which must be in the following format:

```
<Machine name or IP address>:<Port number>
```

9. Enter the database name (the value could be a SID or TNS service name).

10. Enter the database schema.

11. An optional load balancer can also be selected if required.

12. Enter the database's username and password.

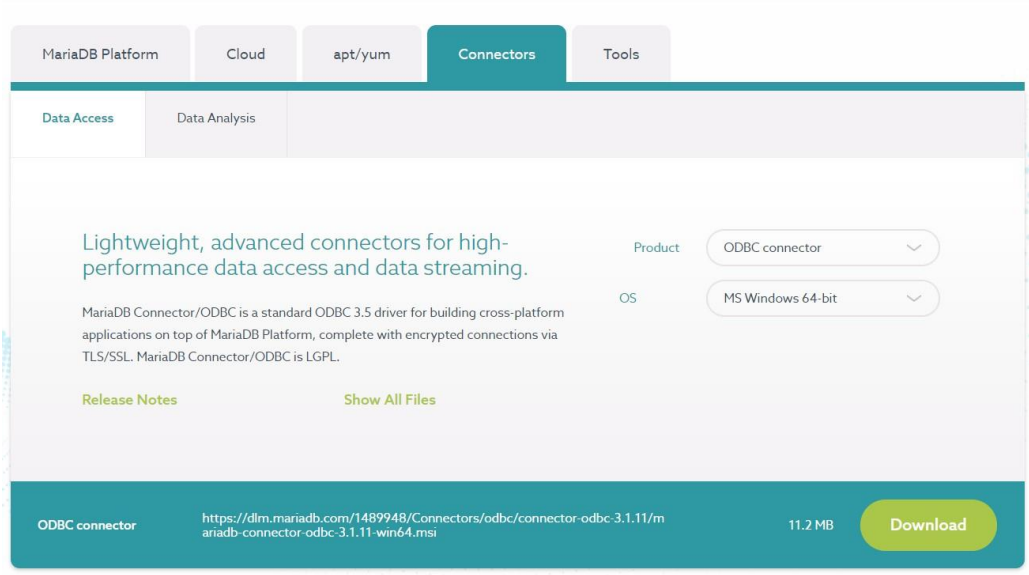
Note that if the OracleTNSAdminRootFolderPath core configuration setting has been set, the Oracle database connection will be sourced from the .ora configuration files in the folder specified by the configuration setting. If EZCONNECT is not specified in the NAMES.DIRECTORY_PATH parameter in the sqlnet.ora configuration file, existing clients using the EZCONNECT connection string format will fail to establish database connections. Any changes made to the Oracle configuration files require a restart of all RPI services to take effect.

MariaDB Configuration

This section describes how to create and configure a MariaDB Data Source Name (DSN).

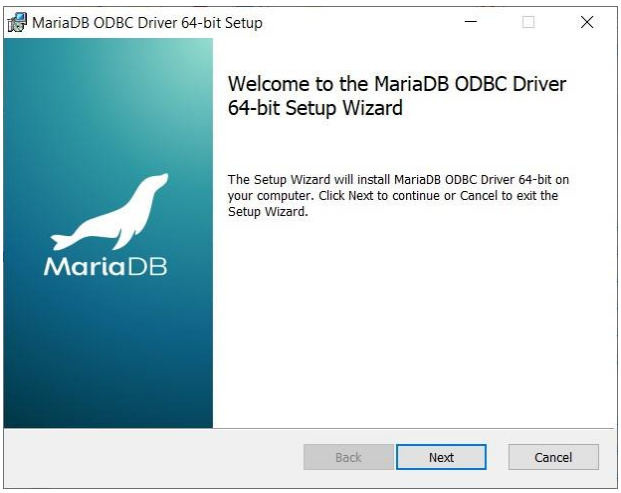
1. In a web browser, navigate to **https://mariadb.com/downloads/#connectors** to download the driver.

2. In the download page, set product to 'ODBC Connector' and OS to 'MS Windows 64-bit'. the installer. Click Download.



3. After downloading the file, go to the downloads folder. Locate and double-click the mlsqlodbc-1.4-6-amd64.msi file.

4. In the MariaDB ODBC Driver 64-bit Setup Window, click Next and follow the required steps to install the driver.

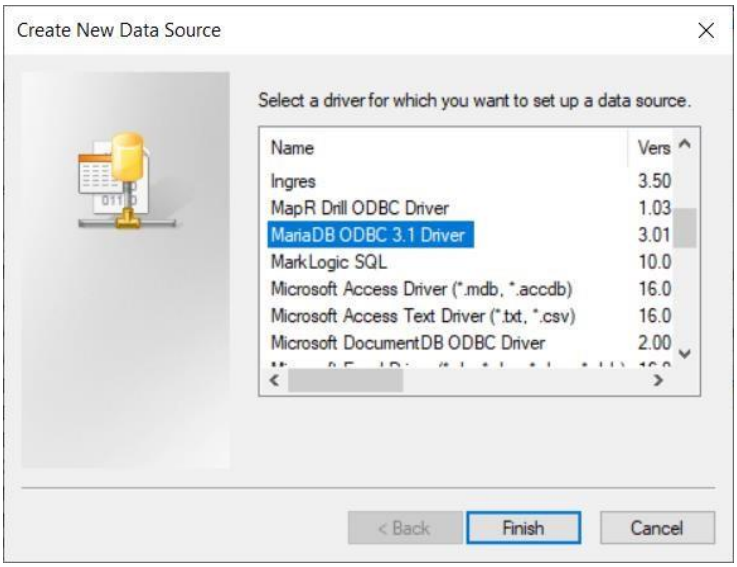


5. Once you have successfully installed the ODBC driver, go to Control Panel\All Control Panel Items\Administrative Tools and click ODBC Data Sources (64-bit).

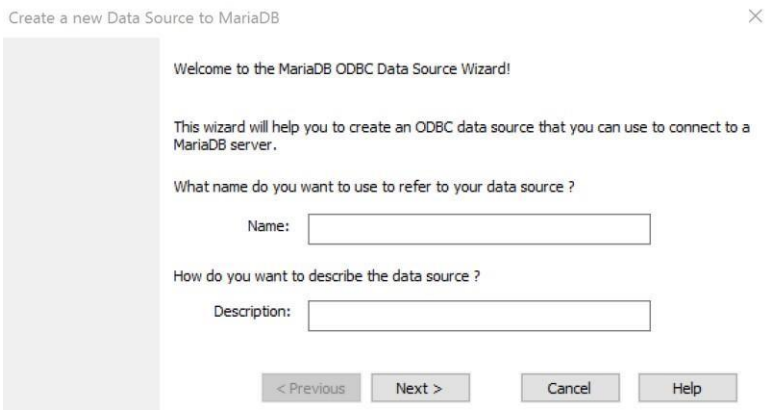
6. In the ODBC Data Source Administrator Window, click the System DSN tab.

7. Click the Add...button to create a new Data Source.

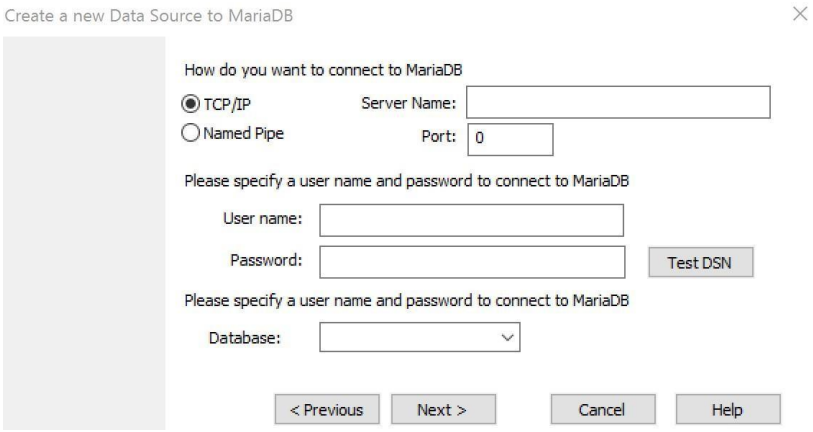
8. Find and select MariaDB ODBC 3.1 Driver and click Finish.



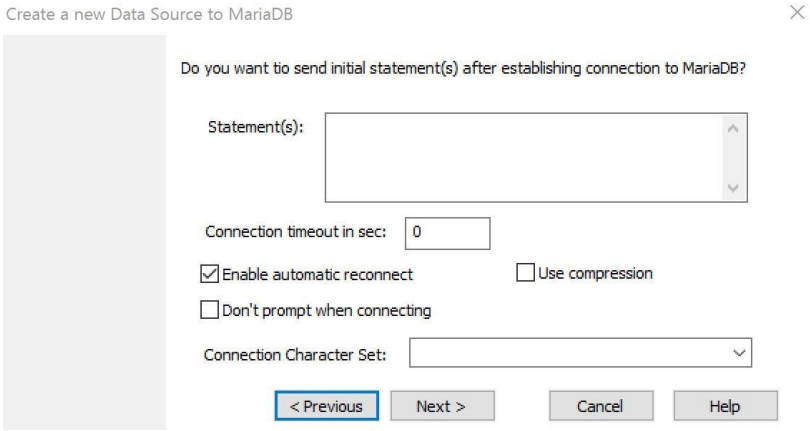
9. In the MariaDB ODBC Data Source Wizard, provide a descriptive name for the Data Source and click the Next button.



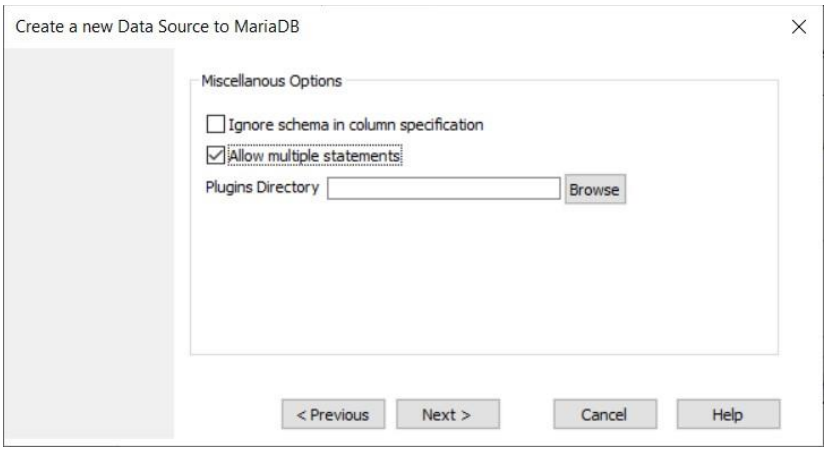
10. Provide the connection parameter details (Server Name, Port, Username, Password and Database). The TCP/IP option must always be selected. The default port is 4006; however, the port might change during configuration of your MariaDB cluster. Click Test DSN to test the connection. Once the connection has been verified, click Next.



11. The Enable automatic reconnect option must always be selected. The other options remain as is. Click Next.



12. The Allow multiple statements option must always be selected. Click Next and click Finish.

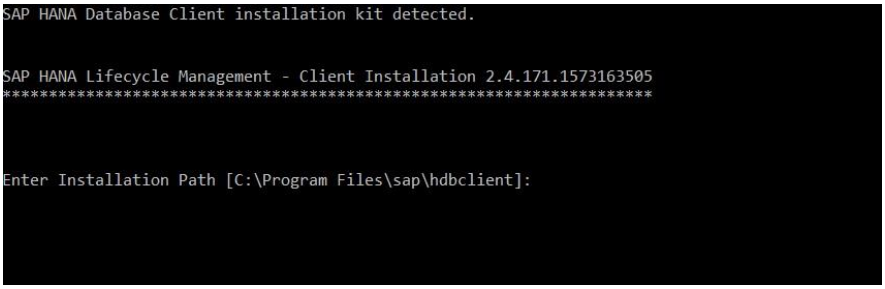


SAP HANA Configuration

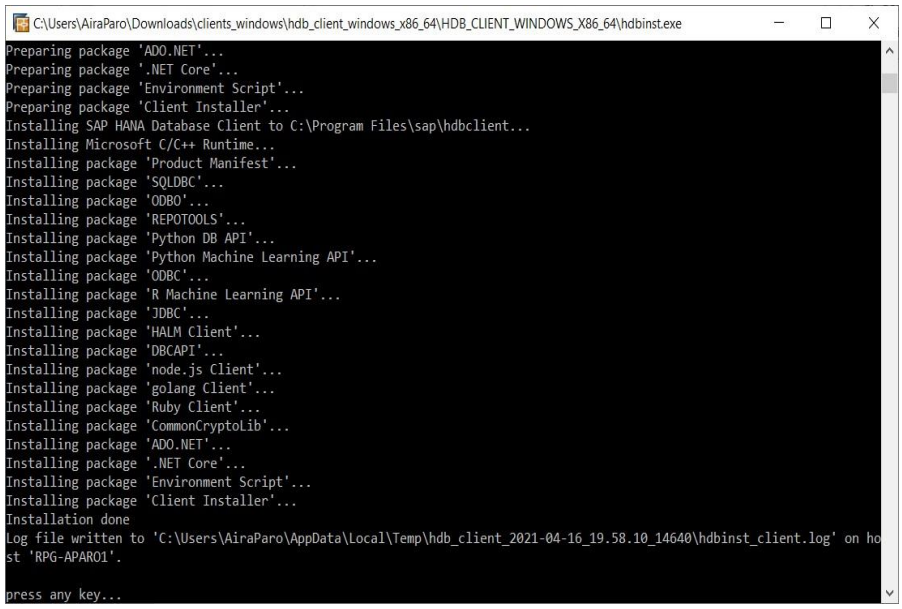
1. To access the driver, navigate to **https://developers.sap.com/trials-downloads.html.** You must be an SAP customer or partner to be able to download the driver.

2. Having downloaded the driver installer, double click hdbinst.exe to start the installation.

3. Press the Enter key to install the driver at a default directory location.



4. Once the installation has completed, press any key to exit the installation.



Google BigQuery Configuration

This section describes how to configure Google BigQuery. Please follow the steps below:

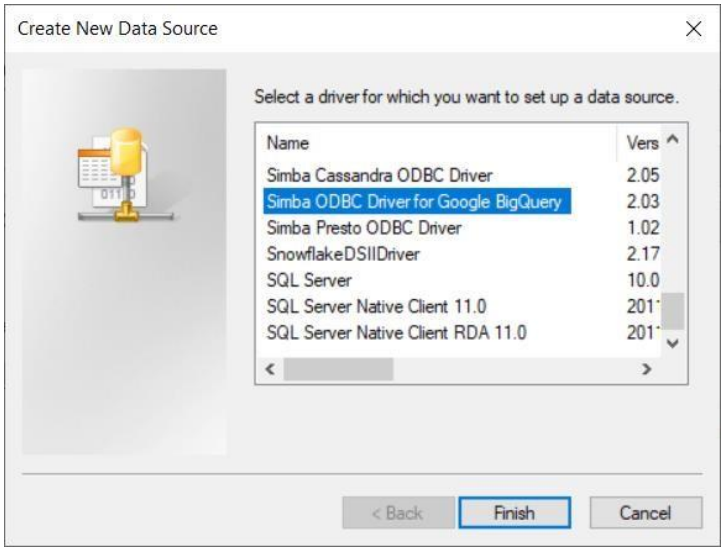
1. Download the driver from **https://cloud.google.com/bigquery/partners/simba-drivers/** and install Windows 64-bit (msi).

   Current ODBC driver release (2.3.3.1005)

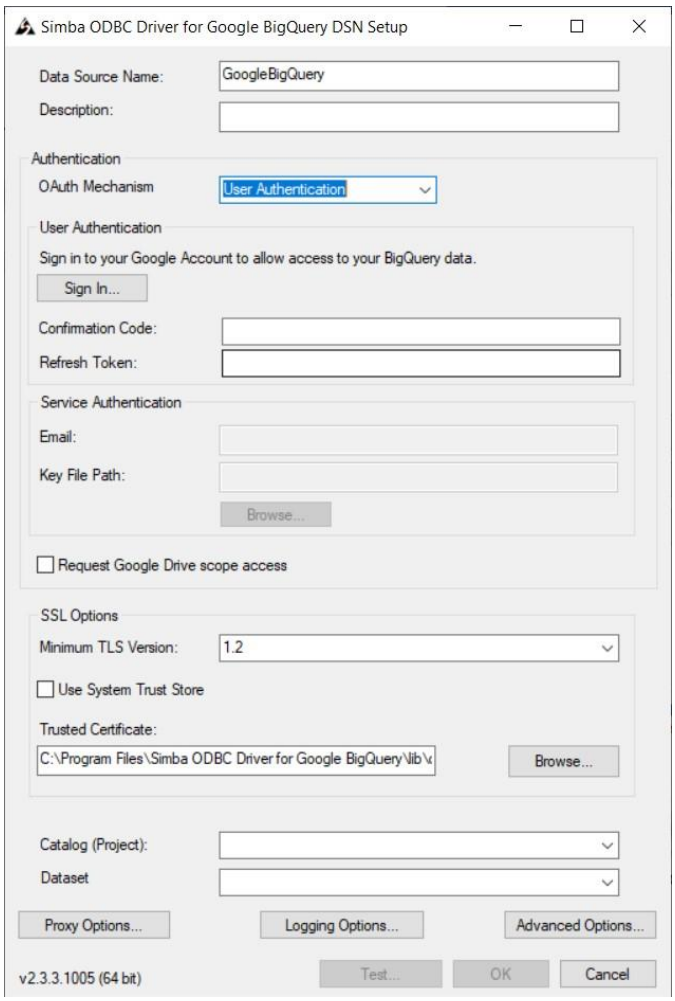
   - Windows 32-bit (msi)
   - Windows 64-bit (msi)

2. Open the ODBC Data Source Administrator, which can be found in Control Panel > Administrative Tools. Select the System DSN tab and click Add.

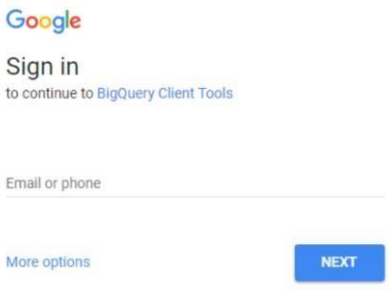
3. Find Simba ODBC Driver for Google BigQuery and click Finish.



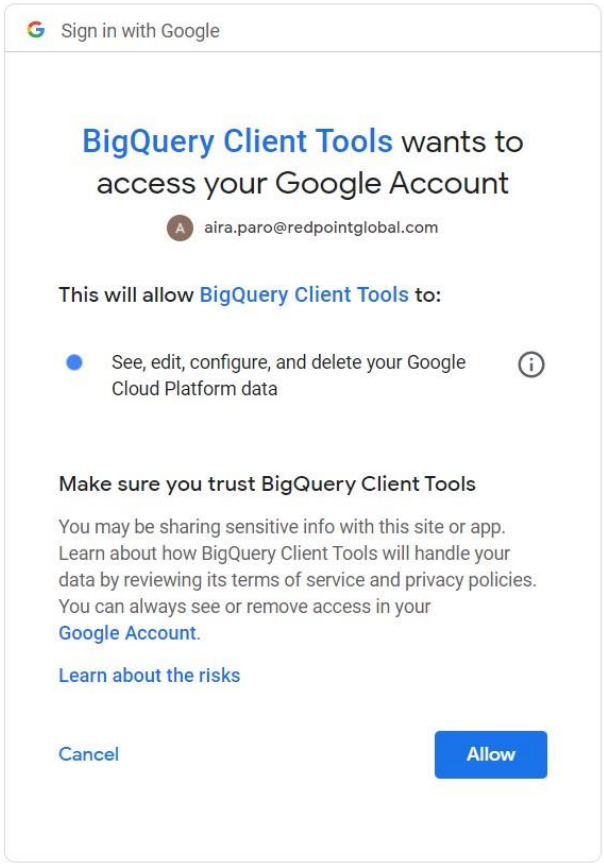
4. In the Simba ODBC Google BigQuery Data Source Configuration Window, click the Sign In... button to log into your Google BigQuery account.



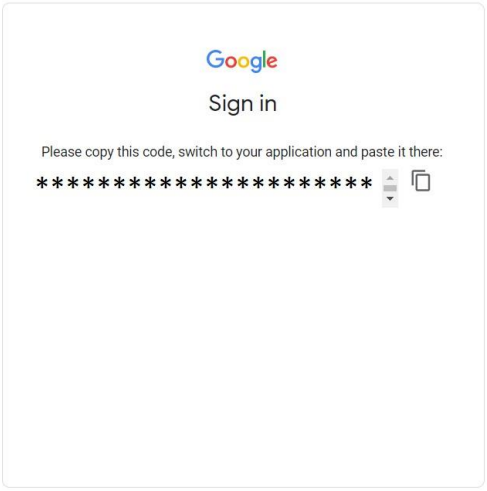
5. In a web browser, enter your email or phone number and your password.

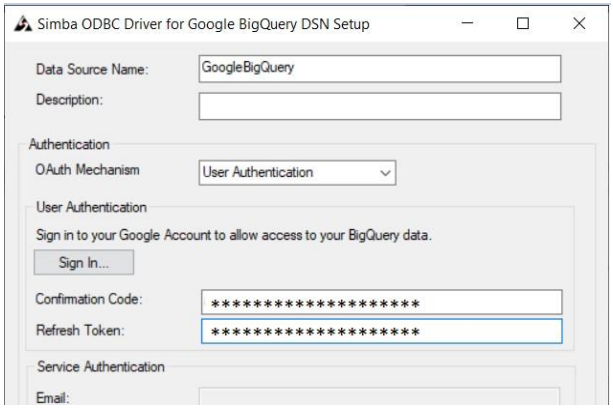


6. In the authorization page, click the Allow button to permit the ODBC driver to access your data.

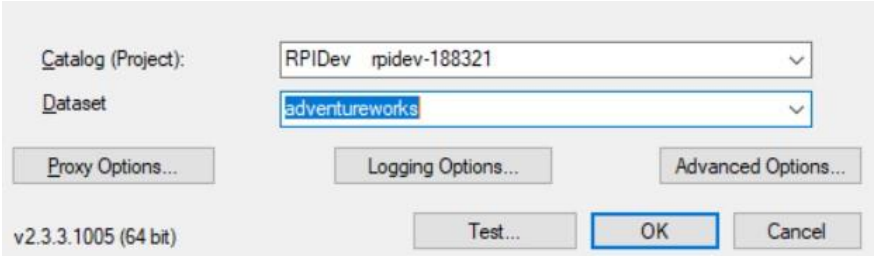


7. Copy the code and paste it into the Confirmation Code in Google BigQuery ODBC DSN settings. The Refresh Token will be automatically populated after you have pasted the Confirmation Code.





8. Select an appropriate project & data set. To test the connection, click the Test… button. Once successful, click OK to close the DSN settings.

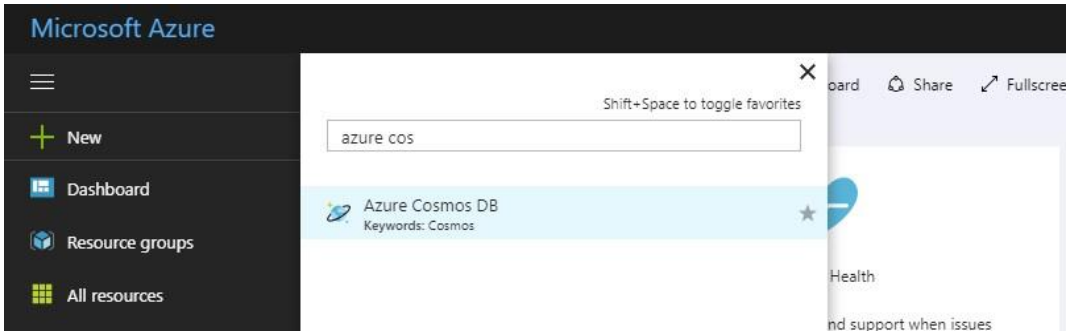


CosmosDB Configuration

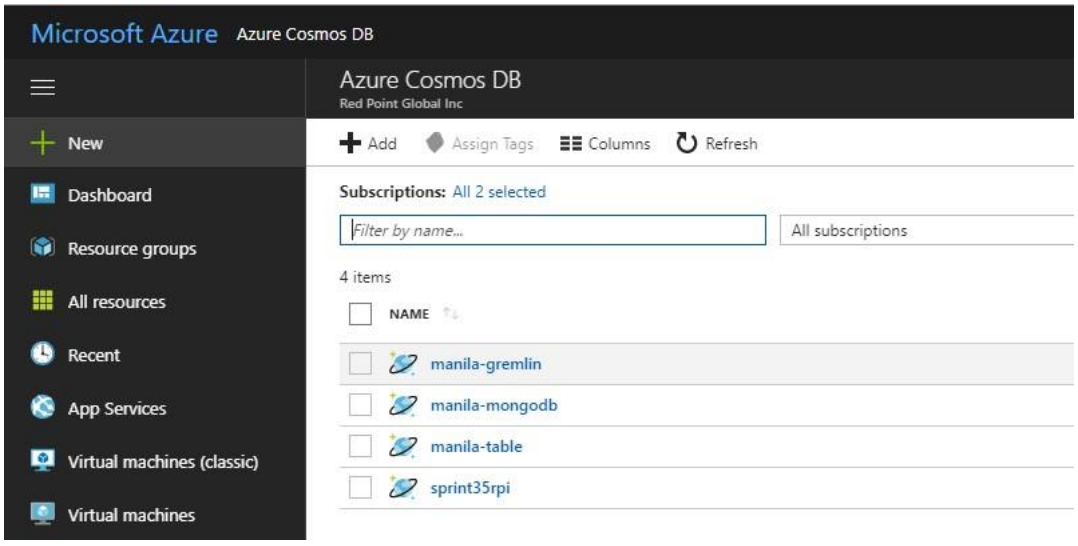
This section describes how to configure CosmosDB. Please follow the steps below:

1. In a web browser, log into https://portal.azure.com.

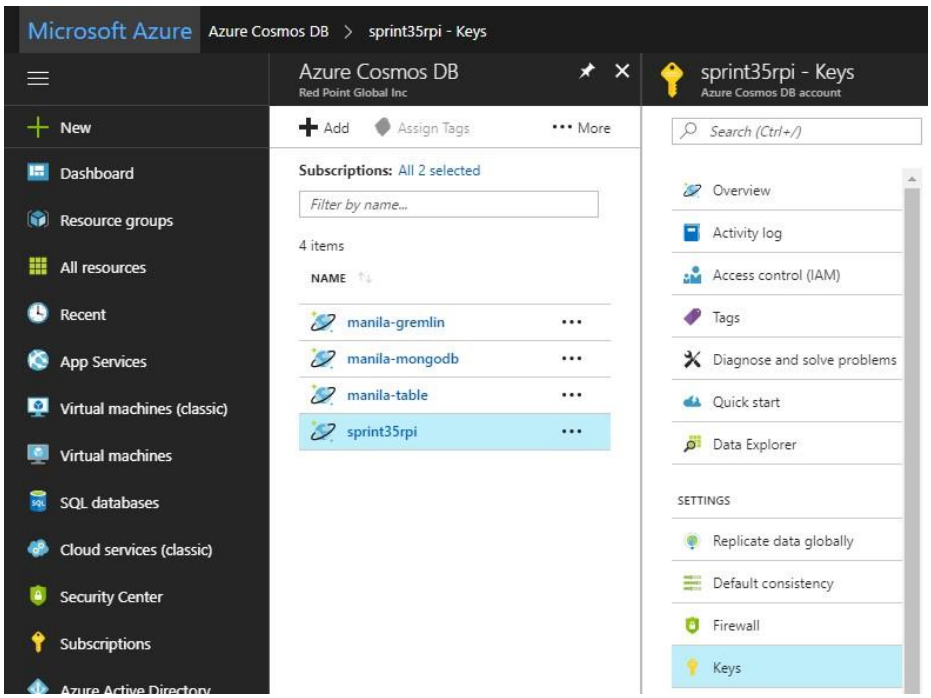
2. Once you have successfully logged into the portal, click More services at the menu navigation to search for the Azure Cosmos DB resource.



3. Click the CosmosDB resource, (in this example "*sprint35rpi*").



4. Under Settings, click *Keys*.



5. Use the URI to configure the Azure Cosmos DB ODBC Host parameter and PRIMARY KEY as the Access key.

6. Download and install the ODBC driver from **https://aka.ms/documentdb-odbc-64x64**.

7. Use “default” as the default value when configuring the Azure Cosmos DB schema in the Server Workbench Manage Auxiliary Databases interface.

Apache Cassandra Configuration

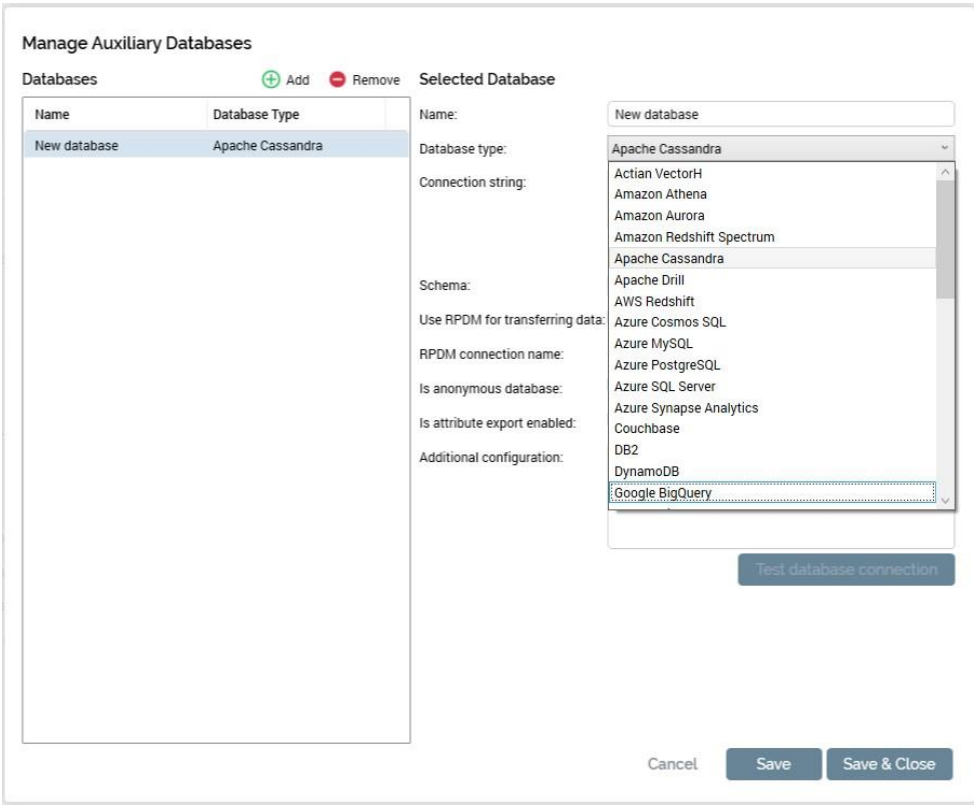
This section describes how to configure data connections for the Apache Cassandra auxiliary database.

1. Launch Interaction Server Workbench and log in.

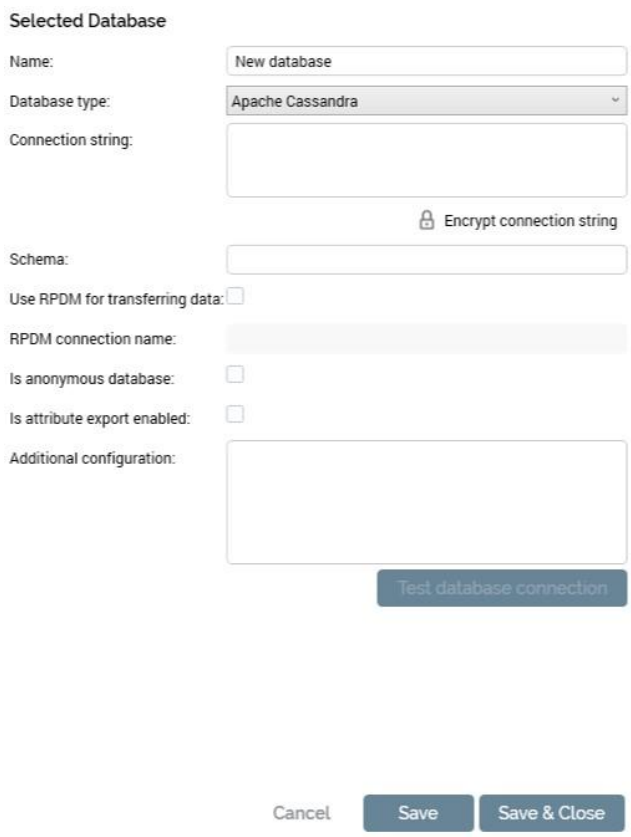
2. Select the Clients tab and select client.

3. Click Manage Auxiliary databases.

4. In Manage Auxiliary Databases, click Add and choose Apache Cassandra.



5. In the Selected Database section, enter the name, connection string and Schema.



6. Connection string must have the following details:

```
Contact Points=<IP Address>;Port=<Port>;Default Keyspace=<schema>
e.g. Contact Points=10.100.68.167;Port=9042;Default Keyspace=demores
```

   To use SSL, make use of the `UseSSLForCosmosDB` parameter.

7. Click Save and click Test database connection to test the connection.

8. Once the connection is successful, Click Save & Close.

Amazon Athena Configuration

1. In a web browser, navigate to **https://docs.aws.amazon.com/athena/latest/ug/connect-with-odbc.html** to download driver.

Amazon Athena ODBC Driver License Agreement

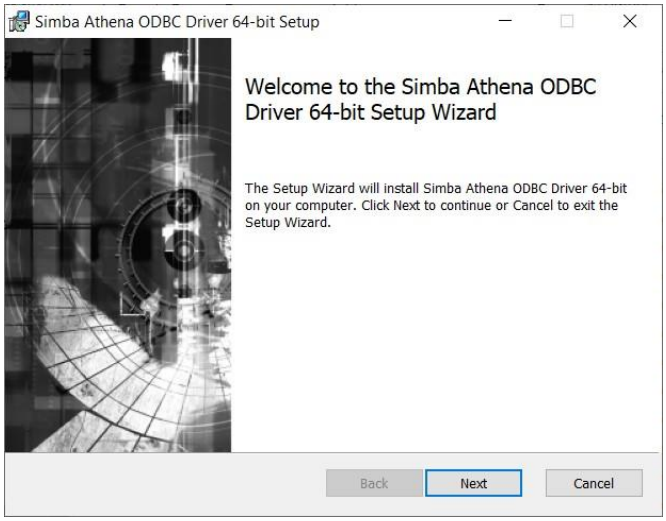
License Agreement

ODBC Driver Download Links

Windows

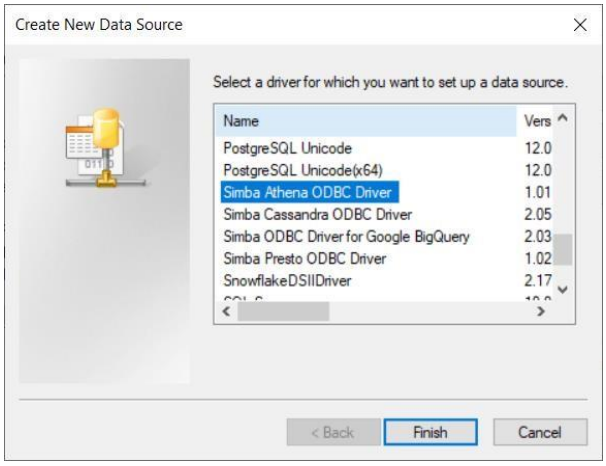
Driver Version	Download Link
ODBC 1.1.6 for Windows 32-bit	Windows 32 bit ODBC Driver 1.1.6
ODBC 1.1.6 for Windows 64-bit	Windows 64 bit ODBC Driver 1.1.6

2. Install Simba+Athena+1.1+64-bit.msi, and follow the instructions provided as the installation progresses.

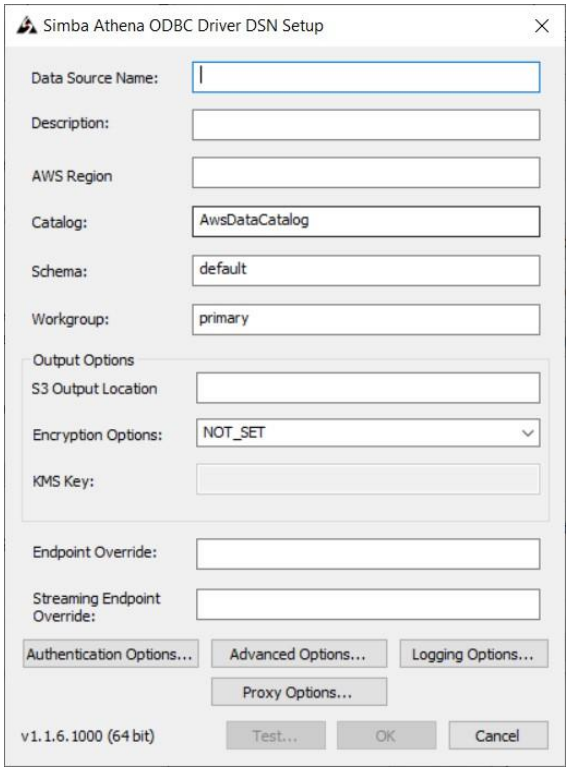


3. Once you have successfully installed the driver, go to Control Panel\All Control Panel Items\Administrative Tools and click ODBC Data Sources (64-bit). Select System DSN Tab and Click Add.. button to create a new Data Source.

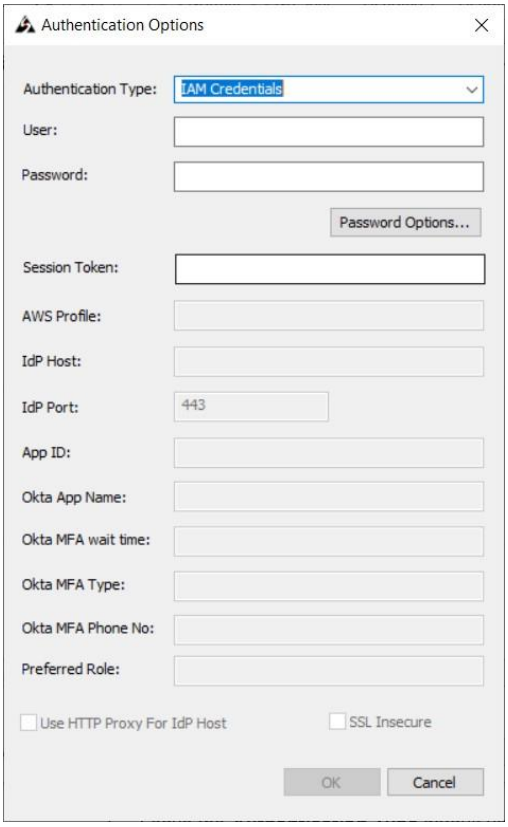
4. Find and select Simba Athena ODBC Driver and click Finish.



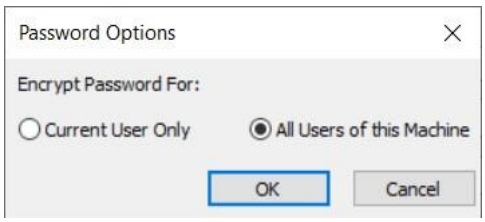
5. In the Simba Athena ODBC Data Source Configuration Window, configure the following mandatory details:
   - Data Source Name
   - AWS Region
   - Schema
   - S3 Output Location



6. Click Authentication Options.. Leave the ‘Authentication Type’ default value as is. Provide the User and Password.



7. Click 'Password Options…'. Make sure the 'All Users of this Machine' is selected. Click *OK*.



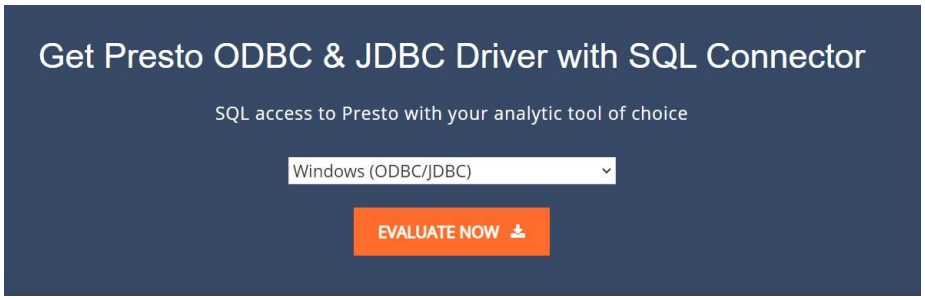
8. Click the Test Connection button. Once the connection has been made successfully, click the OK button to create the DSN.

9. You can also configure an Athena ODBC connection string that can be used in Redpoint Interaction using the following format:

```
Driver=Simba Athena ODBC Driver;AwsRegion=<the
reqgion>;S3OutputLocation=<output location>;AuthenticationType=IAM
Credentials;UID=<User>;PWD=<password>;
```

Presto Configuration

1. In a web browser, navigate to https://www.simba.com/drivers/presto-odbc-jdbc to download the driver.



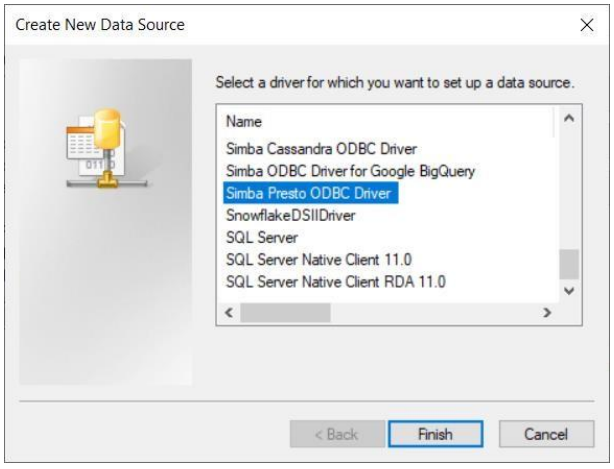
2. Install Magnitude Gateway Desktop 64-bit.msi, following the instructions provided as the installation progresses.

3. Once you have successfully installed the driver, go to Control Panel\All Control Panel Items\Administrative Tools and click ODBC Data Sources (64-bit).

4. In the ODBC Data Source Administrator Window, click the System DSN tab.

5. Click the Add… button to create a new Data Source.

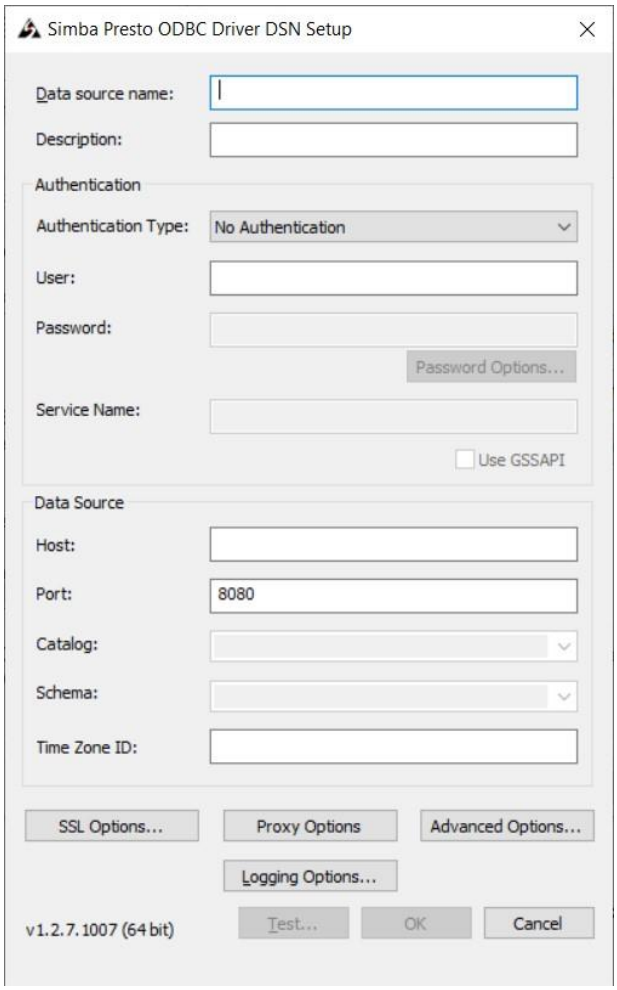
6. Find and select Simba Presto ODBC Driver and click Finish.



7. In the ODBC Data Source Configuration Window, provide all mandatory connection details for Presto:

   - Data source name
   - Host

- Port
- Catalog
- Schema



8. Click the Test Connection button. Once the connection has been made successfully, click the OK button to create the DSN.

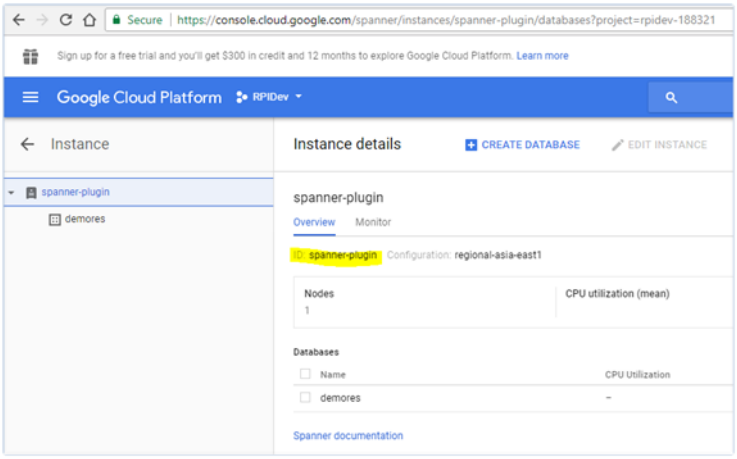
Google Spanner Configuration

Obtaining a Spanner Instance ID

1. In a web browser, log into the Google Cloud console at **https://console.cloud.google.com**.

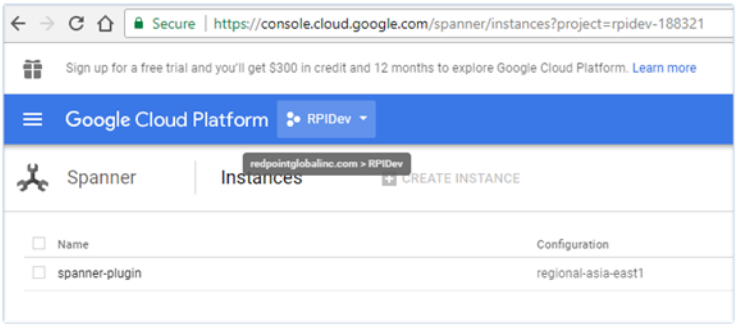
2. In the dashboard navigation menu, click Spanner.

3. Once the Spanner page is displayed, click the Spanner instance and take note of the ***Instance ID*** (you will use this to configure the Spanner plugin in Server Workbench).

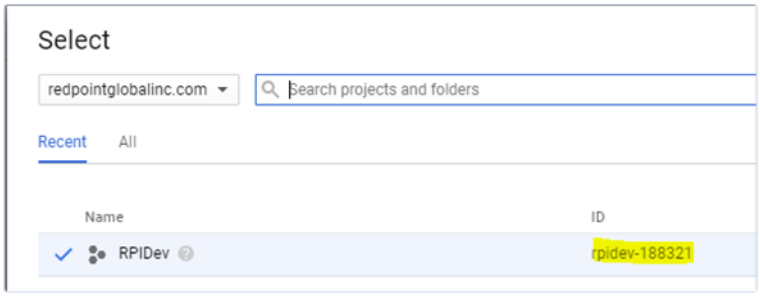


Obtaining a Google Project ID

1. While in the Google Cloud console, click the project selector at the right side of the Google Cloud Platform interface.



2. Once the project selector is displayed, take note of the currently-selected project ID (you will use this to configure the Spanner plugin in Server Workbench).



Download and Install the Google Cloud SDK

1. In a browser, navigate to the Google Cloud SDK download page at **https://cloud.google.com/sdk/downloads**.

2. Locate the download link for Windows 64-bit (*x86_64*). The Spanner plugin was developed and tested using the SDK that can be download here: **https://dl.google.com/dl/cloudsdk/channels/rapid/downloads/google-cloud-sdk182.0.0-windows-x86_64.zip**.

3. Once downloaded, install the Google Cloud SDK. Follow the steps provided as the installation progresses.

Configure the Google Application Default Credentials

1. Log in to **https://console.cloud.google.com**.

2. Go to the API Console Credentials page by navigating to ***APIs & services*** > ***Credentials***.

3. From the project drop-down, select your project.

4. On the Credentials page, select the Create credentials drop-down, then select Service account key.

5. From the Service account drop-down, select App engine app default service account.

6. At Key type, select the JSON key option, then select Create. The file will automatically be downloaded to your computer.

7. Place the **.json* file you have just downloaded within your preferred directory location. The directory must be secured, but accessible to your web server code.

8. Set the environment variable GOOGLE_APPLICATION_CREDENTIALS to the path of the downloaded JSON file.

Creating the Environment variable for your Google Credentials

1. Navigate to Windows ***Control Panel*** > ***System*** > ***Advanced System Settings*** > ***Advanced Tab*** > ***Environment Variables***.

2. In System Variables, add a new variable using the following details:

   Name: GOOGLE_APPLICATION_CREDENTIALS
   Variable value: C:\[PATH TO PRIVATE KEY FILE]\gcp-xxxxxxx-privatekey.json

   Note – the variable value must be set according to the actual directory location of the private key file.

3. Having completed the above, use the following connection string format when provisioning a new client utilizing Spanner in Server Workbench:

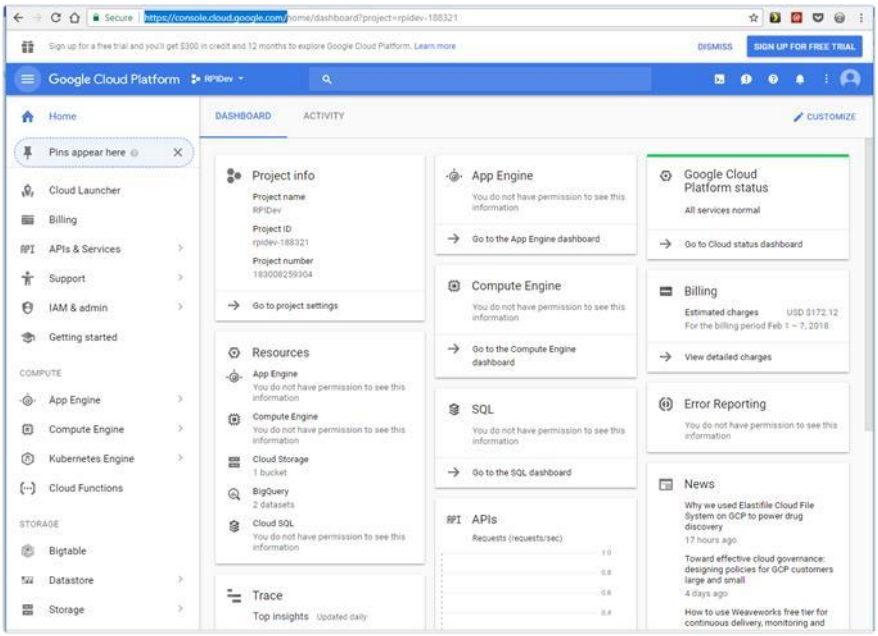
   Instance ID: <GOOGLE_PROJECT_ID>/<GOOGLE_SPANNER_INSTANCEID>
   Database ID: The Spanner Database to be accessed Schema:
   The schema of the Spanner database

Google BigTable Configuration

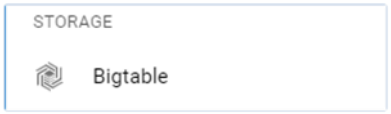
Obtaining a BigTable instance ID

1. In a web browser, navigate to **https://console.cloud.google.com** to log into the Google Cloud console.

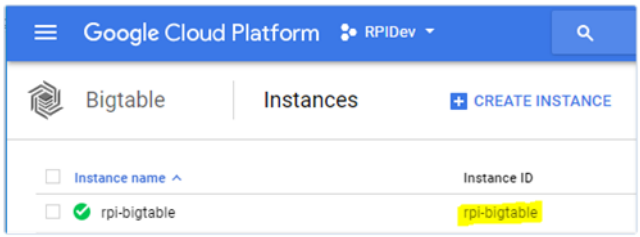
2. Once you have successfully logged on, you will be redirected to the portal's main page:



3. In the dashboard navigation menu, click **BigTable**.

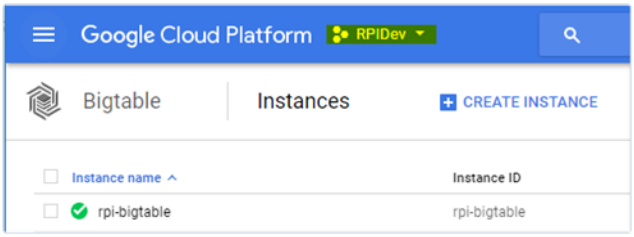


4. Once the BigTable interface is displayed, take note of the BigTable Instance ID, as this will be used to configure Google BigQuery.

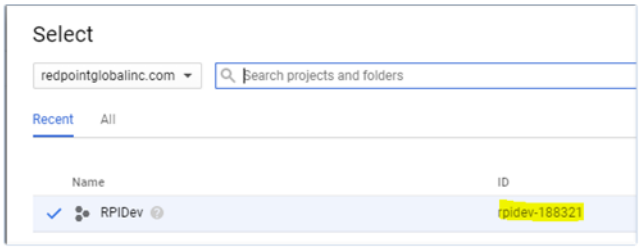


Obtaining a Google Cloud project ID

1. While in the Google Cloud console, click the project selector to the right side of the Google Cloud Platform interface.

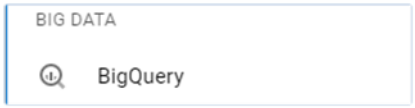


2. Once the project selector is displayed, take note of the currently-selected project ID, as this will be used to configure Google BigQuery.

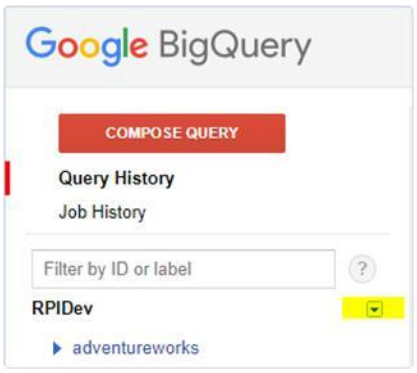


Creating a Google BigQuery Dataset

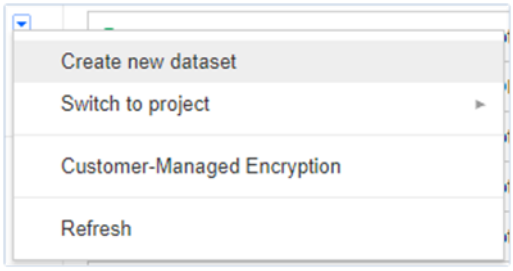
1. While in the Google Cloud console, click ***BigQuery***. You will be redirected to the Google BigQuery portal.



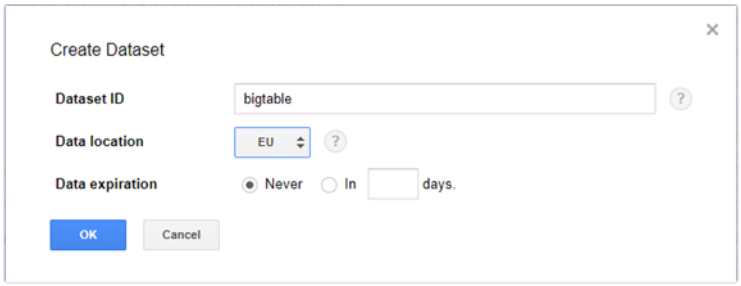
2. Click the small down arrow icon to the lower right corner.


Google BigQuery
COMPOSE QUERY
Query History
Job History
Filter by ID or label
RPIDev
adventureworks

3. Click ***Create new dataset***.

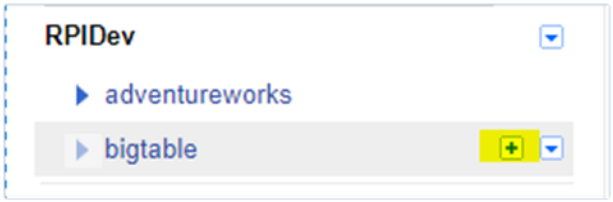


4. Enter your desired dataset name, and also set your preferred data location. Click ***OK***.

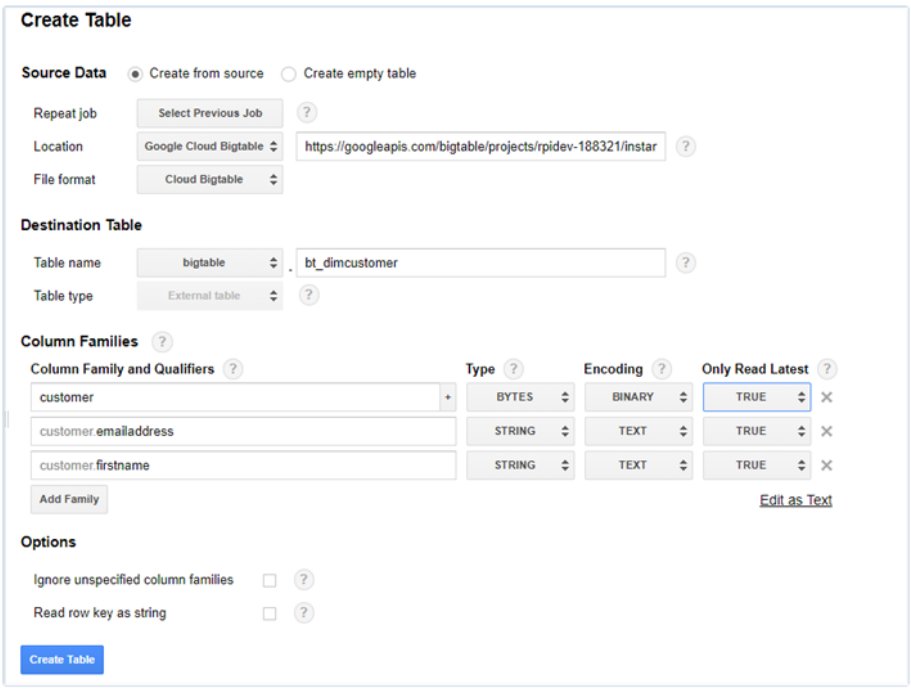


Creating a Google BigQuery Table

1. While in the Google BigQuery portal, hover over the target dataset where you want to create a table. Click the (+) icon to the right side of the dataset name to create a new table.



2. Configure your table as shown below.



3. Select the location "***Google Cloud BigTable"***.

4. Specify the Google Cloud BigTable location source in the following URL format:

   https://googleapis.com/bigtable/projects/[YOUR-PROJECT-ID]/instances/[YOUR-BIGTABLE-INSTACEID]/tables/YOUR-BIGTABLE-TABLE-NAME

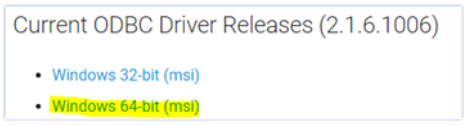
5. Configure your table name under Destination Table.

6. Configure your columns under Column Families.

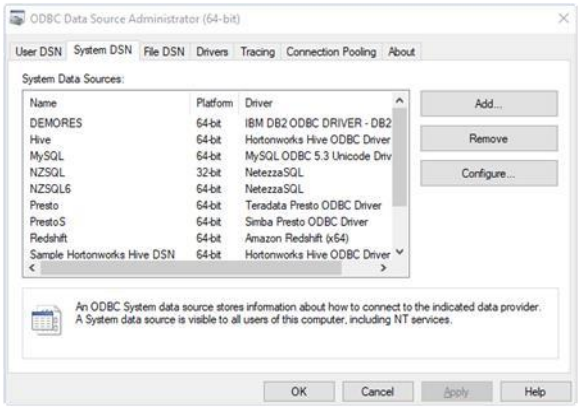
7. Click ***Create Table***.

Installing Google BigQuery ODBC driver for BigTable

1. In a web browser, navigate to **https://cloud.google.com/bigquery/partners/simbadrivers**. Install the Windows 64-bit msi, and follow the instructions provided as the installation progresses.



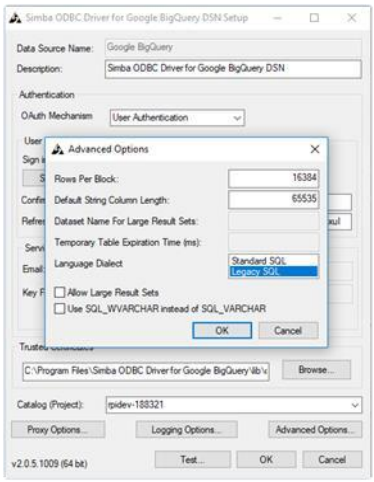
2. Open the ODBC Data Source Administrator, which can be found in ***Control Panel*** > ***Administrative Tools***. Select the ***System DSN*** tab and click ***Add***.



3. Find the ***Simba BigQuery ODBC driver*** and click ***Finish***.

4. Provide all mandatory connection details for BigTable.

5. Make sure to set the ODBC language dialect to Legacy SQL. This is to make sure to get the complete schema for tables in the context of BigTable.



1. Once all set, you may want to test the connection by clicking ***Test***. Otherwise, click ***OK*** to save your settings.

Snowflake Configuration

This section describes how to prepare pre-requisites and configure data connections for the Snowflake data warehouse.

1. In a web browser, navigate to **https://docs.snowflake.com/en/user-guide/odbc.html/** to download the ODBC driver.

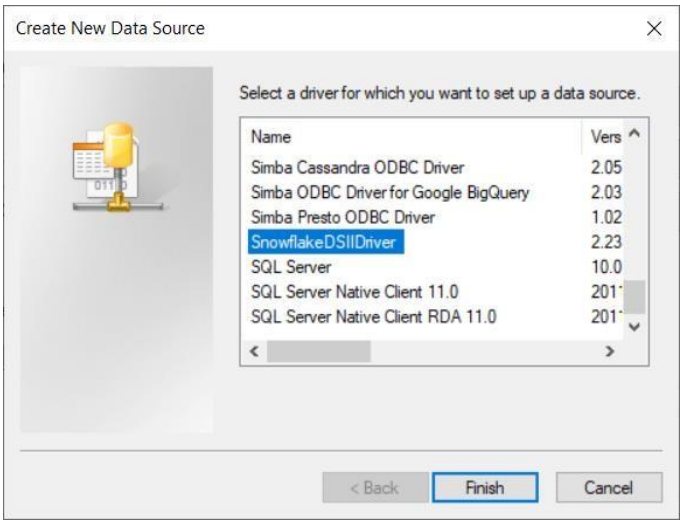
2. In the download folder, double click 'snowflake64_odbc-2.23.1.msi' and follow the installation steps.

3. Once you have successfully installed the driver, go to Control Panel\All Control Panel Items\Administrative Tools and click ODBC Data Sources (64-bit).

4. In the ODBC Data Source Administrator Window, click the System DSN tab.

5. Click the Add… button to create a new Data Source.

6. Find and select SnowflakeDSII Driver and click Finish.



7. In the Snowflake Configuration dialog, supply all the required parameters.

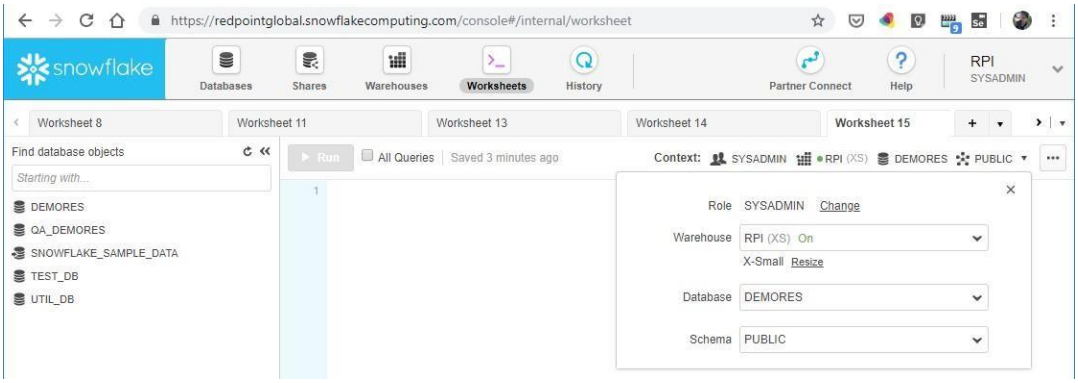
   The Server parameter should be set to *<Snowflake_account>.snowflakecomputing.com* (where '*Snowflake_account'* is a valid Snowflake account name).

   Set the Tracing parameter to 6. Finally, set the NoProxy parameter to *.amazonaws.com

8. To get details of the required parameters in Step #7, in a web browser, log into your Snowflake account. The login link is in the following format:

   Error! Hyperlink reference not valid.

9. Once logged in, click ‘WorkSheets’, and click ‘Context’ settings.



10. Make sure that you have access to a Warehouse, Database and Schema. Take note of the settings here as you will use this to configure the Snowflake DSN parameters.

11. Return to step 6. Once all set, click OK.

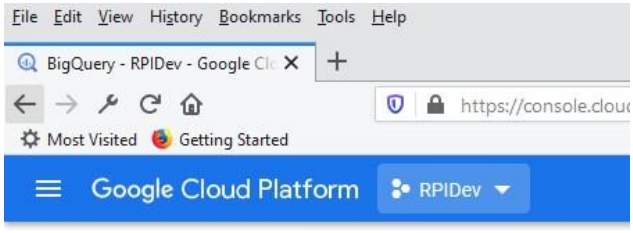
Google Datastore Configuration

This section describes how to prepare pre-requisites and configure data connections for the Google Datastore NoSQL auxiliary database.

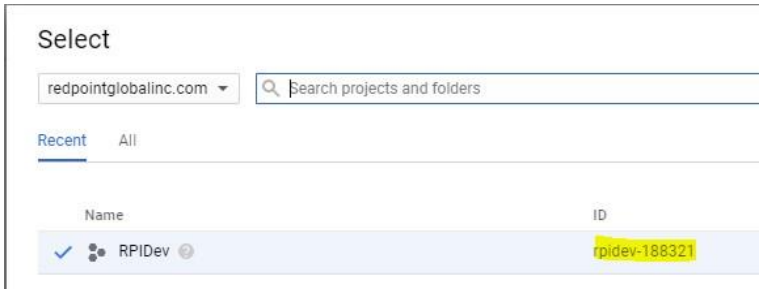
Obtaining a Google Project ID

1. In a web browser, log into the Google Cloud console at **https://console.cloud.google.com**.

2. Select a project using the dropdown to the right of the Google Cloud Platform header.



3. Once the selected project is displayed, take note of the currently selected project's ID, as it is required when configuring the Google Datastore connection in Server Workbench.



Configuring Google Application Default Credentials

1. In the GCP console, go to the API Console Credentials page by navigating to APIs & services > Credentials.

2. Select your project from the project drop-down.

3. In the Credentials page, select the Create credentials drop-down, then the Service account key.

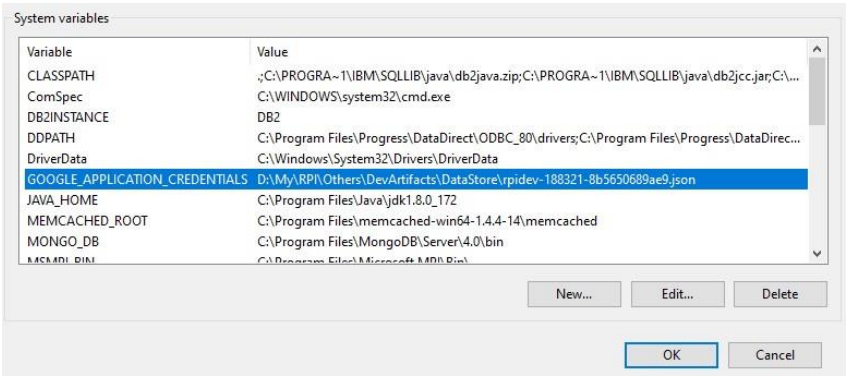
4. From the Service account drop-down, select the App engine app default service account.

5. For Key type, select the JSON key option, then select Create. The file is automatically downloaded.

6. Place the downloaded *.json file into your preferred local directory location. The directory must be secure, but accessible to your web server code.

Creating a Google Credentials Environment Variable

1. Navigate to Windows Control Panel > System > Advanced System Settings > Advanced Tab > Environment Variables.



2. In System Variables, add a new variable as follows:

   Name: GOOGLE_APPLICATION_CREDENTIALS
   Value: C:\[PATH TO PRIVATE KEY FILE]\<project id>-xxxxxxxxxxxx.json

   Note that the variable value must be set according to the actual directory location of the private key file.

3. Use the following connection string format when provisioning an auxiliary database connection utilizing Google DataStore in Server Workbench:

   {"ProjectId": "Project ID", "Namespace":"Database Name"}

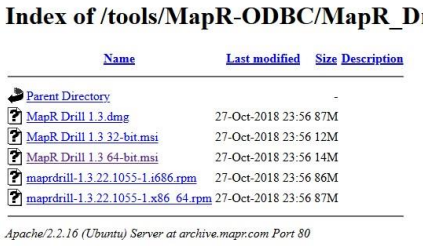
MapR/Apache Drill DB Configuration

This section describes how to create and configure a MapR Apache Drill Data Source Name (DSN). Please follow the steps below:

1. (Note: you may skip this step if you have downloaded the ODBC driver) In a web browser, navigate to https://mapr.com/docs/60/Drill/drill_odbc_connector.html#drill_odbc_connector to download the driver.

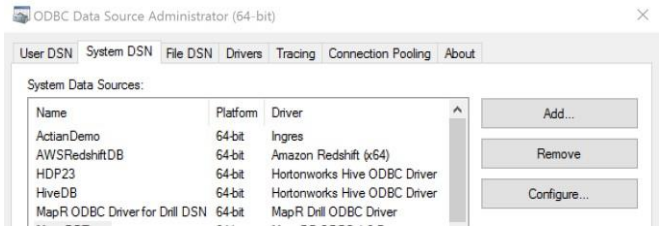
Drill Version	ODBC Version
1.15.0	1.3.22.1055

2. In the download page, click MapR Drill 1.3 64-bit.msi to download the installer.

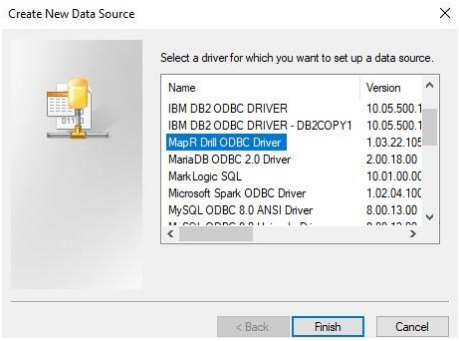


3. After downloading the file, locate the download and double click the file named MapR Drill 1.3 64-bit.msi. Follow the Setup Wizard steps.

4. To configure a new MapR Apache Drill DSN, open Control Panel. Under the Administrative group, click ODBC Data Source. Select the System DSN tab and click Add to create a new DSN.



5. Select MapR Drill ODBC Driver and click Finish.



6. In the MapR Drill ODBC Data Source dialog window, provide the details of the following parameters:
   - Data Source Name
   - Direct to Drillbit
   - Default Schema

Amazon DocumentDB Configuration

This section describes how to prepare pre-requisites and configure data connections for the Amazon DocumentDB NoSQL auxiliary database.

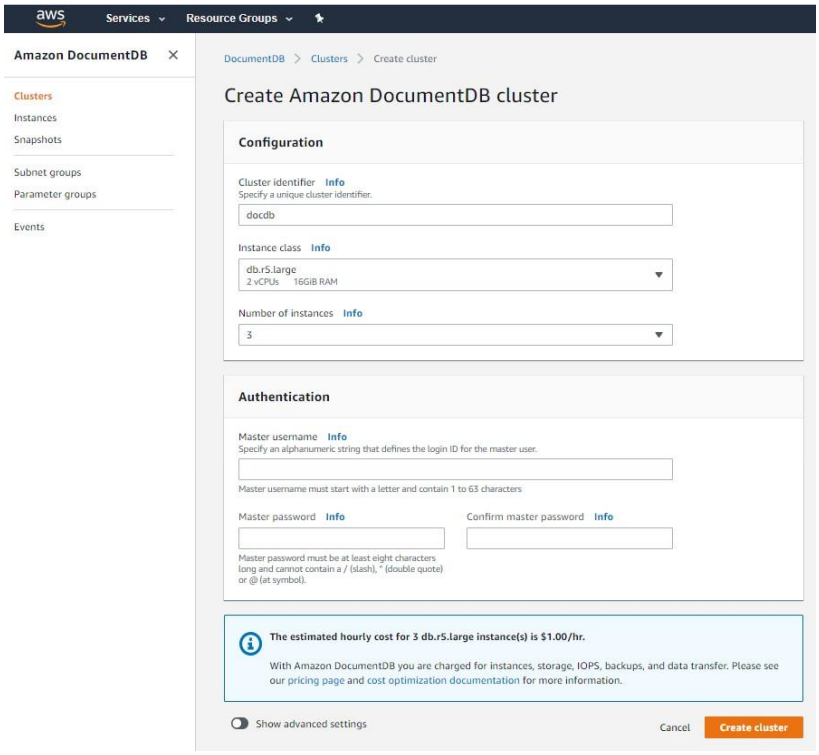
1. In a web browser, log into the AWS console at:

   https://<account id>.signin.aws.amazon.com/console

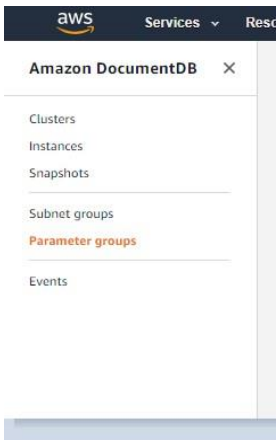
   ...replacing '<account id>' with your AWS account ID.

2. In the AWS console home page, search for Amazon DocumentDB.

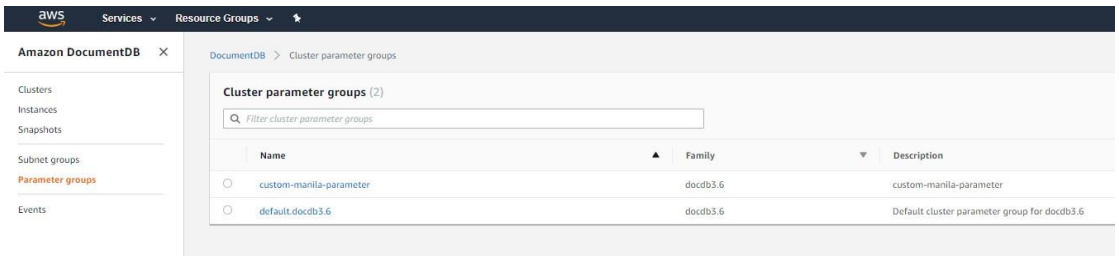
3. In the Amazon DocumentDB home page, click Clusters to create new cluster. The Clusters option is located on the left navigation menu. Provide all the mandatory parameters for the new cluster and click Create cluster.



4. Once the cluster has been created, click Parameters Group to create new a custom parameter group.

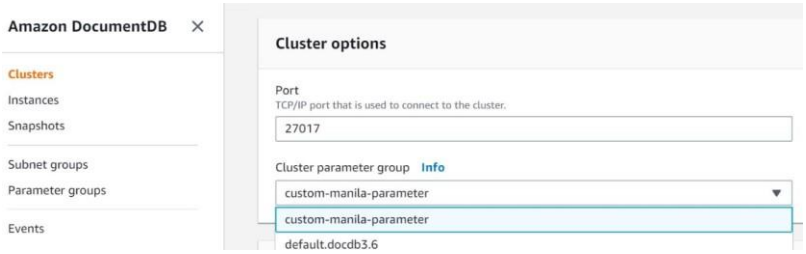


5. Provide a descriptive name and click Create. See the example below.

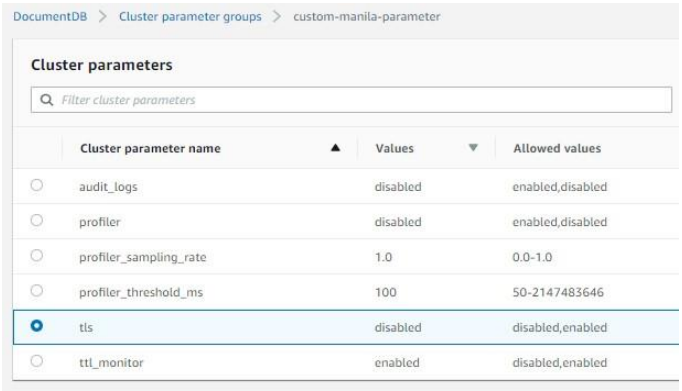


6. You must associate your custom parameter group with your Amazon DocumentDB clusters. To associate a custom parameter group to a cluster, click Clusters on the navigation menu. Locate the instance in the list of instances and click the button to the left of its name. Click Modify.

7. Select your custom parameter group in the Cluster parameter group dropdown box.



8. To disable the TLS option for the cluster, click Parameter groups on navigation menu. Double-click your custom parameter group in the list of Clusters parameter groups. Find the 'tls' option and select disabled in the Modify tls dialog.



Click Modify cluster parameter to save the changes.

9. Once TLS has been disabled, you must reboot the Amazon DocumentDB cluster for the change to take effect. To reboot the Amazon DocumentDB cluster, click Instances on the left navigation menu. Locate the instance in the list of instances and click the button to the left of its name. Select Actions > Reboot.

10. The following connection string format should be used when configuring an Amazon DocumentDB auxiliary database in the Server Workbench Clients interface:

```
{
   "ConnectionString": "mongodb://<username>:<password>@<cluster_name>.<AWS-
Region>.docdb.amazonaws.com:27017?ssl=false&replicaSet=rs0",
   "DatabaseName": "the_name_of_database" }
```

Azure SQL Managed Identity Configuration

This section describes how to prepare Azure SQL Managed Identity for use with RPI.

Prerequisites

- Grant the VM access to Azure SQL Database.
- Enable Azure AD authentication.
- Create a contained user in the database that represents the VM's system assigned identity.
- Get an access token using the VM identity and use it to query Azure SQL Database.
- Additional details can be found at:

  https://docs.microsoft.com/en-us/azure/active-directory/managed-identities-azureresources/tutorial-windows-vm-access-sql

Steps

1. Log into SSMS using the AD account and run the following query in the primary data warehouse:

   ```
   CREATE USER [<VMName-FQDN>] FROM EXTERNAL PROVIDER
   ```

   Run this query for all VMs in the cluster. This will create the contained user for each VM.

2. Run the following query in the primary data warehouse:

   ```
   ALTER ROLE db_datareader ADD MEMBER [<VMName-FQND>]
   ```

   Again, repeat his query for all VMs in the cluster. This will grant the VM access to read the entire database.

3. Repeat steps 1 and 2 for all RPI operational databases:

   - Interaction_XXX
   - InteractionAudit_XXX
   - Pulse
   - Pulse_Logging

4. Open the Node Manager config file:

C:\Program Files\RedPoint Global\RedPoint Interaction\Node Manager\Resonance.NodeManagerService.exe

Edit the connection strings as per the following (replacing the server name):

```
<connectionStrings>

    <add name="LoggingDatabase" connectionString="Server=
<AzureSQLDBServer>;Database=Pulse_Logging;UID=AnyString;Encrypt=True;Authentica
tion=Active Directory
Interactive;ConnectRetryCount=12;ConnectRetryInterval=10;"
providerName="System.Data.SqlClient" />

    <add name="OperationalDatabase" connectionString="Server=
<AzureSQLDBServer>;Database=Pulse;UID=AnyString;Encrypt=True;Authentication=Act
ive Directory Interactive;ConnectRetryCount=12;ConnectRetryInterval=10;"
providerName="System.Data.SqlClient" />

</connectionStrings>
```

5. Open the Execution Service config file:

C:\Program Files\RedPoint Global\RedPoint Interaction\ExecutionService\Resonance.ExecutionService.exe

Edit the connection strings as per the following (replacing the server name):

```
<connectionStrings>

    <add name="LoggingDatabase" connectionString="Server=
<AzureSQLDBServer>;Database=Pulse_Logging;UID=AnyString;Encrypt=True;Authentica
tion=Active Directory
Interactive;ConnectRetryCount=12;ConnectRetryInterval=10;"
providerName="System.Data.SqlClient" />

    <add name="OperationalDatabase" connectionString="Server=
<AzureSQLDBServer>;Database=Pulse;UID=AnyString;Encrypt=True;Authentication=Act
ive Directory Interactive;ConnectRetryCount=12;ConnectRetryInterval=10;"
providerName="System.Data.SqlClient" />

</connectionStrings>
```

6. Open the Interaction Web.config file:

C:\inetpub\wwwrpi\Interaction\Web.config

Edit the connection strings as per the following (replacing the server name):

```
<connectionStrings>
```

```
    <add name="LoggingDatabase" connectionString="Server=
<AzureSQLDBServer>;Database=Pulse_Logging;UID=AnyString;Encrypt=True;Authentica
tion=Active Directory
Interactive;ConnectRetryCount=12;ConnectRetryInterval=10;"
providerName="System.Data.SqlClient" />

    <add name="OperationalDatabase" connectionString="Server=
<AzureSQLDBServer>;Database=Pulse;UID=AnyString;Encrypt=True;Authentication=Act
ive Directory Interactive;ConnectRetryCount=12;ConnectRetryInterval=10;"
providerName="System.Data.SqlClient" />
</connectionStrings>
```

7. Restart IIS, the Node Manager Service, and the Execution Service to pick up the changes.

8. Repeat steps 4-7 for all VMs on the cluster.

9. Log into Server Workbench and navigate to the Clients tab. Update the following connection strings (updating the server and databases):

   - Operational Database:

     ```
     Server= <AzureSQLDBServer>;Database=<interaction
     DB>;UID=AnyString;Encrypt=True;Authentication=Active Directory
     Interactive;ConnectRetryCount=12;ConnectRetryInterval=10
     ```

   - Auditing Database:

     ```
     Server= <AzureSQLDBServer>;Database=<interaction audit
     DB>;UID=AnyString;Encrypt=True;Authentication=Active Directory
     Interactive;ConnectRetryCount=12;ConnectRetryInterval=10
     ```

   - Data Warehouse Database:

     ```
     Server= <AzureSQLDBServer>;Database=<data warehouse
     DB>;UID=AnyString;Encrypt=True;Authentication=Active Directory
     Interactive;ConnectRetryCount=12;ConnectRetryInterval=10
     ```

   If applicable, update the Read-Only data warehouse database as well

10. Execute tests to validate that RPI can connect to the databases.

Note: if applicable, update the connection strings for any services that are connecting to the databases, including the SendGrid Callback service, and Realtime Agent service.

Appendix B – External Plugin Services Setup

The following sections describe how to set up each of RPI's external plugin services.

dotDigital External Dynamic Content Service

Installation

To install the dotdigital EDC Service:

1. Copy the entire DeploymentFiles\Plugin Services\DotMailerEdcService folder into the root of the IIS web site where the dotdigital EDC Service will be hosted.
2. In the App_Config folder, copy the ConnectionStrings.Example.config file and paste it into the same location. Rename the file to ConnectionStrings.config.
3. In the ConnectionStrings.config file, modify the two entries within the <connectionString> tags to point to the database server where the core RPI operations database is housed by replacing the 'localhost' string in the entries below with the name of the operations database server. If the RPI operational database is on the same machine as the RPI application server, then this will not need to be changed:

```
<add name="LoggingDatabase" connectionString="Data Source=.;Initial
Catalog=Pulse_Logging;Integrated Security=True;Connect
Timeout=90;ConnectRetryCount=12;ConnectRetryInterval=10"
providerName="System.Data.SqlClient" />

  <add name="OperationalDatabase" connectionString="Data Source=.;Initial
Catalog=Pulse;Integrated Security=True;Connect
Timeout=90;ConnectRetryCount=12;ConnectRetryInterval=10"
providerName="System.Data.SqlClient" />
```

To modify the dotdigital EDC service's default folder location for outbound personalized content:

1. Launch Redpoint Interaction - Server Workbench.
2. In Configuration tab, replace the DotMailerExternalDynamicContentSharedFolderPath default value with the folder path you want to use.
3. Save the changes.

In IIS Manager:

1. Expand the web site you created for the dotdigital EDC service.
2. Right-click on the 'DotMailerEdcService' folder and choose 'Convert to Application'
3. Set the Application Pool to the same application pool that RPI runs under (RPIAppPool) or, if hosted on a different server, make sure the application pool runs under a security context that has access to the RPI Operations database referenced in the step above.
4. You can test that your service is running correctly by right-clicking on the

'DotMailerEdcService' application in IIS and choosing 'Browse'. If the service is running, you will see the content 'Redpoint Interaction DotMailer'

Upgrade

To upgrade the dotdigital EDC service at an existing RPI installation, please follow these steps:

Upgrading from v5.x and below

1. Back up the dotdigital EDC service web.config file.
2. Stop the dotdigital EDC service in IIS Manager.
3. Copy the contents of the DeploymentFiles\Plugin Services\DotMailerEdcService into the dotdigital EDC service website's root folder.
4. In the App_Config folder, copy the ConnectionStrings.Example.config and paste it in the same location. Rename the file to ConnectionStrings.config.
5. In the ConnectionStrings.config file, modify the two entries within the <connectionString> tags to point to the database server where the core RPI operations database is housed by replacing the 'localhost' string in the entries below with the name of the operations database server. If the RPI operational database is on the same machine as the RPI application server, then this would not need to be changed:

```
<add name="LoggingDatabase" connectionString="Data Source=.;Initial
Catalog=Pulse_Logging;Integrated Security=True;Connect
Timeout=90;ConnectRetryCount=12;ConnectRetryInterval=10"
providerName="System.Data.SqlClient" />

<add name="OperationalDatabase" connectionString="Data Source=.;Initial
Catalog=Pulse;Integrated Security=True;Connect
Timeout=90;ConnectRetryCount=12;ConnectRetryInterval=10"
providerName="System.Data.SqlClient" />
```

6. If the dotdigital EDC service has configured custom folder location previously in 'ExternalDynamicContentSharedFolder' setting found in web.config file, copy the value of the setting and paste it on Server Workbench under Configurations > 'DotMailerExternalDynamicContentSharedFolderPath' setting and save the changes.
7. Start 'DotMailerEdcService' application from IIS manager.

Upgrading from v6.x and up

1. Back up the dotdigital EDC service App_Config\ ConnectionStrings.config file.
2. Stop the dotdigital EDC service from the IIS Manager.
3. Copy the contents of DeploymentFiles\Plugin Services\DotMailerEdcService into the dotdigital EDC service website's root folder.
4. Start 'DotMailerEdcService' application from IIS manager.

In Server Workbench, any modification made from the

DotMailerExternalDynamicContentSharedFolderPath setting requires a restart of dotdigital EDC service from the IIS Manager for the changes to take effect.

Outbound Delivery Callback Service

Installation

To install the Outbound Delivery Callback Service:

1. Copy the entire DeploymentFiles\Plugin Services\OutboundDeliveryCallbackService folder into the root of the IIS web site where the Outbound Delivery callback service will be hosted.
2. In the App_Config folder, copy the ConnectionStrings.Example.config file and paste it in the same location. Rename the file to ConnectionStrings.config.
3. In the ConnectionStrings.config file, modify the two entries within the <connectionString> tags to point to the database server where the core RPI operational database is housed by replacing the 'localhost' string in the entries below with the name of the operational database server. If the RPI operational database is on the same machine as the RPI application server, then this does not need to be changed:

```
<add name="LoggingDatabase" connectionString="Data Source=.;Initial
Catalog=Pulse_Logging;Integrated Security=True;Connect
Timeout=90;ConnectRetryCount=12;ConnectRetryInterval=10"
providerName="System.Data.SqlClient" />

<add name="OperationalDatabase" connectionString="Data Source=.;Initial
Catalog=Pulse;Integrated Security=True;Connect
Timeout=90;ConnectRetryCount=12;ConnectRetryInterval=10"
providerName="System.Data.SqlClient" />
```

To modify Outbound Delivery callback service's default folder location for inbound events:

1. Launch Server Workbench.
2. In the Configuration tab, replace the OutboundDeliveryEventFolderPath default value with the correct folder path.
3. Save the changes.

In IIS Manager:

1. Expand the web site you created for the Outbound Delivery callback service.
2. Right-click the 'OutboundDeliveryCallbackService' folder and choose 'Convert to Application'
3. Set the Application Pool to the same application pool that RPI runs under (RPIAppPool) or, if hosted on a different server, make sure the application pool runs under a security context that has access to the RPI Operational database referenced in the step above.
   You can test that your service is running correctly by right-clicking on the 'OutboundDeliveryCallbackService' application in IIS and choosing 'Browse'. If the service is running, you will see 'Nuget Dependencies' content.

Upgrade

To upgrade the Outbound Delivery Callback Service at an existing RPI installation, please follow these steps:

Upgrading from v5.x and below:

1. Back up the Outbound Delivery callback service web.config file.
2. Stop the Outbound Delivery callback service in IIS Manager.
3. Copy the contents of DeploymentFiles\Plugin Services\ OutboundDeliveryCallbackService into the Outbound Delivery callback service website's root folder.
4. In the App_Config folder, copy the ConnectionStrings.Example.config and paste it in the same location. Rename the file to ConnectionStrings.config.
5. In the ConnectionStrings.config file, modify the two entries within the <connectionString> tags to point to the database server where the core RPI operational database is housed by replacing the 'localhost' string in the entries below with the name of the operational database server. If the RPI operational database is on the same machine as the RPI application server, then this does not need to be changed:

```
<add name="LoggingDatabase" connectionString="Data Source=.;Initial
Catalog=Pulse_Logging;Integrated Security=True;Connect
Timeout=90;ConnectRetryCount=12;ConnectRetryInterval=10"
providerName="System.Data.SqlClient" />

<add name="OperationalDatabase" connectionString="Data Source=.;Initial
Catalog=Pulse;Integrated Security=True;Connect
Timeout=90;ConnectRetryCount=12;ConnectRetryInterval=10"
providerName="System.Data.SqlClient" />
```

6. If the Outbound Delivery callback service has a custom folder location configured at the web.config file's 'eventFolderLocation' setting, copy the value and paste it into Server Workbench's 'OutboundDeliveryEventFolderPath' configuration setting, and save the changes.
7. Start the 'OutboundDeliveryCallbackService' application in IIS manager.

Upgrading from v6.x and up:

1. Back up the Outbound Delivery callback service App_Config\ ConnectionStrings.config file.
2. Stop the Outbound Delivery callback service from the IIS Manager.
3. Copy the contents of DeploymentFiles\Plugin Services \OutboundDeliveryCallbackService into the Outbound Delivery callback service website's root folder.
4. Start 'OutboundDeliveryCallbackService' application from IIS manager.

Note that any changes made to the Server Workbench OutboundDeliveryEventFolderPath setting will necessitate a restart of the Outbound Delivery service in IIS Manager for the changes to take effect.

SendGrid Web API Callback Service

Installation

To install the SendGrid Web API Callback Service:

1. Copy the entire DeploymentFiles\Plugin Services\ SendGridCallbackServiceWebAPI folder into the root of the IIS web site where the SendGrid Web API callback service will be hosted.
2. In the App_Config folder, copy the ConnectionStrings.Example.config file and paste it in the same location. Rename the file to ConnectionStrings.config.
3. In the ConnectionStrings.config file, modify the two entries within the <connectionString> tags to point to the database server where the core RPI operational database is housed by replacing the 'localhost' string in the entries below with the name of the operational database server. If the RPI operational database is on the same machine as the RPI application server, then this does not need to be changed:

```
<add name="LoggingDatabase" connectionString="Data Source=.;Initial
Catalog=Pulse_Logging;Integrated Security=True;Connect
Timeout=90;ConnectRetryCount=12;ConnectRetryInterval=10"
providerName="System.Data.SqlClient" />

<add name="OperationalDatabase" connectionString="Data Source=.;Initial
Catalog=Pulse;Integrated Security=True;Connect
Timeout=90;ConnectRetryCount=12;ConnectRetryInterval=10"
providerName="System.Data.SqlClient" />
```

To modify the SendGrid Web API callback service's default folder location for inbound email events:

1. Open Server Workbench.
2. In the Configuration tab, replace the SendGridEmailMetricsFolderPath default value with the folder path you want to use.
3. Save the changes.

To modify the SendGrid Web API callback service's external folder for inbound email events:

1. Open RedPoint Interaction.
2. Open Configuration > Channels > SendGrid channel.
3. Click 'Copy JSON configuration to clipboard'.
4. Open Server Workbench.
5. In the Configuration tab's SendGridWebAPIExternalContentProviderSettings, setting paste the copied JSON configuration from the SendGrid channel.

Save the changes.

In IIS Manager:

1. Expand the web site you created for the SendGrid callback service
2. Right-click on the 'SendGridCallbackServiceWebAPI' folder and choose 'Convert to Application'
3. Set the Application Pool to the same application pool that RPI runs under (RPIAppPool) or, if hosted on a different server, make sure the application pool runs under a security context that has access to the RPI Operational database referenced in the step above.
   You can test that your service is running correctly by right-clicking on the 'SendGridCallbackServiceWebAPI' application in IIS and choosing 'Browse'. If the service is running, you will see 'Nuget Dependencies' content.

Upgrade

To upgrade the SendGrid callback service at an existing RPI installation, please follow these steps:

Upgrading from v5.x and below:

1. Back up the SendGrid Web API callback service web.config and appsettings.json file.
2. Stop the SendGrid Web API callback service in IIS Manager.
3. Copy the contents of DeploymentFiles\Plugin Services\
   SendGridCallbackServiceWebAPI into the SendGrid callback service website's root folder.
4. In the App_Config folder, copy the ConnectionStrings.Example.config and paste it in the same location. Rename the file to ConnectionStrings.config.
5. In the ConnectionStrings.config file, modify the two entries within the <connectionString> tags to point to the database server where the core RPI operational database is housed by replacing the 'localhost' string in the entries below with the name of the operational database server. If the RPI operational database is on the same machine as the RPI application server, this does not need to be changed:

   ```
   <add name="LoggingDatabase" connectionString="Data Source=.;Initial
   Catalog=Pulse_Logging;Integrated Security=True;Connect
   Timeout=90;ConnectRetryCount=12;ConnectRetryInterval=10"
   providerName="System.Data.SqlClient" />

   <add name="OperationalDatabase" connectionString="Data Source=.;Initial
   Catalog=Pulse;Integrated Security=True;Connect
   Timeout=90;ConnectRetryCount=12;ConnectRetryInterval=10"
   providerName="System.Data.SqlClient" />
   ```

6. If the SendGrid Web API callback service has a previously-configured custom folder location in the 'emailMetricsFolderLocation' setting found in the web.config file, copy the value of the setting and paste it into the SendGridEmailMetricsFolderPath configuration setting within Server Workbench.

7. Save the changes.
8. Start the 'SendGridCallbackServiceWebAPI' application in IIS manager.

Upgrading from v6.x and up:

1. Back up the SendGrid Web API callback service App_Config\ ConnectionStrings.config file and appsettings.json file.
2. Stop the SendGrid Web API callback service in IIS Manager.
3. Copy the contents of DeploymentFiles\Plugin Services\ SendGridCallbackServiceWebAPI into the SendGrid Web API callback service website's root folder.
4. Start the 'SendGridCallbackServiceWebAPI' application from IIS manager.

AWS SES Callback Service

Setup of the Amazon Simple Email Service (AWS SES) Callback Service is only required if using the AWS SES email service provider to execute RPI email offers.

The RPI AWS SES callback service is a public facing ASP.Net application used by both the AWS SES channel plugin and AWS SES Event Notifications. With this service, AWS SES is able to post any event notifications that occur as emails are processed. The AWS SES channel plugin can then retrieve these event notifications and display them as results in the RPI Results Window.

Deployment Files

The AWS SES Callback service is available in the RPI Deployment Files in the following folder:

DeploymentFiles\Plugins Services\AmazonSESCallbackService

Pre-Requisites

To set up the AWS SES callback service, the following pre-requisites must exist:

- A public-facing web server.
- An AWS SES callback service web site, hosted in IIS.

Installing the AWS SES Callback Service

To install the AWS SES Callback Service, copy the entire DeploymentFiles\Plugin Services\AmazonSESCallbackService folder into the root of the IIS web site where the AWS SES Callback service will be hosted.

In IIS Manager:

- Expand the AWS SES callback service website.
- Right-click on the 'AmazonSESCallbackService' folder and select 'Convert to Application'.
- Set the Application Pool to Pipeline Mode: Integrated.

You can test that the service is running correctly by browsing to **Error! Hyperlink reference not valid.**, where the following response should be returned:

```
<string xmlns="http://schemas.microsoft.com/2003/10/Serialization/">OK!</stri
ng>
```

Upgrading the AWS SES Callback Service

To upgrade the AWS SES callback service at an existing RPI installation, please copy the contents of DeploymentFiles\Plugin Services\AmazonSESCallbackService folder into the AWS SES callback service website's root folder.

AWS SES Channel Plugin Callback Service Configuration

To configure the RPI AWS SES channel plugin to use the AWS SES callback service, ensure that the AWS SES channel's configuration Callback service URL property is set correctly with the following format: http://<server name>

AWS SES Callback Service Write to External Location

To configure the AWS SES callback service to write to an external Location, please follow these steps:

- Stop the AWS SES callback service in IIS.
- Set the AWS SES channel configuration's Events external folder property to an external folder.
- In the same context, click Copy JSON configuration to clipboard.
- Open Server Workbench > Configuration.
- In the Connectors group, select the AmazonSesExternalContentProviderSettings setting and click the Value textbox, then paste the copied JSON configuration.
- Start the Amazon SES callback service in IIS.

You can test that your service has access to the External folder by browsing to **Error! Hyperlink reference not valid.**. Information about the selected folder should be shown in the EmailMetricsLocationStatus section.

AWS SES Callback Service Write to Queue Provider

To configure the AWS SES callback service to write to a queue provider, ensure that the value of the AWS SES channel configuration's Events external folder property, and the Server Workbench configuration setting AmazonSesExternalContentProviderSettings are empty, and that you have a working default queue provider (in RPI client's Queue Providers configuration interface). The service uses the CallbackServiceQueuePath client system configuration setting to write email events and the Callback service processor system task to read email events.

You can test that your service is set up to write to the queue provider by browsing to **Error! Hyperlink reference not valid.**. Queue provider information should be shown in the EmailMetricsLocationStatus section.

Appendix C – Database Plugins Required Permissions

The following tables document the database permissions required when working with Redpoint Interaction database plugins.

ActianVectorH Database Required Permissions

SELECT, INSERT, DELETE, UPDATE
EXECUTE
NEXT
CREATE_TABLE
SELECT_SYSCAT
ACCESS

ActianVectorH Database Default Permissions:

SELECT, INSERT, DELETE, UPDATE	Only the owner can perform select, insert, delete, or update operations on objects it owns
EXECUTE	Only the owner of a database procedure can execute the procedure
NEXT	Allows the grantee to execute the NEXT VALUE and CURRENT VALUE functions on the specified sequences. To grant the NEXT privilege on sequences, the grantor must either own the sequence or have NEXT privilege WITH GRANT OPTION for the sequence. Only the owner of a database sequence can execute the next value and current value operators on the sequence
CREATE_TABLE	Any user can create tables (create_table)
SELECT_SYSCAT	Allows a session to query system catalogs to determine schema information. When connected to the master database (*iidbdb*), this includes the master database catalogs such as *iiuser* and *iidatabase*. SELECT_SYSCAT can be granted to user, group, role or public, and can only be issued when connected to the *iidbdb* database. This privilege restricts user queries against the core DBMS

	catalogs containing schema information, such as *iirelation* and *iiattribute*. Standard system catalogs such as *iitables* can still be queried.
ACCESS	Allows the specified authorization IDs to connect to the specified database

AWS Redshift Database Default Permissions:

SELECT	Grants privilege to select data from a table or view using a SELECT statement. The SELECT privilege is also required to reference existing column values for UPDATE or DELETE operations
INSERT	Grants privilege to load data into a table using an INSERT statement or a COPY statement
UPDATE	Grants privilege to update a table column using an UPDATE statement. UPDATE operations also require the SELECT privilege, because they must reference table columns to determine which rows to update, or to compute new values for columns
DELETE	Grants privilege to delete a data row from a table. DELETE operations also require the SELECT privilege, because they must reference table columns to determine which rows to delete
CREATE	For schemas, CREATE allows users to create objects within a schema. To rename an object, the user must have the CREATE privilege and own the object to be renamed
EXECUTE ON ALL FUNCTIONS IN SCHEMA	Grants the specified privileges on all functions in the referenced schema

Greenplum Database Default Permissions:

PRIVILEGE	**DESCRIPTION**

SELECT	Allows SELECT from any column of the specified table, view, or sequence. Also allows the use of COPY TO. For sequences, this privilege also allows the use of the *currval* function
INSERT	Allows INSERT of a new row into the specified table. Also allows COPY FROM
UPDATE	Allows UPDATE of any column of the specified table. SELECT ... FOR UPDATE and SELECT ... FOR SHARE also require this privilege (as well as the SELECT privilege). For sequences, this privilege allows the use of the *nextval* and *setval* functions
DELETE	Allows DELETE of a row from the specified table.
CREATE	For databases, allows new schemas to be created within the database. For schemas, allows new objects to be created within the schema. To rename an existing object, you must own the object and have this privilege for the containing schema. For tablespaces, allows tables and indexes to be created within the tablespace, and allows databases to be created that have the tablespace as their default tablespace. (Note that revoking this privilege will not alter the placement of existing objects.)
CONNECT	Allows the user to connect to the specified database. This privilege is checked at connection startup (in addition to checking any restrictions imposed by pg_hba.conf)
EXECUTE	Allows the use of the specified function and the use of any operators that are implemented on top of the function. This is the only type of privilege that is applicable to functions. (This syntax works for aggregate functions, as well.)
USAGE	For procedural languages, allows the use of the specified language for the creation of functions in that language. This is the only type of privilege that is applicable to procedural languages. For schemas, allows access to objects contained in the specified schema (assuming that the objects' own privilege requirements are also met). Essentially this allows the grantee to look up objects within the schema.
	For sequences, this privilege allows the use of the *currval* and *nextval* functions

MariaDB Database Default Permissions:

PRIVILEGE	DESCRIPTION
ALTER	Change the structure of an existing table using the ALTER TABLE statement
CREATE	Create a table using the CREATE TABLE statement. You can grant the CREATE privilege on tables that do not yet exist
CREATE VIEW	Create a view using the CREATE_VIEW statement
DELETE	Remove rows from a table using the DELETE statement
DROP	Drop a table using the DROP TABLE statement or a view using the DROP VIEW statement
INDEX	Create an index on a table using the CREATE INDEX statement. Without the INDEX privilege, you can still create indexes when creating a table using the CREATE TABLE statement if you have the CREATE privilege, and you can create indexes using the ALTER TABLE statement if you have the ALTER privilege
INSERT	Add rows to a table using the INSERT statement. The INSERT privilege can also be set on individual columns; see Column Privileges below for details
SELECT	Read data from a table using the SELECT statement. The SELECT privilege can also be set on individual columns; see Column Privileges below for details
SHOW VIEW	Show the CREATE VIEW statement to create a view using the SHOW CREATE VIEW statement

UPDATE	Update existing rows in a table using the UPDATE statement. UPDATE statements usually include a WHERE clause to update only certain rows. You must have SELECT privileges on the table or the appropriate columns for the WHERE clause. The UPDATE privilege can also be set on individual columns; see Column Privileges below for details
ALTER ROUTINE	Change the characteristics of a stored procedure using the ALTER PROCEDURE statement
EXECUTE	For stored procedure, execute a stored procedure using the CALL statement. The privilege to call a procedure may allow you to perform actions you wouldn't otherwise be able to do, such as insert rows into a table. For function, use a stored function. You need SELECT privileges for any tables or columns accessed by the function
CREATE ROUTINE	Create Stored Programs using the CREATE PROCEDURE and CREATE FUNCTION statements

MySQL Database Default Permissions:

PRIVILEGE	**DESCRIPTION**
CREATE	Enables creation of new databases and tables or indexes
DROP	Enables you to drop (remove) existing databases, tables, and views. The DROP privilege is required in order to use the statement ALTER TABLE ... DROP PARTITION on a partitioned table. The DROP privilege is also required for TRUNCATE TABLE. If you grant the DROP privilege for the mysql database to a user, that user can drop the database in which the MySQL access privileges are stored
ALTER	Enables use of ALTER TABLE to change the structure of tables. ALTER TABLE also requires the CREATE and INSERT privileges. Renaming a table requires ALTER and DROP on the old table, CREATE, and INSERT on the new table
DELETE	Enables rows to be deleted from tables in a database

INDEX	Enables you to create or drop (remove) indexes. INDEX applies to existing tables. If you have the CREATE privilege for a table, you can include index definitions in the CREATE TABLE statement
INSERT	Enables rows to be inserted into tables in a database. INSERT is also required for the ANALYZE TABLE, OPTIMIZE TABLE, and REPAIR TABLE table-maintenance statements
SELECT	Enables you to select rows from tables in a database. SELECT statements require the SELECT privilege only if they actually retrieve rows from a table. Some SELECT statements do not access tables and can be executed without permission for any database
UPDATE	Enables rows to be updated in tables in a database
ALTER ROUTINE	This privilege is needed to alter or drop stored routines (procedures and functions)
CREATE ROUTINE	This privilege is needed to create stored routines (procedures and functions)
EXECUTE	This privilege is required to execute stored routines (procedures and functions)
Note: *For full access to the sys schema, a user must have SELECT privilege on all sys tables and views*	

Netezza Database Default Permissions:

PRIVILEGE	**DESCRIPTION**
ALTER	Allows the user to modify object attributes. Applies to the following objects: SCHEMA, SEQUENCE, TABLE, FUNCTION, AGGREGATE, VIEW
DELETE	Allows the user to delete table rows. Applies only to tables
DROP	Allows the user to drop the objects: SCHEMA, SEQUENCE, TABLE, FUNCTION, AGGREGATE, VIEW

GROOM	Allows the user to do general housekeeping and cleanup operations on tables by using the GROOM TABLE command. The GROOM TABLE command runs reclaim operations to remove deleted rows and also reorganizes tables that are based on the clustered base table's organizing keys
INSERT	Allows the user to insert rows into a table. Applies only to tables
LIST	Allows the user to display an object name, either in a list or in another manner. Applies to the following objects: DATABASE, SCHEMA, SEQUENCE, TABLE, FUNCTION, AGGREGATE, VIEW
SELECT	Allows the user to select (or query) rows within a table. Applies to tables and views
UPDATE	Allows the user to modify table rows, such as changing field values or changing the next value of a sequence. Applies to tables only
EXECUTE	Allows the user to execute UDFs and UDAs in SQL queries

Oracle Database Default Permissions:

PRIVILEGE	**DESCRIPTION**
CREATE SESSION	Enables a user to create a connection to the database
CREATE ANY PROCEDURE	Enables a user to create a PL/SQL procedure, function or package owned by any user in the database
CREATE ANY VIEW	Enables a user to create a view owned by any user in the database
CREATE ANY TABLE	Enables a user to create a table owned by any user in the database
CREATE ANY INDEX	Enables a user to create an index on any table or materialized view in the database
ALTER SESSION	Set or modify any of the conditions or parameters that affect your connection to the database

ALTER ANY PROCEDURE	Enables a user to alter any PL/SQL procedure, function or package in the database
ALTER ANY TABLE	Enables a user to alter any table in the database
DELETE ANY TABLE	Enables a user to delete from any table in the database
SELECT ANY TABLE	Enables a user to select from any table, view, materialized view, or synonym in the database
SELECT ANY SEQUENCE	Enables a user to select from any sequence or synonym on a sequence in the database
UPDATE ANY TABLE	Enables a user to update any table or synonym in the database
INSERT ANY TABLE	Enables a user to insert into any table in the database. It also enables the user to insert into any table using the synonym, public or private, to that table
EXECUTE ANY PROCEDURE	Enables a user to execute any PL/SQL procedure, function or package in the database
DROP ANY PROCEDURE	Enables a user to drop any PL/SQL procedure, function or package in the database
DROP ANY SEQUENCE	Enables a user to drop any sequence in the database
DROP ANY TABLE	Enables a user to drop any table in the database
DROP ANY VIEW	Enables a user to drop any view in the database
DROP ANY INDEX	Enables a user to drop any index in the database

PostgreSQL Database Default Permissions:

PRIVILEGE	**DESCRIPTION**
SELECT	Allows SELECT from any column, or the specific columns listed, of the specified table, view, or sequence. Also allows the use of COPY TO. This privilege is also needed to reference existing column values in UPDATE or DELETE. For sequences, this privilege also allows the use of the currval function. For large objects, this privilege allows the object to be read

INSERT	Allows INSERT of a new row into the specified table. If specific columns are listed, only those columns may be assigned to in the INSERT command (other columns will therefore receive default values). Also allows COPY FROM
UPDATE	Allows UPDATE of any column, or the specific columns listed, of the specified table. (In practice, any nontrivial UPDATE command will require SELECT privilege as well, since it must reference table columns to determine which rows to update, and/or to compute new values for columns.) SELECT ... FOR UPDATE and SELECT ... FOR SHARE also require this privilege on at least one column, in addition to the SELECT privilege. For sequences, this privilege allows the use of the nextval and setval functions. For large objects, this privilege allows writing or truncating the object
DELETE	Allows DELETE of a row from the specified table. (In practice, any nontrivial DELETE command will require SELECT privilege as well, since it must reference table columns to determine which rows to delete.)
CREATE	For databases, allows new schemas to be created within the database. For schemas, allows new objects to be created within the schema. To rename an existing object, you must own the object and have this privilege for the containing schema. For tablespaces, allows tables, indexes, and temporary files to be created within the tablespace, and allows databases to be created
	that have the tablespace as their default tablespace. (Note that revoking this privilege will not alter the placement of existing objects.)
CONNECT	Allows the user to connect to the specified database. This privilege is checked at connection startup (in addition to checking any restrictions imposed by pg_hba.conf)
EXECUTE	Allows the use of the specified function and the use of any operators that are implemented on top of the function. This is the only type of privilege that is applicable to functions. (This syntax works for aggregate functions, as well.)

USAGE	For procedural languages, allows the use of the specified language for the creation of functions in that language. This is the only type of privilege that is applicable to procedural languages. For schemas, allows access to objects contained in the specified schema (assuming that the objects' own privilege requirements are also met). Essentially this allows the grantee to "look up" objects within the schema. Without this permission, it is still possible to see the object names, e.g. by querying the system tables. Also, after revoking this permission, existing backends might have statements that have previously performed this lookup, so this is not a completely secure way to prevent object access. For sequences, this privilege allows the use of the currval and nextval functions

SpliceMachine Database Default Permissions:

PRIVILEGE	**DESCRIPTION**
SELECT	To grant permission to perform SELECT statements or SelectExpressions on a table or view. If a column list is specified with the SELECT privilege, the permission is valid on only those columns. If no column list is specified, then the privilege is valid on all of the columns in the table. For queries that do not select a specific column from the tables involved in a SELECT statement or SelectExpression (for example, queries that use COUNT(*)), the user must have at least one column-level SELECT privilege or table-level SELECT privilege
INSERT	To grant permission to insert rows into the specified table
DELETE	To grant permission to delete rows from the specified table
UPDATE	To grant permission to use the UPDATE statement on the specified table. If a column list is specified, the permission applies only to the specified columns. To update a row using a statement that includes a WHERE clause, you must have the SELECT privilege on the columns in the row that you want to update

USAGE	In order to use a sequence generator, you must have the USAGE privilege on it
EXECUTE	Enables granting access to function or procedure

SybaseIQ Database Default Permissions:

PRIVILEGE	DESCRIPTION
CREATE ANY INDEX	Allows to create an index on a specified table, or pair of tables
CREATE ANY OBJECT	Allows to create any database objects
CREATE ANY PROCEDURE	Allows to create a procedure
CREATE ANY SEQUENCE	Allows to create a sequence generator
CREATE ANY TABLE	Allows to create a table
CREATE ANY VIEW	Allows to create a view
ALTER ANY INDEX	Allows to modify indexes
ALTER ANY OBJECT	Allows to modify any objects
ALTER ANY SEQUENCE	Allows alter to a sequence object
ALTER ANY PROCEDURE	Allows alter to a procedure object
ALTER ANY TABLE	Allows to modify tables
ALTER ANY VIEW	Allows to modify views
DROP ANY INDEX	Allows to delete any explicitly created index
DROP ANY OBJECT	Allows to delete any database objects
DROP ANY PROCEDURE	Allows to delete procedure
DROP ANY SEQUENCE	Allows to delete sequence generator
DROP ANY TABLE	Allows to delete a table
DROP ANY VIEW	Allows to delete a view

SELECT ANY TABLE	Allows to retrieve information from the database
DELETE ANY TABLE	Allows to Delete all the rows from the named table that satisfy the search condition
INSERT ANY TABLE	Allows to insert a single row or a selection of rows, from elsewhere in the current database, into the table
UPDATE ANY TABLE	Allows to modify existing rows of a single table, or a view that contains only one table
USE ANY SEQUENCE	Allows to use sequence generator
EXECUTE ANY PROCEDURE	Allows to execute a procedure

Vertica Database Default Permissions:

PRIVILEGE	**DESCRIPTION**
CONNECT	Allows the user to connect to a database
EXECUTE	The type of privilege to grant the procedure
CREATE	Allows the user read access to the schema and the right to create tables and views within the schema
USAGE	Allows the user access to the objects contained within the schema. This allows the user to look up objects within the schema. Note that the user must also be granted access to the individual objects
SELECT	For Sequence, allows the right to use both the CURRVAL() and NEXTVAL() functions on the specified sequence. For Table, allows the user to SELECT from any column of the specified table. For View, grants a user or role SELECT operations to a view, and any resources referenced within it

INSERT	Allows the user to INSERT tuples into the specified table and to use the COPY command to load the table. **Note:** COPY FROM STDIN is allowed to any user granted the INSERT privilege, while COPY FROM <file> is an admin-only operation
UPDATE	Allows the user to UPDATE tuples in the specified table
DELETE	Allows DELETE of a row from the specified table

Teradata Database Default Permissions:

PRIVILEGE	**DESCRIPTION**
CREATE TABLE, CREATE VIEW, CREATE PROCEDURE	CREATE granted for the object type for the specified space
DROP TABLE, DROP VIEW, DROP PROCEDURE	DROP granted for the object type for the specified space
INDEX	Allows access to CREATE and DROP INDEX and COLLECT and DROP STATISTICS statements
EXECUTE PROCEDURE	Allows access the corresponding CALL statement. This privilege applies to all stored procedures in the specified space. For the grantee to use the privilege on the procedure, the owner of the stored procedure must have the appropriate privileges on the objects referenced by the stored procedure
INSERT, DELETE, SELECT, UPDATE	Privilege applies to all tables or views in the specified database. UPDATE applies to table or column of the table. For a grantee to use the granted privileges on a view, the owner of a view must have appropriate privileges on the underlying tables of the view

DB2 Database Default Permissions:

PRIVILEGE	DESCRIPTION
REORG	Grants the privilege to use the REORG utility to reorganize table spaces and indexes
EXECUTE	For procedure, this must only allow to grant access to the following procedures: - SYSPROC.ADMIN_CMD procedure - "{rpi_schema}"."RPIExecDDL"
ALTERIN	Grants the privilege to alter stored procedures and user-defined
	functions, or specify a comment for distinct types, cast functions that are generated for distinct types, sequences, stored procedures, triggers, and user-defined functions in the designated schemas
CREATEIN	Grants the privilege to create distinct types, sequences, stored procedures, triggers, and user-defined functions in the designated schemas
DROPIN	Grants the privilege to drop distinct types, sequences, stored procedures, triggers, and user-defined functions in the designated schemas
USAGE	Grants the USAGE privilege to use a sequence. This privilege is needed when the NEXT VALUE or PREVIOUS VALUE expression is invoked for a sequence name
ALTER	Grants the privilege to alter the specified table or create a trigger on the specified table. ALTER cannot be used if the statement identifies an auxiliary table or a view
DELETE	Grants the privilege to delete rows in the specified table or view. DELETE cannot be granted on an auxiliary table
INDEX	Grants the privilege to create an index on the specified table. INDEX cannot be granted on a view
INSERT	Grants the privilege to insert rows into the specified table or view. INSERT cannot be granted on an auxiliary table

SELECT	Grants the privilege to create a view or read data from the specified table or view. SELECT cannot be granted on an auxiliary table
UPDATE	Grants the privilege to update rows in the specified table or view. UPDATE cannot be granted on an auxiliary table

SQL Azure Database Default Permissions:

PRIVILEGE	DESCRIPTION
dbmanager	Server-level role for creating databases
loginmanager	Server-level security role for creating logins

SQL PDW Database Default Permissions:

PRIVILEGE	DESCRIPTION
CONNECT ON DATABASE, ALTER ON DATABASE	Permissions that apply to servers
CREATE PROCEDURE, CREATE TABLE, CREATE VIEW	Permissions that apply only to databases
ALTER,DELETE,EXECUTE,INSERT,SELECT,UPDATE	Permissions that apply to databases, schemas, and objects

Snowflake Database Default Permissions:

PRIVILEGE	DESCRIPTION
SYSADMIN	The default role to use

Redpoint Global Inc.

888 Worcester Street, Suite 200
Wellesley,
MA 02482
T: +1 781 725-0250 F:
+1 888 259 6052

www.Redpointglobal.com