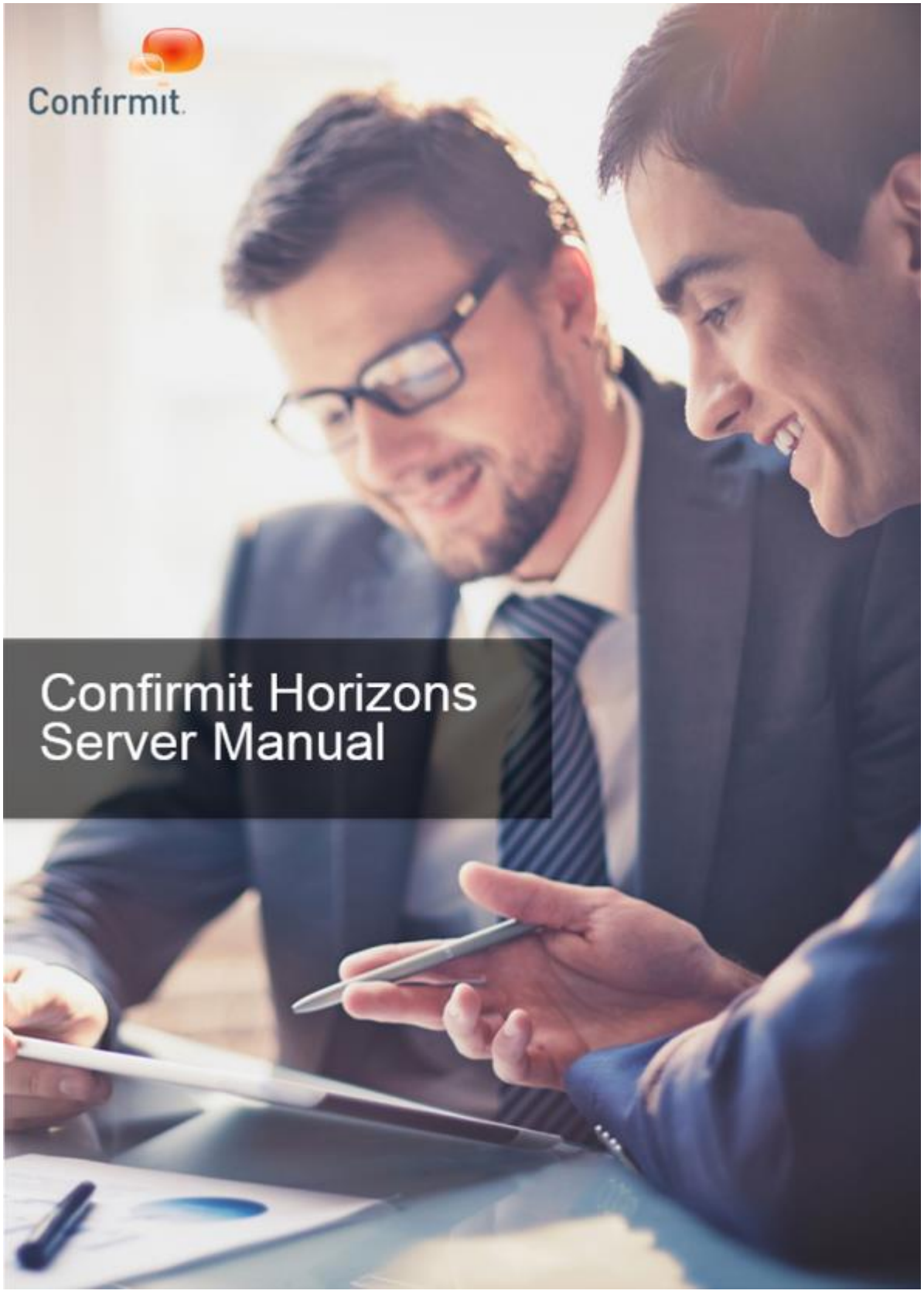


Confirmit Horizons v2019

This is revision 1 of the Confirmit Horizons v2019 Server Manual published in June 2019. The information herein describes Confirmit Horizons Server and its features as of Build nr. 2019.6.31. New features may be introduced into the product after this date. Go to www.confirmit.com or check “News” on the Customer Extranet for the latest updates.

Table of Contents

What's New in this Issue?

Note: Only the latest changes to this documentation are listed here. Changes made to earlier revisions are listed in the "Changes to the User Documentation" document which can be downloaded from the Confirmit Extranet at https://extranet.confirmit.com.

The following changes have been made in this revision of the Manual:

- An Important note is added to the Introduction section (see About This Manual on page 1 for more information).
- The Confirmit Studio section is added to the Confirmit Architecture and Features chapter (see Confirmit Studio on page 3 for more information).
- The text in the Confirmit Reportal section is edited (see Confirmit Reportal on page 4 for more information).
- The text in the Confirmit SmartHub section is edited (see Confirmit SmartHub on page 6 for more information).
- The text in the disclaimer is edited (see Virtual Infrastructure on page 18 for more information).
- A number of rows in the Preparing an Installation table are edited (see Preparing an Installation on page 20 for more information).
- Three obsolete sections are removed from the Preparing an Installation chapter.
- The text in the Microsoft SQL Server Integration Services (SSIS) section is edited (see Microsoft SQL Server Integration Services (SSIS) on page 24 for more information).
- Digital Feedback and Studio are added to the Confirmit Server Roles table (see Appendix C: Octopus Roles and Variables on page 118 for more information).

Note: The general layout and language in this document is continually being corrected, adjusted and improved to ensure the user has the best possible source of information. Only NEW information and details of functionality that has changed since the previous issue will be listed here - minor corrections to the text and document layout are not listed.

Important
We need your feedback so we can improve this document and provide you with the information you require. If you have any comments or constructive criticism concerning the content or layout of this documentation, please send an email to documentation@confirmit.com. Please include in your email the section number and/or heading text of the section to which your comment applies.

1. About This Manual

Important!

Version Numbering Change

As from the release of Confirmit Horizons of May 2019, the Horizons version naming format is changed to <year><month><build no.>, for example v2019.5.20, and subsequent major releases will adopt this version naming convention. We do not plan to adjust the frequency of these major releases; a new major release will be made available approximately every 9 months.

For on-premise releases we will continue to provide a Continuous Delivery version, however the previously named "Frozen" release will be renamed to "Long Term Support" (LTS) release. We believe this new terminology better reflects the reason this version is made available, and the principles of the support we provide.

This manual is intended as guide for preparing, installing and maintaining a Confirmit server environment. The content of the document is written from a technical viewpoint, and some experience with referenced Microsoft server products is recommended for achieving full value from the material contained herein.

Note: This manual does not include the installation of the Octopus system on an On-Premise site; this describes the planning and administration of Octopus deployment. Please contact the Confirmit support team for assistance with the installation.

Older versions of manuals can be downloaded from our Extranet at http://extranet.confirmit.com (requires login).

Please report any errors and misspellings in this document to documentation@confirmit.com.

2. Confirmit Architecture and Features

2.1. Basics

Confirmit is a three-tier web based application based on i386-compatible hardware architecture and runs on the Microsoft Windows Server platform.

The first tier is based on the Microsoft .NET Framework. This supports dynamic page content by providing an environment for interpreting server side scripts that process HTTP requests and generate HTML, which is returned to the client browser. The Microsoft .NET Framework is natively supported by Microsoft's web platform, Internet Information Services.

The .NET application accesses a middle tier of components that implement business logic and data access.

In the third tier, data storage is provided by Microsoft SQL Server for storing project and system data.

Figure 1 illustrates the Confirmit architecture and layout:

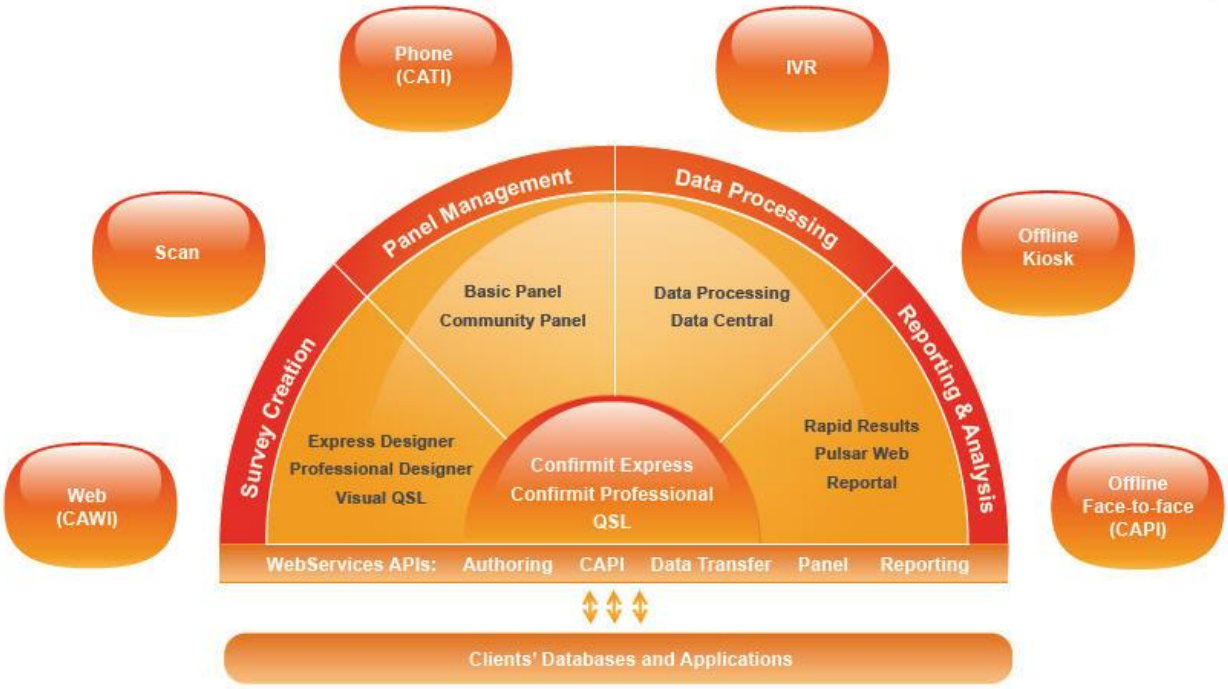


Figure 1 Confirmit architecture

2.2. Features and Modules

To ensure scalability and versatility for different usage scenarios, Confirmit has been designed from the ground up as a scalable, module-based application. Different application modules can run on dedicated hardware, and can scale both upwards and outwards with more powerful and/or additional servers, load balancing, load controlling and clustering up to theoretically any setup size.

2.2.1. Confirmit Authoring

The Confirmit Authoring system is a session-based web application running under Microsoft Internet Information Server and the Microsoft .NET Framework. Clients can use a range of modern Internet browsers for connecting to the application via the Internet or intranet. The term 'Confirmit', although not technically accurate, is widely used for describing the GUI of the Authoring module.

The Authoring module can be extended onto multiple web servers using a load balancing-capable network device that supports sticky user sessions.

The Authoring module uses Internet Information Services with WWW/.NET Framework features configured for content delivery and SMTP for dispatching error notifications from the system to system administrators. FTP services are an optional add-on for uploading files for importing into the system.

2.2.2. Confirmit Survey Designer

Survey Designer is the next generation of survey creation software. The application will eventually encompass all Confirmit Professional Authoring functionality, and will then supersede Professional Authoring. It is a session-less web application running under Microsoft Internet Information Server (IIS) on the node.js framework.

The role can be extended onto multiple web servers using a load balancer-capable network device.

Confirmit Professional Authoring is required to access Survey designer. While Survey Designer is intended to be a complete replacement to Confirmit Professional Authoring, not all functionality has yet been implemented.

2.2.3. Confirmit MetaData

With the removal of FastObjects (removed in version 18), a new Rest API was introduced to manage project definitions.

Data that was previously stored in FastObjects is now stored in the MetaDataDocumentStore, items that are currently being worked on will be loaded to the MetaDataNodeStore, and checked back to MetaDataDocumentStore when changes are committed. Both MetaDataDocumentStore and MetaDataNodeStore are stored in Microsoft SQL Server databases.

A new front end server role MetaDataRestApi was introduced in Confirmit v18 to manage communication between servers and the database layer. This provides a more scalable solution regarding performance and resilience. The MetaDataRestApi role can be load balanced across several servers.

2.2.4. Confirmit Studio

Confirmit's Studio™ is a platform for designing research solutions for different types of stakeholders, either in the form of full end-to-end work-flows from survey creation to reporting and taking action, or in the form of engaging dashboards that help bring the right insights to the right people. Studio helps users to design programs to collect, interact with, explore, visualize and take action based on data collected from research programs, in conjunction with other business systems' data to help businesses answer the burning question "why" and find out what undiscovered issues may be present.

Confirmit Studio™ is an easy-to-use tool that facilitates deep exploratory analysis using innovative analytical techniques.

It enables users to:

- Build work-flows that users can follow to create and deploy surveys.
- Build dashboards that enable users to analyze data combined from multiple sources across businesses, and conduct ad-hoc analysis of all data sets from SmartHub.
- Perform automated exploratory analysis and visualization on research programs, in conjunction with other data from internal and external sources.
- Use built-in analysis views that illustrate insights.

The benefits of Confirmit Studio™ include:

- Identifying the deeper insights that impact the business – those that are often hidden beneath the surface.
- Eliminating data silos that hide the most important insights.
- Saving time and effort by employing automated analysis.
- Improving business collaboration and communication and taking action based on insights.
- Displaying analytical findings in a format that is easy for non-analysts to understand.

2.2.5. Confirmit Deployment

The Deployment module consists of the Survey Engine, which loads interview pages requested by respondents and renders them for display in the respondents' browser.

Generated surveys are binary application packages served by Internet Information Services and the .NET Framework. Client scripts and HTML are session less; i.e. all necessary user states are preserved in the rendered HTML and relevant survey databases and are submitted between pages as hidden form values. This makes the Deployment module suitable for hardware or software load balancing. Load balancing also provides redundancy, which is strongly recommended for minimizing the risk of system unavailability in most setups.

The SMTP service is used for dispatching emails via web surveys. Scripts can be added to Confirmit surveys using Jscript.NET for sending emails directly from survey interview pages.

A standard Confirmit survey complies with the HTML v3.2 specification, which should make it viewable in most browsers in the field today. Project managers can deviate from the HTML v3.2 standard by designing Web Interview templates, survey layouts or using custom HTML code or scripts calling active content that might require extra functionality from the respondent's browser. The Confirmit user should keep this in mind when designing surveys.

2.2.6. Confirmit Reportal

Confirmit Reportal provides a dynamic interface for designing and publishing real-time reports based on data gathered in the Deployment module. Reports are created and generated in a GUI similar to the Authoring module. Reports can also be ordered via email in different formats for offline viewing.

In order to preserve SQL server performance, recently viewed report pages are cached by a dedicated system service (Confirmit Caching Service) for faster repeated views. The caching interval can be customized on a report level. For faster report generation, much of the strain has also been lifted from SQL with the introduction of the BitStream service. The service makes a periodical pull of data from the SQL database into BitStream files, which are used by the web servers for extracting data. This assures greatly improved performance in report delivery.

Implicit HUB as a BitStream replacement was introduced with V18. This utilizes Columnstore indexing provided by SQL Server to improve performance over BitStream and to introduce the ability to use report on Open Text data.

Reportal uses Internet Information Services and the .NET Framework for content delivery, and attaches to the Confirmit BitStream and Confirmit Caching Services for accelerated report content delivery. Reportal is normally run in parallel with surveys on the deployment servers, but can also be configured to run on dedicated servers. Reportal supports sticky-free load balancing, and user session persistence can be achieved via SQL or Redis. The session state control can be configured via Octopus variables.

2.2.7. Confirmit Task System

Project managers wishing to deploy a survey into production, order data exports or send out email invitations, will do so by initiating a Confirmit task on the system. Upon this action, the task is queued, and later picked up and executed at first available time spot by the Confirmit Task System, which runs as a system service on one or several servers. Different task types can be set to run on different servers for multiplying the number of possible concurrently running tasks in high-performance systems.

The task system requires some additional application components for enabling certain data export formats (Microsoft Office Open XML, Microsoft SQL Server Integration Services), and uses SMTP for dispatching emails. Exports can be encrypted via PGP before delivery, and (secure) FTP services can be optionally installed for downloading exported contents directly from the server instead of getting data via email.

2.2.8. Response Data Storage

Respondent answers are stored in Microsoft SQL Server databases. Each survey has its separate database and multiple servers/instances can be utilized to distribute survey database access. Some common system databases need to reside on a common database instance, while Confirmit supports a virtually unlimited number of dedicated survey database instances/servers. For high availability systems, SQL server can also be set up in a failover cluster.

Important
The SQL Server databases are critical components of any Confirmit installation and regular backup routines should always be implemented in a production environment. Backup recommendations are given in the maintenance chapter of this manual.

Important
The SQL software must run on equipment dedicated to Confirmit. The sharing of SQL Server with other applications is not supported.

2.2.9. Subsystems and Optional Components

2.2.9.1. Confirmit Web Services

The Confirmit Web Services provide application programming interfaces for allowing customers to attach custom applications to a set of APIs provided with the Confirmit Web Services. The web services are available as an add-on component for server customers.

2.2.9.2. Confirmit Translator

The Confirmit Translator module can be acquired as an add-on to Confirmit. When enabled, it allows for questionnaire translation to be out-sourced to external users working within a GUI limited to translation, with access only to the projects the project manager has enabled. Dedicated translators will typically not require standard Confirmit training.

2.2.9.3. Confirmit BitStream Service

The Confirmit BitStream service acts as an optimized buffer between Rapid Results / Reportal and the SQL Server databases. The service optimizes queries for large and complex reports, and effectively improves the I/O performance on database servers in environments with heavy report usage. The service generates BitStream files, a proprietary binary file format that supports incremental updates and provides faster access to report data.

Note: While Bitstream files are available as a data set for Reportal, Confirmit SmartHub has superseded this component.

2.2.9.4. Confirmit Agent Controller Service

The Confirmit Agent Controller service manages the agent processes related to Rabbit MQ queues. The service runs on servers assigned with the 'Task System' role. Certain events triggered by user interaction in Horizons are added to the RabbitMQ queue. Agents (in memory processes) monitor these queues for events and perform a task depending on the event request. This could be for example updates to SQL data or triggering a request from an internal web service.

2.2.9.5. Confirmit Multimode

Confirmit not only supports surveys over the web, but also provides options for offline personal interviewing (CAPI) and telephone-assisted interviewing (CATI). These features require a separate installation of the Confirmit Multimode module, which is covered in a separate manual.

2.2.9.6. Confirmit Flex Extensions

An extension is a computer program designed to be incorporated into and run by another "parent" application to provide additional functionality for the parent application - effectively a type of add-on.

Extensions can be used to modify the behavior of existing Confirmit features or add entirely new features. The Flex Extension feature is a separate installation not covered by this manual.

Read more about the Flex extensions in the Confirmit Flex User Guide, which you can download from the Confirmit Extranet.

2.2.9.7. Confirmit URL

This service provides the means to create short URLs for survey invitations. This is similar to other URL shorteners.

2.2.9.8. Confirmit Search Service

This is a service to facilitate the search for questions and nodes across all projects the users has access to. The service uses a Lucene search index to deliver fast results.

2.2.9.9. Confirmit SmartHub

A centralized area for data management and optimization, allowing multiple data sources of different types to be brought together. This feature is built on Columnstore indexing provided by SQL Server.

2.2.9.10. Confirmit Identity Service

The Identity Service provides authentication of users and applications for accessing Confirmit applications and services. It can also be utilized to provide Single-Sign-On integration with other systems, using OAuth2 and OpenIdConnect standards, which it fully supports.

This service is deployed as a dedicated Octopus role “IdentityService”, which can be assigned any number of servers and shared with other roles depending on the setup. Variables (Octopus and System Configuration) are defined in later sections of the manual.

Hosting this server using HTTPS is highly recommended, however this can be disabled if required.

2.2.9.11. Add-ons

Several other features can be unlocked by purchasing Confirmit add-ons. For a full list of available add-ons, consult the installation section of this document.

2.3. Application Dataflow

2.3.1. Confirmit Authoring Dataflow

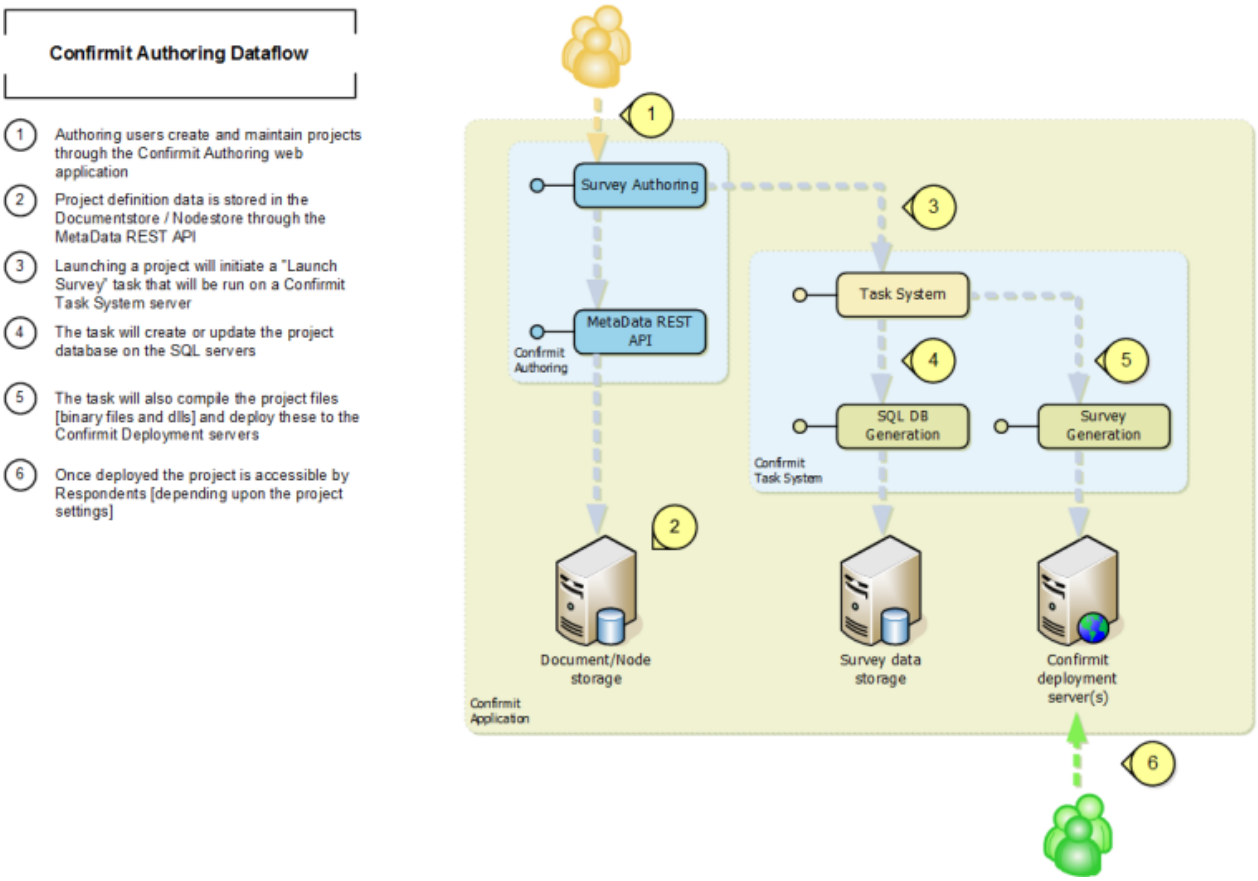


Figure 2 The Confirmit Authoring dataflow

2.3.2. Confirmit Survey Dataflow

Confirmit Survey Dataflow

1. Respondens are accessing the projects through the Confirmit Deployment servers [using the project specific URL]
2. The Confirmit Survey Engine displays the survey according to the deployed survey, reading background data from the project database and writes the answers to the project database

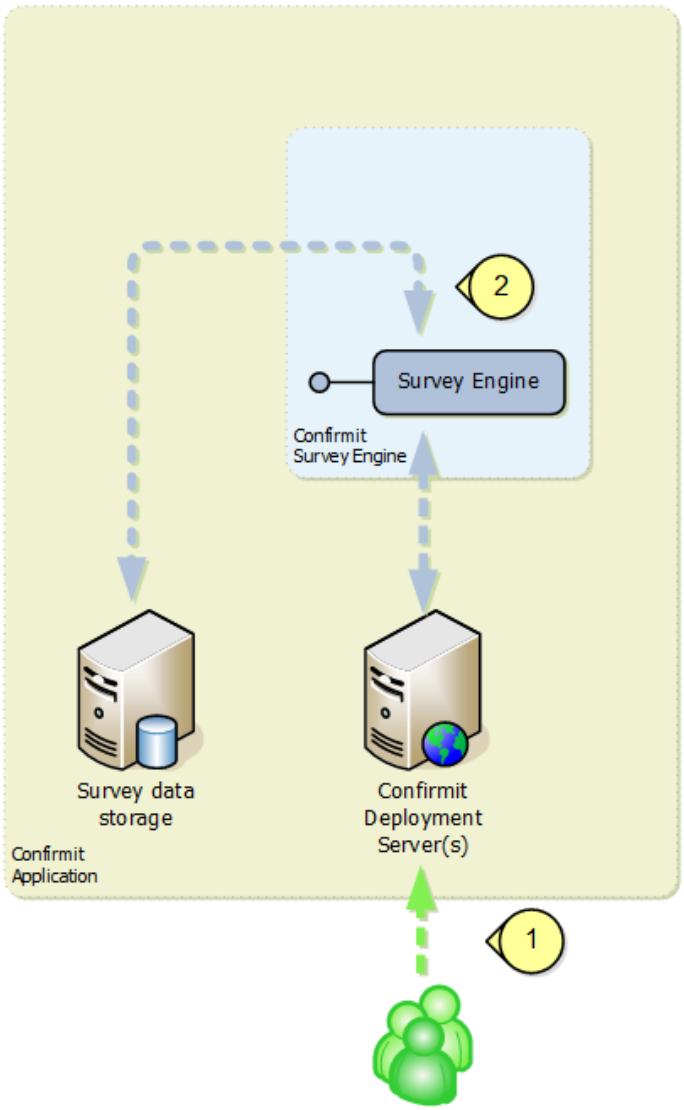


Figure 3 The Confirmit Survey dataflow

2.3.3. Confirmit Reportal Dataflow

Confirmit Reportal Dataflow (with BitStream)

1. Reportal users create and maintain reports while Reportal viewers can view predefined reports through the Confirmit Reportal web application.
2. Report definition are stored in a SQL database while project definition data is read from the Documentstore database. Reportal users may initiate a Report export that accesses the project data directly
3. The bitstream files will have to be created by the Confirmit Authoring users through a Bitstream Generation task
4. The task will read the data from the data source SQL database(s) and store to the Bitstream format
5. Reportal viewers can use the Bitstream data store for fast access to the project data

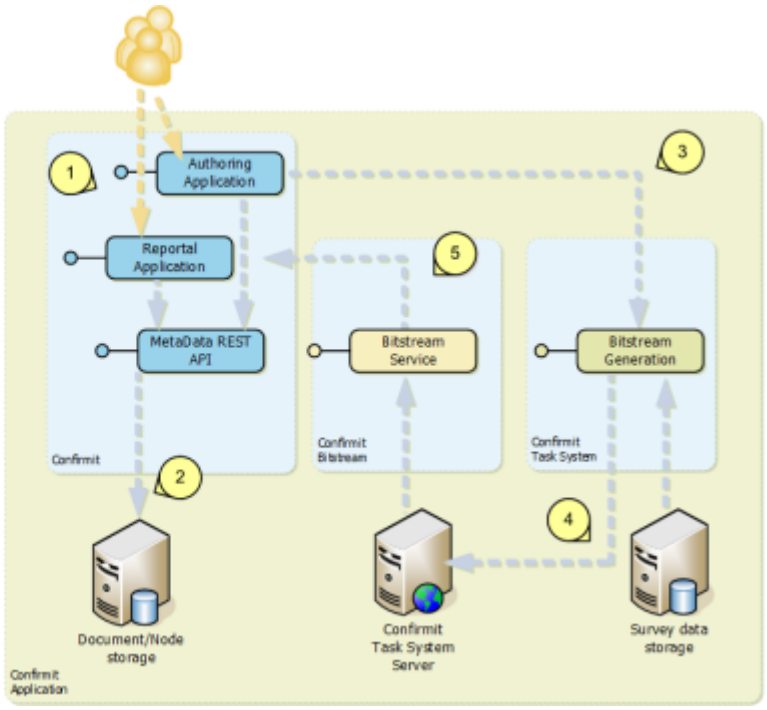


Figure 4 The Confirmit Reportal dataflow with BitStream files

Confirmit Reportal Dataflow (with SmartHub)

1. Reportal designers create and maintain reports while Reportal viewers can view predefined reports through the Confirmit Reportal web application. Reportal designers can use the SmartHub data store for fast access to report data
2. Report definitions are stored in a SQL database while project definition data is read from the DocumentStore database
3. The SmartHub database is created in the Confirmit Hub interface
4. A Hub loader task will read the data from the survey database(s) and store it in the Hub database (as a column store index)

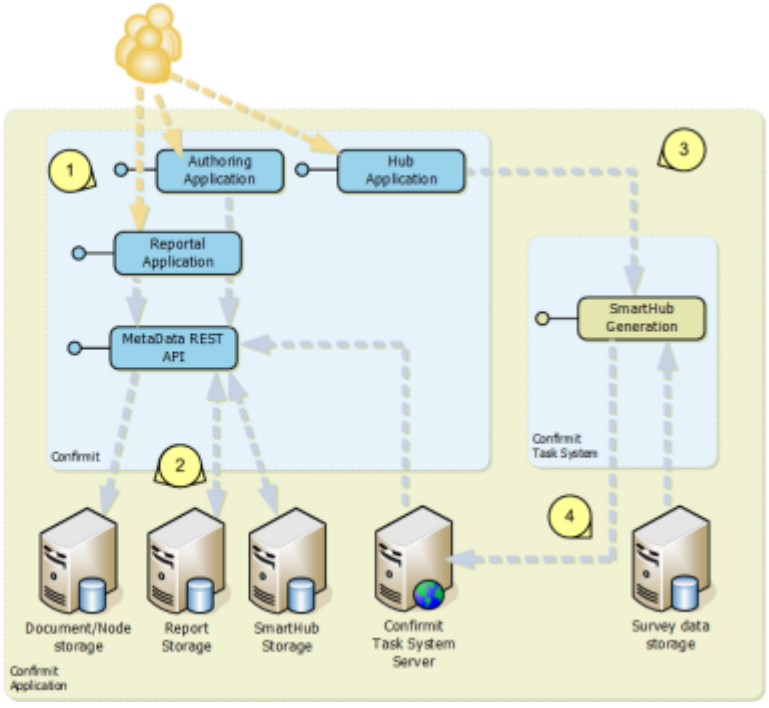


Figure 5 The Confirmit Reportal dataflow with SmartHub

2.3.4. Confirmit Data Export Dataflow

Confirmit Data Export Dataflow

1. Users order an data export through the Confirmit Application
2. A data export task will be created and picked up by the Confirmit Task System server(s)
3. The data export task will get the data from the project database stored in SQL and create the corresponding data export file(s)
4. Based on the data export settings the file can be email to the user
5. Or placed on the customer (S)FTP area for download

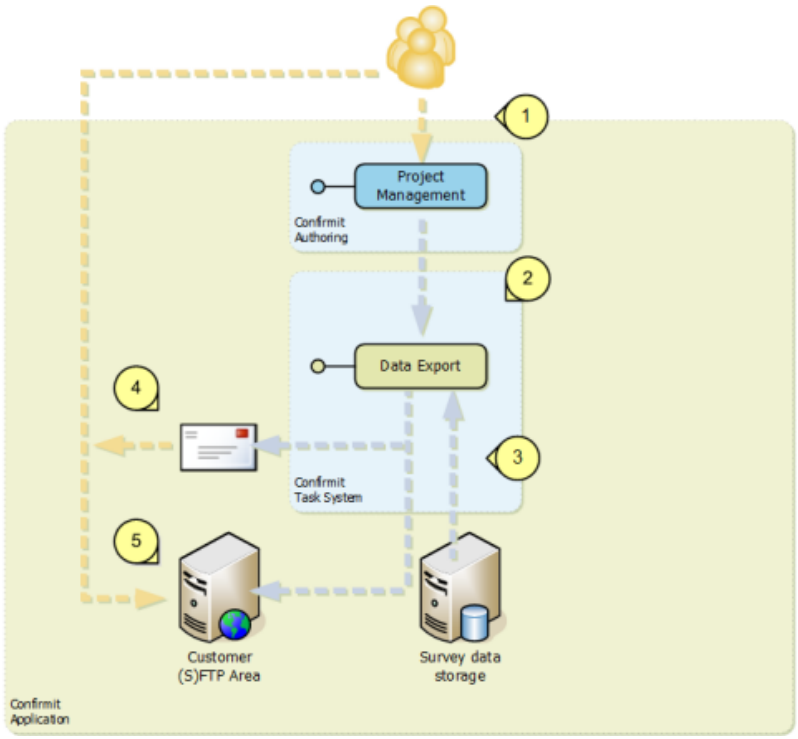


Figure 6 The Confirmit Data Export dataflow

3. Building a Confirmit Site

The topics under this heading describe the hardware required for a Confirmit site.

3.1. Before You Start

As mentioned in the previous chapter, Confirmit is built to be highly scalable both outwards and upwards. Performance capability or availability prediction can be increased by either deciding to use additional servers, or by upgrading existing servers with additional and/or faster CPU's and more memory. When one or more servers are working together towards the same database repository, the group is collectively named a Confirmit Site. The number of servers that can be used for a site is virtually unlimited, and servers can be added or removed from the configuration on the fly in most scenarios. All the servers in the same Confirmit Site can use the same Confirmit license file.

The Confirmit SaaS Operations team can assist you both with deciding which configuration is more suited to your needs and with planning site architecture based on some key figures from the expected usage levels. Contact support@confirmit.com or your account manager for assistance.

The most important factors to consider when planning a Confirmit site are:

- **Availability -** Everyone involved in systems maintenance knows that servers are prone to error both in hardware and software. If your Confirmit installation is business critical, consider at least N+1 redundancy for your critical components such as deployment servers and database servers (as well as underlying services such as power and network.)
- **Performance -** With more features running on a single server, the Confirmit system is more vulnerable for several components to stop working. For instance, Confirmit tasks might often use high amounts of CPU and memory resources for shorter periods of time, which might also affect database server or web server performance.
- **Scalability for future growth -** In virtually all scenarios, it will be possible to extend or expand existing Confirmit site architecture with additional or updated hardware. Analyzing expected usage patterns and planning one or two upgrade steps ahead could save both time and expenses later.

3.2. Network and Hardware Configuration

When planning and setting up network components and servers, there are some basic recommendations to consider:

3.2.1. Server Hardware

Server hardware from a reputable vendor is generally recommended due to the regular inclusion of support agreements that guarantee hardware replacement within a given time period in the case of a failure, as spare parts are usually widely available. Confirmit should be installed on 64-bit platforms ((go to Supported Operating Systems for more information), but individual components will run in 32-bit Windows-on-Windows mode. The exception is SQL Server services, which can run in native 64-bit.

The hardware recommendations provided in the following sub-sections should be considered as examples only, and the actual recommended hardware will depend strongly on the type and level of usage expected. The Confirmit SaaS Operations Team can also assist in setting up a tailored recommendation on request.

3.2.1.1. Octopus Deploy Server

The typical hardware outfit for an Octopus Deployment server would be:

- 2 CPU cores
- 4GB RAM (more recommended for high-volume usage)
- 10GB disk free minimum
- 100Mbit NIC

For the most up-to-date recommendations and prerequisites for Octopus, refer to the vendor's website which will have current recommended requirements - http://docs.octopusdeploy.com/display/OD/Installing+Octopus.

3.2.1.2. Web and Webservice Servers

The typical minimum hardware outfit for a web or combined server (Deployment, Webservice, API Rest Service, Authoring or Reportal) would be:

- 2 CPU cores
- 2GB RAM (more recommended for high-volume usage)
- RAID-1 disk system for system partition and Confirmit application
- Separate RAID-1 or RAID-5 disk system for Confirmit data partition
- 100Mbit NIC

Testing of Confirmit modules in Microsoft's ISV labs in Redmond have shown that it's usually more effective scaling out rather than scaling up when adding additional capacity. Generally one would scale up to a dual-CPU server then add additional dual-CPU servers. This makes it easier to scale the site to handle large loads. This will also provide extra redundancy for the service module being strengthened. Note that additional web servers will require load-balancing services (see Load Balancers on page 12 for more information).

A typical recommended hardware setup:

- 4 CPU cores
- 4GB RAM
- RAID-1 disk system for system partition and Confirmit application in separate volumes
- Separate RAID-1 disk system for Confirmit data partition
- Dual teamed Gbit NICs

3.2.1.3. Task System Servers (Backend)

Dedicated Task System servers would typically follow that of a web server (see Web and Webservice Servers on page 11 for more information). There should also be enough disk space on batch servers to store temporary task system files for certain task types such as archiving. data processing jobs and data exports.

Important
The archiving task specifically will store temporary files on the C drive, so there must have enough free space for this purpose otherwise the archiving functionality will fail.

You also need to ensure space for BitStream files unless dedicated BitStream servers are used. BitStream servers would typically be scaled the same as Task System servers.

Note: Space requirements will vary depending on the number and size of projects on the site. We recommend continual monitoring of disk space to allow for proactive disk expansions as required.

3.2.1.4. Database Servers (Backend)

Generally, for database servers we recommend ensuring an abundance of CPU power, memory and disk I/O. Multiple disk spindles or SAN storage can provide improved performance for database I/O operations. With .NET survey engine tuning, stress testing has shown that the database server is more likely to become a bottleneck in an under-dimensioned system due to the increasing levels of I/O on database servers during high loads, than web servers ever will, even when otherwise matched 1-to-1 on specifications.

Confirmit supports multiple SQL server instances as a Premium feature, and additional instances can be added to existing servers using dedicated disk spindles or SAN volumes to provide improved performance. If possible, system and survey databases should run on separate spindles/instances; and in a large multi-server setup, additional survey database instances or even separate servers (also with potentially multiple instances) can be added for optimal performance.

Minimum hardware setup recommendation:

- Dual-core CPU (additional/faster CPUs recommended if more than one SQL Server instance is running).
- 4GB RAM (recommended: increase by minimum 4GB per additional instance).

- RAID-1 disk system for system partition.
- Separate RAID-1 disk system for SQL instance running Confirmit databases.
- 100Mbit NIC.

Typical recommended hardware setup:

- 2x or 4x quad-core CPUs.
- 8GB RAM (recommended: increase by minimum 8GB per additional SQL Server instance - the more the better).
- RAID-1 disk system for system partition.
- RAID-10 disk system for SQL instance running Confirmit system databases (data files) (local or SAN storage).
- RAID-10, RAID-5 or RAID-1 disk system for SQL instance running Confirmit system databases (log files) (local or SAN storage).
- Additional data/log volumes for each SQL instance running Confirmit survey databases (local or SAN storage).
- Dual, teamed Gbit NICs.

Important
The SQL software must run on equipment dedicated to Confirmit. The sharing of SQL Server with other applications is not supported.

3.2.1.5. Multimode Servers

Confirmit Multimode requires (a) separate, dedicated machine(s) for installation of the multimode component. Hardware requirements are the same as for Web and Webservice Servers (see Web and Webservice Servers on page 11 for more information). The multimode component can be load-balanced if necessary for performance distribution and redundancy.

If using a dialer service, a separate dedicated WebService server of the same specification as a Web Server is required.

3.2.2. Network Equipment

3.2.2.1. Switches

High-speed, full duplex network switches are recommended. Dual switch layout can be used to provide network redundancy, although this could require additional configuration in the network department in terms of setting master/slave devices, routing etc.

3.2.2.2. Load Balancers

If using a load balancer or pair of load balancers, the device(s) should be able to balance traffic based on different patterns, and preferably also supporting multiple virtual IP addresses with individual rule sets (i.e. Authoring would require sticky session management via the LB, while Reportal, being written entirely in .NET, supports sessions across servers using SQL Server, ASP state or Redis session control. Confirmit surveys are completely session-less, and require no session tracking on the LB device.)

Load balancers should preferably also be able to use health checks using URL patterns or HTTP HEAD requests to check for actual availability in the real server pool, in order to allow performing maintenance on servers in the load balanced pool without interrupting the service as a whole.

Network Load Balancing (NLB) services in the Windows Server platform may also be used, but its functionality for providing sticky sessions for Authoring servers is not tested by Confirmit and therefore cannot be recommended in a production environment.

Note: Confirmit are not Load Balancer experts, and configurations and/or technology used varies from site to site (see Using Confirmit with a Load Balancer with SSL Acceleration on page 78 for more information). Please contact support for further assistance.

3.2.2.3. Firewalls/IDS

There are no specific requirements in terms of firewall hardware. However it is strongly recommended that firewalls from reputable vendors are deployed to provide some security on inbound connections to web servers to minimize the surface area for potential attacks.

3.2.2.3.1. Network Traffic Requirements to/from the Confirmit Site

From	To	Service	Ports	Protocol
Users	Deployment servers	HTTP / HTTPS	80,443 (customizable)	TCP
Users	Authoring servers	HTTP / HTTPS	80,443 (customizable)	TCP
Users	Webservice servers	HTTP / HTTPS	80,443 (customizable)	TCP
Users	Rest API servers	HTTP / HTTPS	80,443 (customizable)	TCP
Users	Reportal servers	HTTP / HTTPS	80,443 (customizable)	TCP
All Confirmit servers	Remote mail servers	SMTP	25	TCP
Users	*Server holding shared data / FTP data*	*FTP / SFTP / SSH / FRTPS*	*20, 21, 22, 990 or custom*	*TCP*
Octopus Server	Artifactory Feed	HTTPS	443	TCP

The rule marked in italics is only required when the FTP add-on is enabled. Most FTP, SFTP / FTPS / SSH servers may be configured to run on custom ports if you wish to use ports other than the default.

Note: With respect to email delivery - some remote mail server systems may run a connection test towards sending servers to verify the validity of the server, as a means to determine spam score. Allowing incoming SMTP connections to Confirmit servers that are sending email may increase email deliverability.

Confirmit web servers can also be set up to listen on non-standard ports if required.

3.2.2.3.2. Network Traffic Requirements Between Servers Internally Within the Site

It is recommended to run servers in the same network segment. However, if network access control between servers is required, the services required by Confirmit are listed below:

From	To	Service	Ports	Protocol
Octopus server	All Confirmit Web servers	Octopus Tentacle	10933	TCP
Octopus server	SQL Database server	SQL Connection	1433	TCP
All Confirmit servers	SQL Database server(s)	SQL connection	1433 (customizable)	TCP
All Confirmit servers	Metadata Rest API server	Web	80	TCP
Authoring servers, Task System servers	All Confirmit servers (also database servers if Archiving is enabled)	File system access over SMB/CIFS	445	TCP
Authoring servers	Searching service	Searching service	9731	TCP
Authoring, Task System and	Server(s) running RabbitMQ	Rabbit MQ	5672	TCP

Metadata REST API servers				
Reportal servers	Server running 'Confirmit Caching Service'	Confirmit Caching Service	8282	TCP
All Confirmit servers	Servers running 'Confirmit BitStream Service'	Confirmit BitStream Service	8285 (customizable)	TCP
(Optional) All Confirmit servers	Logstash	Logging	9998 (configurable)	TCP[1]
Task System servers	Confirmit Authoring URL	Web	80 / 443	TCP
Deployment servers	Confirmit Services accessed in Custom Code Libraries	Web	80 / 443	TCP
Reportal servers	Confirmit Authoring URL	Web	80 / 443	TCP
Confirmit Translator servers	Confirmit Authoring URL	Web	80 / 443	TCP
Confirmit CATI servers[2]	Confirmit Webservices	Web	80 / 443	TCP
Confirmit CATI servers[2]	Confirmit Authoring URL	Web	80 / 443	TCP
Confirmit CATI servers[2]	Confirmit Deployment URL	Web	80 / 443	TCP
1 - This rule is optional				
2 - If applicable				

3.3. Site Layout

In this chapter we have provided some examples for Confirmit server and infrastructure scenarios for different requirements relevant to the points above. The examples show how Confirmit can be an easily maintainable application running off a server, and how it via different scenarios can evolve into a fully-fledged enterprise solution. The scenarios presented are only meant as examples, and are not absolute. Furthermore, they are deliberately conservative, and real capacity should exceed the outlined calculations. Other combinations of modules than those listed here are of course possible. Confirmit Technical Support can help design the site based on your requirements; please contact Confirmit Support or your Account Manager for more information.

3.3.1. Basis for Setting Up the Recommendations

When setting up site recommendations for customers, and in the examples provided below, some key numbers are collected in advance. These include:

- The number of concurrent Authoring users.
- The number of concurrent Reportal viewers.
- The number of launched surveys per year.
- Estimated maximum surveys (peak) launched over a period of one month.
- The number of expected completed responses collected per year.
- Estimated average survey pages per complete response ratio / average survey complexity. In the examples below, we have used an average ratio of 20 survey pages per completed response.
- Estimated response completion ratio (total page hits versus completes.) In the examples below, we have used an average completion ratio of 20% across all surveys.
- Estimated survey pages visitation peak over 24 hours (for instance if surveys are only visited by internal users within a company during work hours).

3.3.2. Example Scenario

A customer with 100 Authoring users expects to launch 2000 surveys over the course of a year. 500 of these will be launched in February, with an expectancy of 20 survey page hits per respondent to complete a survey. 1,000,000 completed responses are expected over the full year, with 150,000 of these expected in February, with 80% of the total page visits occurring during the working hours between 8am and 6pm, and 60% between 12pm and 4pm. The second year, a usage growth of 25% in response data collection is estimated. The following three years, 30% growth from the year before. Additionally, 25 new Authoring users will be hired each year.

From this, we can calculate that about 150,000 completed responses x 20 page hits will yield 3,000,000 page hits for the completes only. However, adding incompletes will result in 3,000,000 x (100/20) = 15,000,000 page hits in total in February alone. Spread out over 20 working days gives 750,000 page hits per day. 80% of these (600,000) hit the server in a ten-hour window, resulting in a total of 60,000 page hits on average, with 450,000 occurring over 4 hours – 112,500 page hits per hour in the peak period. The second year: ~140,000 page hits. The fifth year: 112,500*(1+25/100)*(1+30/100)*(1+30/100)*(1+30/100) resulting in ~310,000 page hits/hour peak. Additionally, there will be 225 Authoring users in the fifth year.

Based on this information, we can set up a site recommendation capable of handling the expected loads. In the first year, the mid-level scenario below should be sufficient, although by the fifth year, a layout somewhere between the mid-level and high-end setups would be recommended.

3.3.3. Entry-Level Multi-Server Setup

The minimum recommended configuration consists of a tri-server configuration, with dedicated database and deployment servers for data collection priority, and a combined Authoring/task system server for a small amount of project managers. In this scenario, the combined Authoring/task system server also runs the Metadata Rest API since this does not add noticeable strain to the server on which it runs with this amount of load.

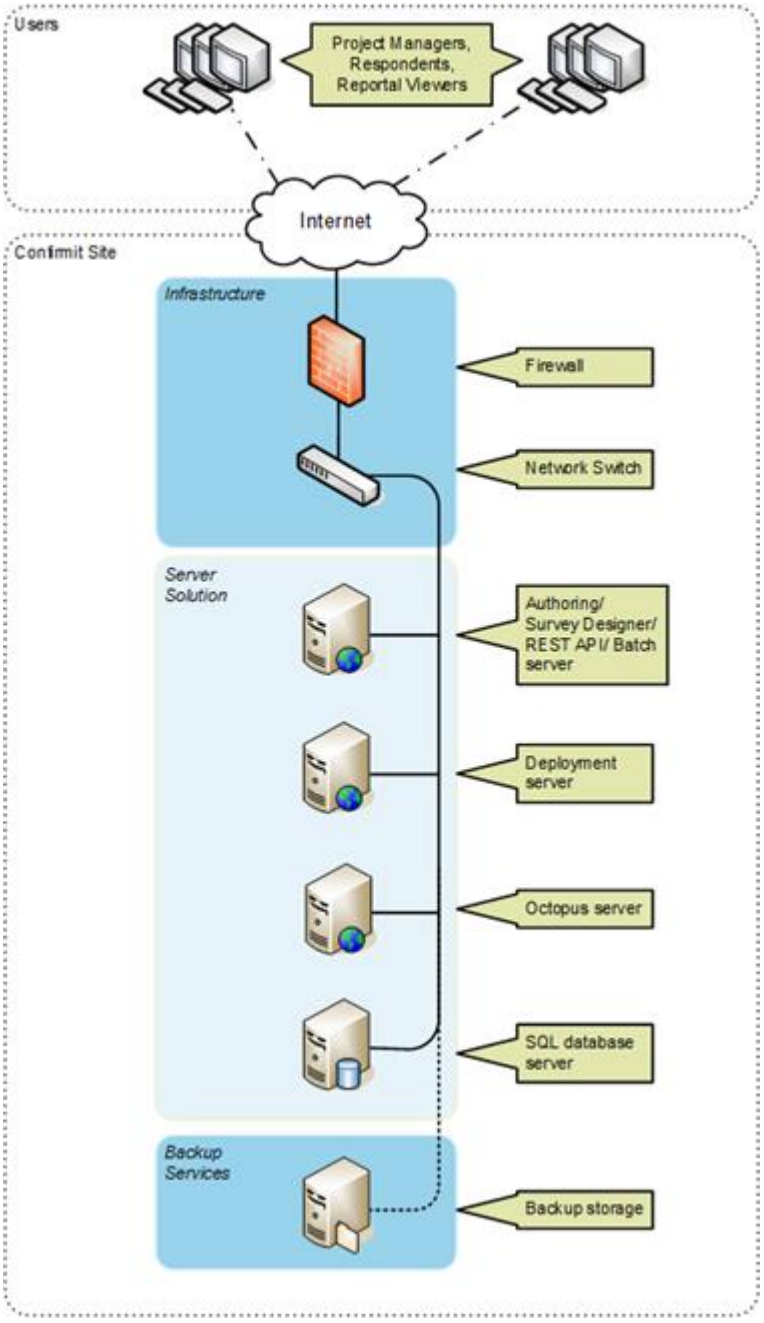


Figure 7 Features divided across three servers

The proposed configuration should be capable of handling approximately 40 concurrent Authoring users, and at least 120000 survey page hits per hour.

3.3.4. Mid-Level Multi-Server Setup

If service availability is starting to become a critical factor, one can start adding several servers into the same role to provide service redundancy, as shown in the example below. The proposed scenario would be recommended when data collection is of high importance, while temporary loss of Authoring services would be less critical.

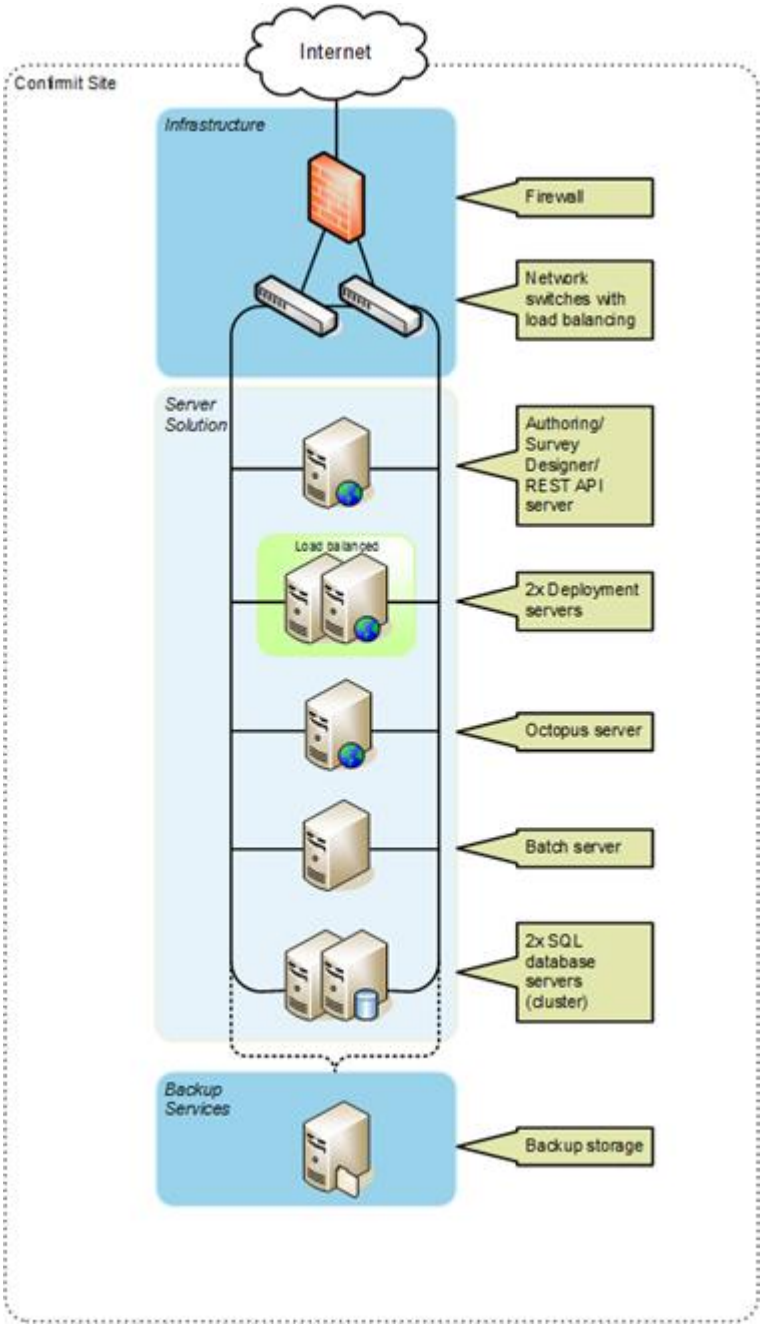


Figure 8 Multi-server setup with redundancy for data collection and response storage

The proposed configuration should be capable of handling approximately 75 concurrent Authoring users, and at least 250000 survey page hits per hour.

3.3.5. High-End Multi-Server Setup

When extreme performance and redundancy in all parts of the application is required, even more servers and network infrastructure can be added to the setup:

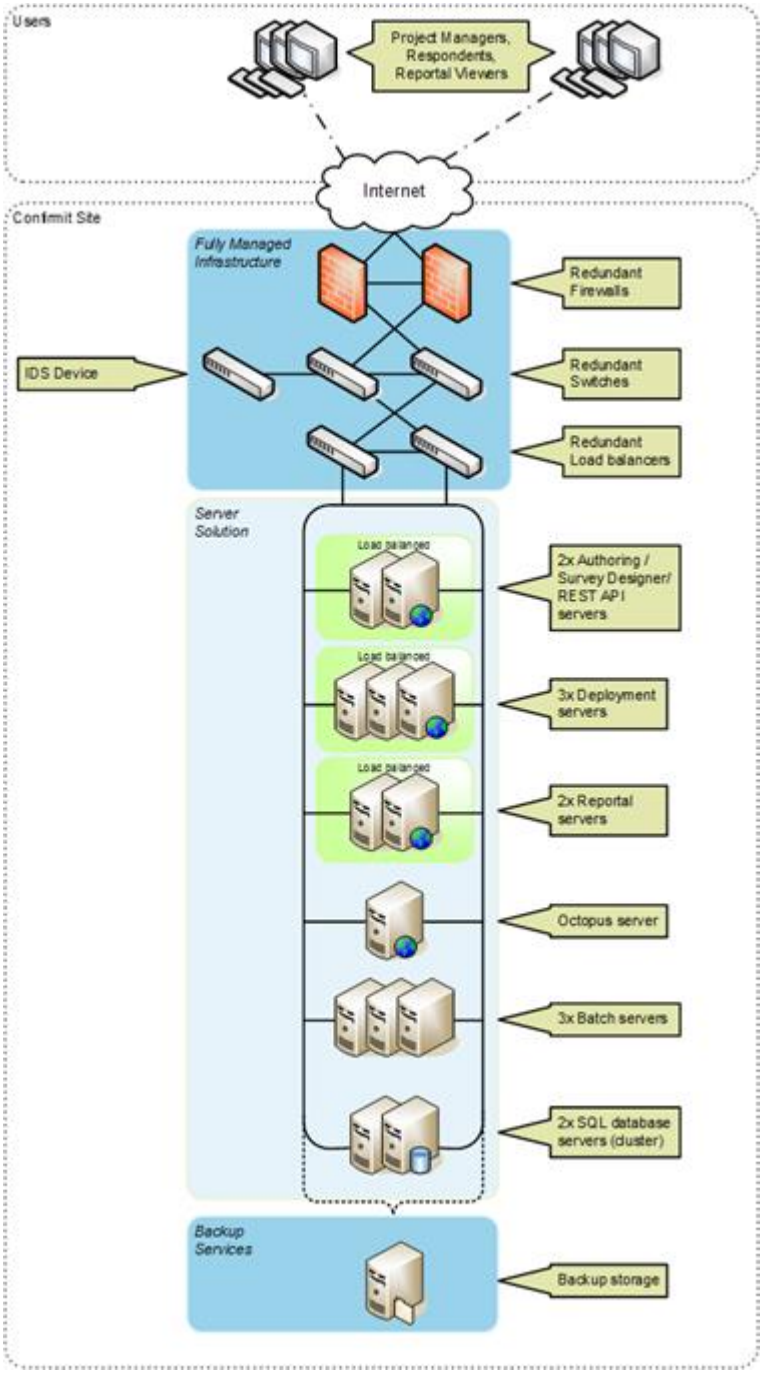


Figure 9 Full redundancy for user-facing features

The proposed configuration should be capable of handling approximately 250 concurrent Authoring users, and at least 400000 survey page hits per hour. Larger environments can be suggested on request.

3.4. Virtual Infrastructure

Virtualization of servers has become a hot topic since many server application environments do not utilize their hardware at an optimal rate. Enterprise virtualization applications have been making way for more optimal processing power distribution, and we receive requests about virtualization support more often than before.

Confirmit runs SaaS services hosted with a Microsoft Gold-certified partner with a strong technology experience and an advantage in communication with third parties, so we have chosen to run our environment in a VMware Infrastructure-based environment for our own hosted Confirmit installations. We have taken recommendations from our hosting provider's dedicated virtualization team to ensure the best possible performance for our environments. No database services are being run under virtualization as this was not recommended for the amount of I/O traffic generated by the Confirmit application on our systems, as it could potentially result in overall performance degradation.

Confirmit supports installation in virtual infrastructures using VMware vSphere 5 and above.

Disclaimer: Limited support for virtualized servers.
As of the date of this manual, Confirmit supports the use of the Horizons software on client equipment that is virtualized, subject however to the limitations set forth herein. Confirmit itself runs VMware on those servers that are virtualized on the Horizons SaaS environments, and Confirmit has not run tests of virtualization on other virtual platforms. Confirmit does not run virtualized database servers for Horizons SaaS and we have no first-hand experience with this. Our recommendation for companies wanting to run high volume / high strain environments, with high I/O requirements, is to run database servers on physical hardware, as we do ourselves. We do however have On-Premise clients that run virtualized database servers for Horizons, and as of mid-2016 we have not been informed of any mayor performance issues related to the virtualization. That said, those clients run low volumes / strain and have low to medium I/O requirements. Please note that Horizons software databases must be installed and hosted on a dedicated SQL instance. In the event you were to experience issues with the Horizons software that, to Confirmit's reasonable discretion, appear to be related to the underlying virtualization infrastructure's availability or ability to deliver sufficient performance to virtual servers, then Confirmit may not be able to support you in relation to such issues unless you are able to reproduce such issues on physical hardware without VMware.

4. Preparing an Installation

Confirmit is an extensive software suite that relies on different underlying components, and configurations on the server to be made, for many of its features. It is therefore important that servers built to host the Confirmit software are set up with the correct requirements before attempting to install Confirmit. Confirmit strongly recommends that servers intended for running Confirmit are dedicated for this purpose to avoid conflicts with other software. Support for problems running the software due to such conflicts will be chargeable if the cause of the problem is proven to be the conflicting software.

A condensed list of requirements and the configurations that must be made before running the Confirmit setup follows. Each item is discussed in detail later in the manual.

Note: To ensure you have the required prerequisites installed for the version of Confirmit you intend to install, please ensure you use the checker tool outlined in section 4.3 Be aware that some versions of components can change between releases. By using the checker tool you ensure you have the correct prerequisites for the Confirmit release you are installing.

Windows Configurations	Confirmit Server	Database Server	Comments	Additional info
Windows Server 2016	Required	Required	Changed from v24.	Windows Server 2016 (Long-Term Servicing Channel) is the supported OS for Confirmit Servers (both Application and SQL).
File Server	Required		Enabled as a Role in Windows Server Manager / Application Server / File and Storage Services / File and iSCSI Services	This role adds several additional features to support this role.
Web Server (IIS) Support	Required		Enabled as a Role in Windows Server Manager / Application Server	This role adds several additional features to support this role.
HTTP Activation	Required		Enabled as a Role in Windows Server Manager / Application Server 7 Windows Process Activation Service Support	This role adds several additional features to support this role.
Microsoft .NET Core 2.2.X	Required			On download page, locate correct version of ASP.NET Core/.NET Core: Runtime & Hosting Bundle. We recommend you always update to the latest patch version of .NET Core 2.2 to make sure you have all their security fixes. NOTE: Confirmit has not been tested on versions lower than 2.2.3
Microsoft .NET Framework 4.7.2	Required			Current release: 461814. See https://msdn.microsoft.com/en-us/library/hh925568(v=vs.110).aspx for mapping between

Windows Configurations	Confirmit Server	Database Server	Comments	Additional info
				release and version
Microsoft .NET Framework 3.5 includes .NET 2.0 and 3.0)	Required		Enabled as a Feature in Windows Server Manager / .Net Framework 3.5 Features	
ASP.NET 3.5	Required		Enabled as a Role in Windows Server Manager / Web Server (IIS) / Web Server / Application Development	
ASP.NET 4.5	Required		Enabled as a Role in Windows Server Manager / Web Server (IIS) / Web Server / Application Development	
.NET Extensibility 3.5	Required		Enabled as a Role in Windows Server Manager / Web Server (IIS) / Web Server / Application Development	
.NET Extensibility 4.5	Required		Enabled as a Role in Windows Server Manager / Web Server (IIS) / Web Server / Application Development	
IIS Management Console	Required		Enabled as a Role in Windows Server Manager / Web Server (IIS) / Web Server / Management Tools	
IIS 6 WMI Compatibility	Required		Enabled as a Role in Windows Server Manager / Web Server (IIS) / Web Server / Management Tools / IIS 6 Management Compatibility	
SMTP	Required		Enabled as a Feature in Windows Server Manager.	This feature adds several additional features to support this feature. NB SMTP service must be started after installation.
Application Initialization	Recommended		Enabled as a Role in Windows Server Manager / Web Server (IIS) / Web Server / Application Development	Allows application pools to be pre-loaded when deploying Confirmit with Octopus deployment.
SQL Filestream Enabled		Required		
Create directories on SQL server for SQL data, SQL Log and SQL Filestream		Required		

Windows Configurations	Confirmit Server	Database Server	Comments	Additional info
ConfirmitDeployer account	Required		Must be created, logged into, and have Administrator permissions when running setup.	Can be local Windows user or a domain user. Note the 20 character SAM account name length limit if using domain accounts: http://msdn.microsoft.com/en-us/library/ms679635.aspx This account cannot include a semicolon in the password.
Confirmit account permissions	Required		Allow "Log on as a service" and "Log on as a batch job"	
Set a CLIConfg Database Alias [32 bit and 64 bit]	Recommended		Setting up a server alias is strongly recommended as this will provide flexibility for future usage.	External guide can be found here
Software				
Microsoft Visual C++ 2008 Redistributable Package [x86], v9.0	Required		Download	
Microsoft Visual C++, V14, [x86]	Required		Download	
Web Services Enhancements 2.0 SP3 [x86]	Required		Download	
Web Services Enhancements 3.0 [x86]	Required		Download (Use Setup Type - Administrator)	
Open XML SDK 2.5 for Microsoft Office [x86]	Required – Authoring / Batch		Download	
Node.js v10 [x64]	Required		Download	We recommend you always update to the latest release of Node.js version 10 to make sure you have all their security fixes. NOTE: Confirmit has not been tested on versions lower than 10.15.3
IIS URL Rewrite Module 2 [x64]	Required		Download	
IISNode for IIS 7.x	Required		Download	

Windows Configurations	Confirmit Server	Database Server	Comments	Additional info
[x64]				
RabbitMQ [x64]	Required		Download	Verify the version supported by Confirmit using the pre-req checker.
SQL Server 2016/2017, Database Engine [x64]		Required		SQL 2016 requires SP1 as a minimum.
Microsoft SQL Server 2016/2017 Integration Services (SSIS) [x64]	Task System Server		Management Tools - Complete must be installed	Requires SQL License
Microsoft Powershell 5 and above	Required			Powershell 5.0+ is required on all Confirmit servers. 5.1 is recommended.

4.1. Operating System and Third-Party Components

Confirmit Horizons v2019.6.31 supports installation on the Windows Server 2016 (Long-Term Servicing Channel) operating system.

4.1.1. Microsoft Visual C++ 2008 Redistributable Package

The Visual C++ Redist package installs runtime libraries of Visual C++ libraries required to run components of Confirmit.

4.1.2. Web Services Enhancements 2.0 SP3

Web Services Enhancements 2.0 SP3 is a required component.

A full (Administrator) WSE 2.0 SP3 installation is only required if the x509 certificate signing tool is required (for servers handling Confirmit CAPI or Debug Station distribution certificates). For other servers, only the WSE Runtime Distributable is required.

4.1.3. Web Services Enhancements 3.0

Web Services Enhancements 3.0 for .NET 2.0 is a required component.

A full ('Administrator') WSE installation is only required if the x509 certificate signing tool is required (for servers handling Confirmit CAPI or Debug Station distribution certificates.) For other servers, only the WSE Runtime Distributable is required.

4.1.4. Open XML SDK 2.5

The Open XML SDK 2.5 is an open standard that defines a set of XML schemas for representing spreadsheets, charts, presentations and word documents. This is used by Confirmit for office related exports and is required on servers with the Authoring and Task System roles.

Assembly "DocumentFormat.OpenXml.dll", version is 2.5.5631.0.

4.1.5. Node.js

Node.js is a web server runtime that is required by some components of Confirmit.

Note: If you use self-signed or privately signed certificates in your environment, you will need to do some additional configuration to make Node.js accept them. Refer to the description for the Confirmit.Server.NodeExtraCACertsFilePath Octopus variable in Appendix C: Octopus Roles and Variables.

4.1.6. IISNode for IIS

IISNode is a native IIS module that allows hosting of node.js applications in IIS on Windows.

4.1.7. IIS URL Rewrite

URL Rewrite Module provides a rule-based rewriting mechanism for changing requested URL's before they get processed by web server and for modifying response content before it gets served to HTTP clients.

4.1.8. RabbitMQ

RabbitMQ is a message broker that Confirmit requires for communication between the many services that runs within the application. It can run as a single instance, or as a cluster for better performance and/or high availability. RabbitMQ must be running on default port (5672) on at least one server. For information on how to setup RabbitMQ, see Appendix B – Installing and Configuring RabbitMQ.

Note: RabbitMQ requires new ports to be opened (see Network Traffic Requirements Between Servers Internally Within the Site on page 13 for more information).

4.2. Additional Components

Some of the main Confirmit features require third party software. Software and licenses are to be purchased, installed and maintained by the Client, at the Client's own expense, unless the 3rd party software is distributed with the Confirmit installation.

It is recommended that servers are dedicated to running Confirmit and that no other software than the requirements (except for monitoring/management tools) and Confirmit are installed. This is to avoid any potential conflicts with other applications running on the same servers. If the investigations from a support request identify problems caused by other software, the support may be deemed chargeable as per the support agreements.

The operating system components and 3rd party applications that are needed will depend on the Confirmit features installed on a server.

4.2.1. Task System Features (Batch Server)

4.2.1.1. Microsoft SQL Server Integration Services (SSIS)

SQL Server Integration Services 2016 must be installed on batch servers to support import features and sampling for Community Panels and respondent uploads. SSIS is also used by Data Central Server Rules, Excel exports from Confirmit Express, and respondent upload to CATI projects. SSIS might require additional SQL Server licenses for batch servers that are set up to run these task types. Note that you will also need "Microsoft SQL Server Management Tools - Complete" installed as part of the SSIS installation.

Note:
- Upgrading from previous versions of SSIS requires a re-installation of Confirmit on the server where SSIS resides.
- A re-installation of Confirmit will also be needed if SSIS is installed after Confirmit has been installed.
- The 'SSISRuntimeVersion' variable under Batch servers roles in System Configuration must be set to the correct SQL version before re-deploying. Refer to Appendix A for the variable definition.

4.2.2. Confirmit API Development

To connect to Confirmit APIs, SOAP (Simple Object Access Protocol) is required.

4.2.3. Log Files

Starting with features introduced to General Availability in Confirmit Horizons v22 Continuous Deployment version, some application and service logs are stored in a new location and format. Where logs traditionally were stored in SQL, they are now logged directly to disk using a built-in logging framework. The log directory for these logs is determined by the **Confirmit.Site.LogPath** variable in Octopus. As with IISlogs, there may be reasons why logs need to be kept in an unchanged format for an extended period, so these logs are not cleaned up by the Horizons application. We strongly recommend that you determine your requirements for keeping logs for audit purposes, and clean them as necessary (see Horizons Logging Format on page 64 for more information).

4.2.4. Database Server - Version Support

The versions currently supported for all database components are SQL Server 2016 and 2017.

4.2.4.1. Microsoft SQL Server Editions

Microsoft SQL Server comes in various editions that greatly vary in price. Choosing the correct version for your environment is important in order to provide the necessary features required by Confirmit.

In most cases, SQL Server Standard Edition will be sufficient for running Confirmit in a production environment. SQL Server Standard Edition also supports installation in a fail-over cluster, albeit only with two servers in the cluster, and database backup compression. For enterprise installations, SQL Server Enterprise Edition can be considered for high availability features such as 16-node cluster support, encryption at rest, online indexing, online restore, fast restore and support for multiple partitions in SQL Server Analysis Services (which may be used in the Panelist reporting feature).

To determine which edition of SQL Server is right for your environment, the following link might be of help:

> https://docs.microsoft.com/en-us/sql/sql-server/editions-and-components-of-sql-server-2016?view=sql-server-2017

SQL Server should be installed with Mixed-mode authentication to allow SQL login authentication from applications rather than using Windows logins. It is recommended that SQL Server is always installed with the SQL_Latin1_General collation. It is highly recommended to set a strong password for the ‘sa’ account or disable the account altogether after installation.

Note: Confirmit requires access to an SQL login with ‘sysadm’ permissions during installation but the account is not needed after the initial installation is complete. As Confirmit v(Undefined variable: AVariables.ConfirmitRev) does not use any login account with sysadm permissions, a login may have to be temporarily enabled to allow future upgrades to be performed.

Note: If installing the SQL Server service to run under a named local or domain account, it is also recommended that the same account is used for the SQL Server Agent service in order to prevent issues with permissions to database files on the file system level.

Note: If using Confirmit CATI, the user account for the SQL Server Agent Service must be added as a local Administrator user on all SQL server instances.

4.2.4.1.1. Required Updates

SQL Server 2016 Service Pack 1 is the minimum requirment for installing Confirmit Horizons v(Undefined variable: AVariables.ConfirmitRev). It is recommended to always keep SQL up to date with Microsoft’s Service Pack and Cumulative Updates.

4.2.4.1.2. Licensing

Due to the way Confirmit uses connection pooling to connect to the database server, a CPU-license is usually the most feasible option as the amount of CALs required for a regular per-user license may not always be clear. For servers running SSRS/SSIS, licensing recommendations will depend on the number of users connecting to the individual features. Assume one CAL license is required for each Confirmit user accessing either SSIS or SSRS will give an estimate of the cost versus a CPU license.

4.2.4.2. Configuring the Model Database

The Model database is set up when SQL Server is installed and works as a template for new databases –settings on the Model database will be inherited by all new databases created on the system, and some settings should be configured before installing Confirmit in order to ensure Confirmit databases are created with recommended options. The database properties can be adjusted in the SQL Server Enterprise Manager.

- **Autogrow –** should be set to allow databases to grow in size automatically when needed. If this setting is not enabled, Confirmit will produce errors similar to "The primary filegroup for database <db name> is full", and a larger size will have to be adjusted manually. We strongly recommend enabling the autogrow feature for both data and log files. Note: The default option is to grow the database by 1MB of the database size. This can lead to high fragmentation, so we suggest setting the initial size to 15MB, and autogrowth to 10MB. For databases that become large (10GB+), this option can be changed individually on each database to set a fixed MB size growth for each growth operation. This may prevent issues with autogrow operations timing out and causing issues on the server.
- **Autoshrink –** when enabled, the database system will periodically optimize database white space and shrink the file size of the database. The feature will produce quite some performance overhead, so in most cases this should be disabled, and instead periodical maintenance jobs may be scheduled to shrink databases at given times if disk space becomes an issue
- **Autoclose –** when enabled, the feature will automatically shut down a database when all connections to it have been closed. The setting must be disabled as it could potentially cause substantial performance overhead on the database server.
- **Recovery model –** Depending on the backup strategy in use, simple or full transaction logging is recommended. For full/differential database backups, simple transaction logging is sufficient, while for systems where point-in-time recovery is required, full transaction logging is required. Refer to Microsoft KB articles related to SQL recovery model for more information.

Important
Some Confirmit databases (particularly [aspstate], [confirm] and [confirmlog]) can produce very large transactions logs for active systems if full transaction logging is enabled. If full transaction logging is used, setting up a maintenance plan for periodically creating checkpoints and backing up transaction logs is paramount, particularly for these two databases.

4.2.4.3. Enable FILESTREAM in SQL Server

FILESTREAM is required on the main (first) SQL Server instance for installing Confirmit Horizons v(Undefined variable: AVariables.ConfirmitRev). This can be enabled by opening the SQL Server Configuration Manager, opening properties for the SQL instance running the Confirmit System databases, opening the FILESTREAM tab and checking the Enable FILESTREAM box:

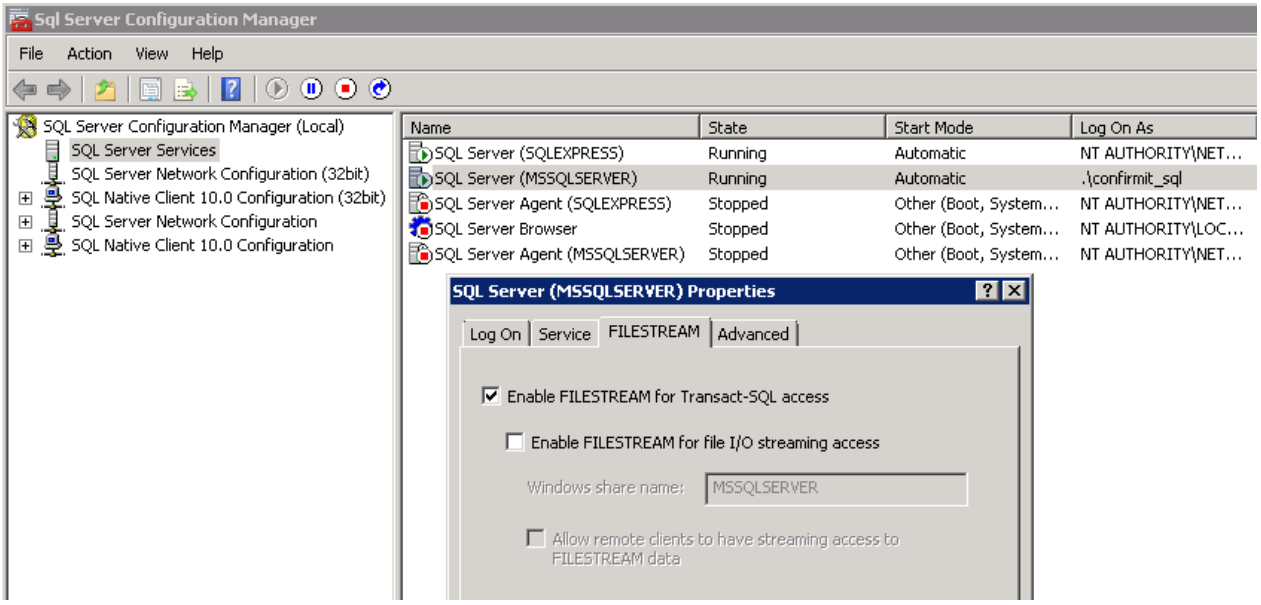


Figure 10 Enabling FILESTREAM

Finish the configuration by running the following statement on the server:

```
exec sp_configure'show advanced',1
go
reconfigure
go
exec sp_configure'filestream access level', 2
go
reconfigure
go
```

The setting can be enabled without having to restart SQL Server.

Note: If you have an SQL cluster, remember to enable Filestream on both cluster nodes.

4.3. Prerequisite Checker

Included as part of each release is a prerequisite checker to help ensure your site has all the correct components installed to ensure a properly functioning Horizons site. There are two options included with each deployment; a stand-alone checker which can be executed without any prior configurations related to the Horizons installation, and a checker that is built in to the Octopus deployment system.

Note: This updated prerequisite checking functionality is only intended for Horizons versions 21 and later. Download the discontinued pre-req checker for versions 20 and earlier - this is also available from the Extranet.

The following sections outline where you can obtain the prerequisite checker tools, and describe how you use them.

4.3.1. Stand-alone Prerequisite Checker

A stand-alone checker, which can be executed without any prior configurations related to the Horizons installation, is available from the Extranet.

4.3.1.1. Downloading

Extranet

Check the Extranet for the most recent stand-alone tool for the version you intend to install. Download and extract the file to the same directory locally on the server.

Octopus

From a server that has a connection to the Confirmit Chocolatey feed, the stand alone pre-req package can be downloaded via the following command:

```
choco install Confirmit.PrereqChecker
```

The package will be downloaded and extracted to a directory called 'Pre-req Checker'. This directory can then be copied out and executed on servers you plan on checking.

4.3.1.2. Execution

Execute PrereqGUI.ps1 from Powershell with administration permissions (from locally in the directory).

4.3.1.3. Interface

Figure 11 The Prerequisite Checker interface

Server Roles

Depending on which Confirmit Horizons roles you will be hosting on the server you are preparing for installation, check the required options.

SQL Input

Enter a valid SQL user and host address (alias or hostname/IP) to validate the connection to your sites system SQL server (SQL aliases should be set up in advance of running this tool).

Confirmit Deployer

This is the local or domain Windows user you intend to use for the installation. This will check that the user is able to run as a server; this is required for the Task System and Bitstream services. Refer to related server manual sections for more information on these requirements.

4.3.1.4. Report Generation

Once the required roles have been selected and any optional SQL and Confirmit Deployer account details have been entered as required, click **Run Pre-Req Checker** to start the process. An HTML report is generated and presented with the status of required components for the roles you have selected.

Note: This report.htm is saved to the same location as the prerequisite checker.

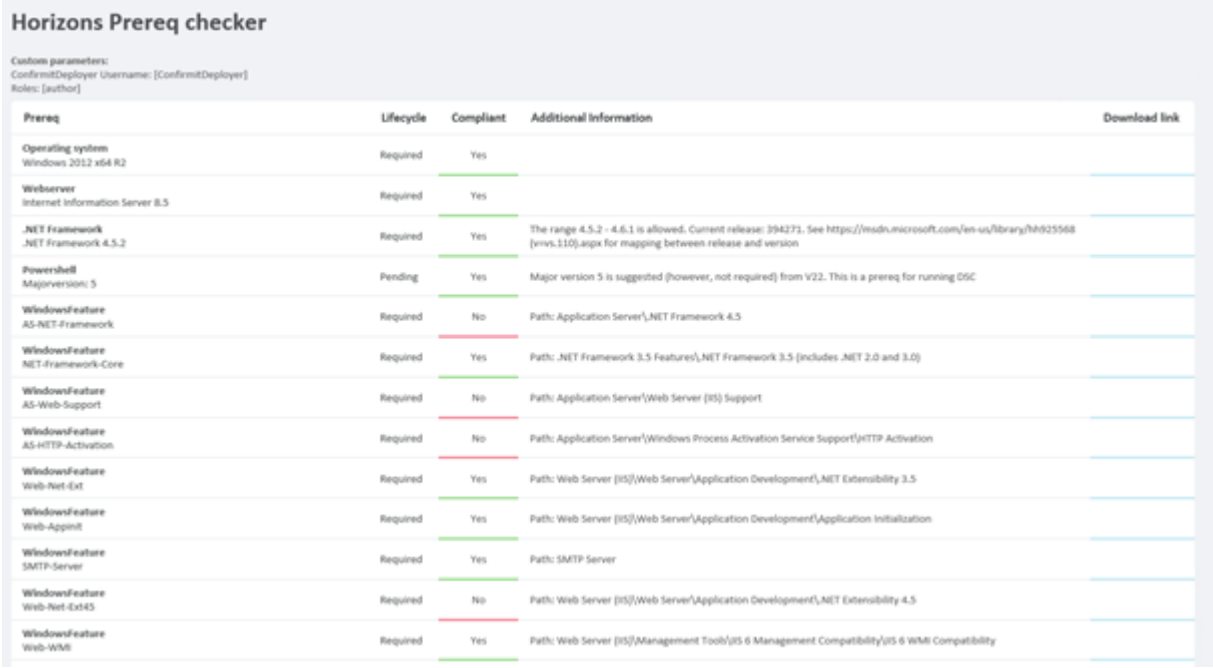


Figure 12 Example of the prerequisite checker report

Components that require separate installers can be downloaded using the **Download** button (a connection to the internet is required). Example below.

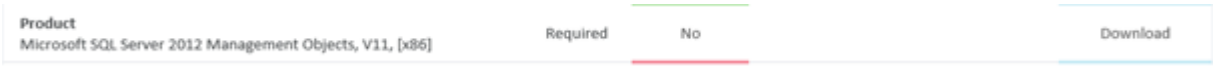


Figure 13 Example of a component that requires a separate installer

4.3.2. Octopus Deployment Prerequisite Checker

Octopus deployment is able to check the status prerequisites across all servers that have a role in your Octopus deployed site, during each deployment. This allows you to ensure your site is always compliant with the release of Horizons you intend to deploy.

Note: To check which servers are listed in Octopus, check the servers under the 'Environments' tab.

The prerequisite check is listed as a step during the release deployment that can be run as part of every deployment, skipped if required, or run on a specific subset of servers. An HTML report is generated for each server that is not fully complaint.

Note: An Octopus server and servers you intend to use for your Horizons site must already be created, installed and configured, to the point where you are ready or have already installed Horizons for this functionality to be available. The stand-alone prereq checker is intended for use on sites that have not yet been integrated into Octopus deployment.

4.3.2.1. Executing a Prerequisite Check

The prerequisite check is performed as a step during deployment. As previously mentioned, this is run during every deployment, but can be skipped by disabling this step during deployment.

Default prerequisite check behaviour

By beginning a deployment, the prerequisite check is performed at step two (2.Confirmit.PrereqChecker). As with other parts of the deployment, the prerequisite check nuget package will be copied out to each machine (Acquire package) and then the check will be executed.

This step can be monitored via 'Task Summary' or for a more detailed via, 'Task Log' tabs.

Once completed, a summary of compliance for each server can be reviewed via the Task Log tab. Expand the sections related to each server you are reviewing. Below is an example of the 'Task Log' preformed against a server with the 'MetadataAPI' role.

```
Deploying package 'C:\Octopus\Files\feeds-1\********.PrereqChecker.2.1.4_0EF4F55075B5804989C5E77FDF8E172A.nupkg' to machine 'https://nor-osl-rest02:10933/'
Deploying package:    C:\Octopus\Files\feeds-1\********.PrereqChecker.2.1.4_0EF4F55075B5804989C5E77FDF8E172A.nupkg
Starting PostDeploy
Running prereq checker
Machine Roles:  MetadataApi
Machine Roles count: 1
Custom parameters: ********DeployerUserName: [NORDIC\nor_CsOe]
Compliant Required prereq:  Operating system: Windows 2012 x64 R2
Compliant Required prereq:  Webserver: Internet Information Server 8.5
Compliant Required prereq:  .NET Framework: .NET Framework 4.5.2
****************************** Non-compliant Pending prereq ***********************************
* .NET Framework 4.6.2: 4.6.2. will become required from V22 Current release:  394271.  See https://msdn.microsoft.com/en-us/library/hh925568(v=vs.110).aspx for mapping
between release and versionDownload link: https://www.microsoft.com/en-us/download/details.aspx?id=53345
***********************************************************************************************
Compliant Pending prereq:  Powershell: Majorversion: 5
Compliant Required prereq:  WindowsFeature: AS-NET-Framework
Compliant Required prereq:  WindowsFeature: NET-Framework-Core
Compliant Required prereq:  WindowsFeature: AS-Web-Support
Compliant Required prereq:  WindowsFeature: AS-HTTP-Activation
Compliant Required prereq:  WindowsFeature: Web-Net-Ext
Compliant Required prereq:  WindowsFeature: Web-Appinit
Compliant Required prereq:  WindowsFeature: SMTP-Server
Compliant Required prereq:  WindowsFeature: Web-Net-Ext45
****************************** Non-compliant Required prereq **********************************
* Web-WMI: Path:  Web Server (IIS)\Management Tools\IIS 6 Management Compatibility\IIS 6 WMI Compatibility
***********************************************************************************************
Compliant Required prereq:  Service: DisplayName: Simple Mail Transfer Protocol (SMTP)
Compliant Required prereq:  Product: Microsoft Visual C++ 2008 Redistributable, V9.0, [x86]
Compliant Required prereq:  Product: Microsoft WSE 3.0, V3.0, [x86]
Compliant Required prereq:  Product: Microsoft WSE 2.0 SP3, V2.0, [x86]
Compliant Required prereq:  Product: Open XML SDK 2.5 for Microsoft Office, V2.5, [x86]
Compliant Required prereq:  Product: Microsoft System CLR Types for SQL Server 2014, V12, [x64]
Compliant Required prereq:  Product: Microsoft System CLR Types for SQL Server 2012, V11, [x86]
Compliant Required prereq:  Product: Microsoft SQL Server 2014 Management Objects, V12, [x64]
Compliant Required prereq:  Product: Microsoft SQL Server 2012 Management Objects, V11, [x86]
Compliant Required prereq:  Product: Node.js, V6.6.0, [x64]
Compliant Required prereq:  Product: IIS URL Rewrite Module 2, V7.2.2, [x64]
Compliant Required prereq:  Product: iisnode for iis 7.x (x64), V0.2.18.0, [x64]
Compliant Required prereq:  Product: Microsoft SQL Server Data-Tier Application Framework (x64), V12, [x64]
Compliant Required prereq:  Database Connectivity: Open connection towards database [confirm_system]
Error: Non-compliant Required prereq detected!
Creating full prereq report with HTML-formatting. Report can be found within the 'Release' details in Octopus
Using the following folder temporary storage:  [C:\Windows\TEMP]
Creating artifact in Octopus for the report
Report published.
Failed to run package script:
Incompliant prereq found
```

Figure 14 Example of the 'Task Log' preformed against a server with the 'MetadataAPI' role

Note: This example indicates there is a non-compliant pending prerequisite to upgrade .NET framework to version 5, and a required prerequisite 'IIS 6 WMI Compatibility', an IIS feature. This information can then be used to plan the upgrading of prerequisites if required.

From the output above, we see the required prerequisite Windows features and stand-alone products listed and checked for compliance. Further information on what these requirements are can be found in the report that is generated . See 'Report Generation' below.

Skipping the prerequisite check step

When beginning deployment, 'untick' the step if it is not required and it will be skipped.

Figure 15 Deselecting a prerequisite step

Performing the check on a select number of servers on the site

A specific server or sub-set of servers:

1. Select the **Advanced** option (next to **Deploy now**).

2. Select **Deploy to a specific subset of deployment targets**.
3. Add the machines you would like to check.
4. Select **Deploy Release**.

The figure below shows an example from Confirmit's SaaS US site when selecting a server to run the check against.

Figure 16 Example of selecting a server

Select **Skip - All** and then enable the prerequisite checker step to ensure only the prerequisite checker step is run and only on the servers you have selected.

4.3.2.2. Octopus Report Generation

An HTML report is generated for all servers that are non-compliant, which provides further detail regarding the compliance status for that particular server. These reports are located under the 'Artifacts' section which is available on the Deploy page where Task Summary and Task Log are also located.

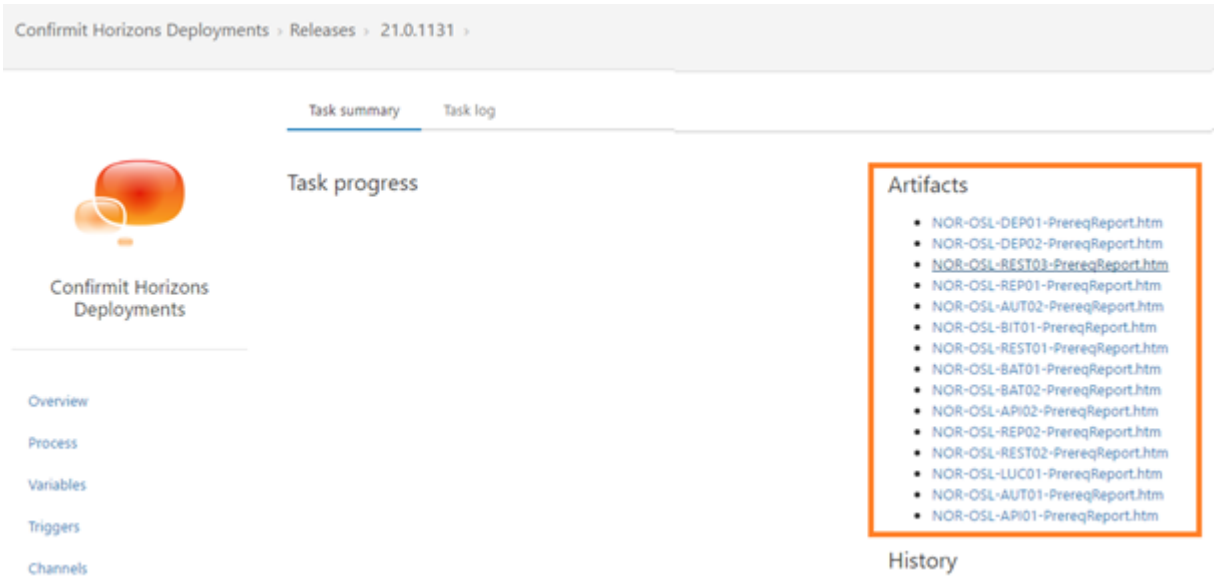


Figure 17 Example of an HTML report

As with the stand-alone prerequisite checker, you can use these reports as a reference to update required features and download and install required components. See the stand-alone prerequisite 'Report Generation' section for further information.

5. Installing Confirmit

After the servers have been set up using Octopus as the deployment method (currently administered by Confirmit Operations team members), all connectivity has been tested and Confirmit system requirements have been installed, the next step is to install Confirmit itself. Installing **Confirmit Horizons v(Undefined variable: AVariables.ConfirmitRev)** is only available via the Octopus deployment method.

5.1. Upgrading from an Existing Version

For an upgrade, most of the data needed will be fetched from the system databases, and does not need editing in the setup. Any discrepancies are outlined below.

- Confirmit Horizons v24 is only available via Octopus deployment. If your site is not already using the Octopus deployment system a conversion is required from the installer setup. This is only available from v20 and v21 Frozen. Assistance is required by the Confirmit Operations team to complete this process.
- The only supported upgrade path to Confirmit Horizons v24 is from Confirmit Horizons v18.5 and newer. For earlier versions, in addition to ensuring the correct SQL version and Windows Server version, you must apply the latest patch of v17.5 and then migrate from Fast Objects (FO) to SQL before installing v18. Assistance is required by the Confirmit Operations team to complete this process.
- For systems running versions v17.5 and older, due to the complexity of the FO to SQL migration we strongly recommend building v24 on a new site rather than performing an upgrade if possible.

For all upgrade assistance and inquires please contact support@confirmit.com or your account manager.

Note: Windows Server and SQL versions can change between the release versions of Confirmit Horizons. Please refer to the 'Overview of Operating system and SQL requirements for Confirmit versions' document under 'Fact Sheets' on the Extranet to better understand the OS and SQL requirements for the version you intend to upgrade to.

5.1.1. Confirmit Licensing

Confirmit is protected software and requires a license file to work. Confirmit is not installed with a standard license.

In order for Confirmit Technical Services to provide you with a valid license file, a **Site ID** value for the site you are installing on is required (see Obtaining and Installing a License File on page 43 for more information).

If you are upgrading from a version that already uses a license, the existing license should continue working with Confirmit Horizons v(Undefined variable: AVariables.ConfirmitRev). Licensable add-ons introduced in newer versions than the one you are upgrading will not be enabled automatically; a new license file enabling these add-ons will have to be installed, even if those add-ons are included in the contract or are provided free of charge with a new version.

Note: In a virtual environment, make sure that the server clocks on all hypervisors are in sync before syncing time with virtual machines. If the clock on a VM rolls back, this may invalidate the license as it will be perceived as tampering to extend the license period.

5.1.2. Keeping Existing Settings

If you upgrade an existing version of Confirmit, .NET configuration files (web.config files for web applications), **<servicename>.config** files for system services, and web folder configuration) will be overwritten by the setup. If you have edited the files to include custom settings (for instance referencing installed custom code), you should back up the files from their original locations and take note of the IIS configuration (or back up your IIS metabase) before proceeding with the upgrade.

Note: All configuration settings in the Confirmit system configuration will be kept during an upgrade. These will have been entered into the Octopus variable set before first installation by the Confirmit Operations team assisting with the installation.

Important
As some new settings have made their way into Confirmit's .config files (for .NET applications) these files should not be replaced with older versions. Instead, it is recommended to edit the new files and paste in the configuration changes required from older ones, such as session state handling, links to custom code libraries, custom error documents, etc.

5.1.3. SSL Requirements

Upgrading to v(Undefined variable: AVariables.ConfirmitRev) requires SSL settings to be defined in Octopus (Confirmit.Site.SSLEnabled = TRUE/FALSE). If the site is already using SSL for HTTPS this variable should already be set to TRUE. However if the site does not use HTTPS then it should be set to FALSE.

If this variable is not defined the site the Identity Provider will reject non secure connections and the upgrade will fail during deployment. See Appendix C for a full list of variables used in Octopus.

Important
If you change this variable, you will need to update the snapshot variables for the release and force redeploy of the packages. If this is not done, the updated variable will be persisted when the next release is loaded onto the Octopus server.

5.1.4. Back-up Confirmit System Databases

Confirmit system databases should be backed up before an update in case a rollback is required (see SQL Server Databases on page 56 for more information). Confirmit Survey databases are not touched during an upgrade; schema updates occur when the survey is launched after upgrade.

5.2. Pre-Installation

5.2.1. Creating Data Folders for SQL Databases

Prior to installing, data folders for SQL databases, log files and the SQL Filestream databases must be created on the database server. If the data folders do not exist, the Confirmit setup cannot continue and will display a warning message stating this.

5.2.2. Setting up User Accounts for Confirmit System Services

Confirmit installs a system service on servers installed with the 'Batch' role that must be run under user specific credentials. This applies to the Confirmit Task System service, which runs scheduled tasks such as deploying files onto other servers. Since the service requires file access to remote servers, it cannot be run under the Local System credentials. The credentials for the service are therefore set during install/upgrade. As the user account must also exist on other servers, the user credentials must be entered for all server role types.

5.2.2.1. Account for the Confirmit Task System and Confirmit BitStream Services

The Confirmit Task System service is installed on combined or dedicated task system servers only, and is configured during installation. This account does not have to be part of the local Administrators group, (although it can be), the user requires at least the following permissions:

- Allow log on locally.
- Log on as batch job
- Log on as a service

After creating the user, it is recommended to log on to the server console with the account, starting any MS Office application (Excel, PowerPoint, or Word) to finalize the personalization for MS Office which requires user input. If this is not done, exports to Office formats may fail when run.

The Confirmit BitStream service, if selected during the installation process, is set up to run with the same credentials as the Confirmit Task System service.

Note: UAC must be turned off during installation. If the application must be run with UAC enabled, special procedures must be followed (see Confirmit and UAC (User Account Control) on page 41 for more information).

5.3. Installation

If all prerequisites are installed, backups are complete, Windows user accounts have been created and the initial Octopus configuration has been completed, the Confirmit upgrade / installation should be ready to commence. Once started, Octopus Deploy will look for existing installations, and if any are found, will proceed to upgrade the existing version. One or more of the steps described in the following sections might be required for the update to continue.

5.3.1. Install / Upgrade Sequence

Octopus deployment governs the installation sequence of Horizons components, beginning with the downloading of packages for the release, checking prerequisites are installed, updating SQL schemas (if required) and then proceeding to install specific packages related to the role of the site and each server.

Note: Changes to any database schemas are performed by the server assigned with the 'SiteBootstrapper' role in Octopus.

You can see the overview of the installation process either by selecting 'Process' in the release you intend to install or, if already installed, the previous task list of the installation.

5.3.2. Confirmit Deployment

Note: The first installation / upgrade of an environment using the Octopus deployment method must be performed with the supervision of the Confirmit Operations team. The following sections provide an overview of how the installation / upgrade is preformed and how you can deploy newer builds to your environment.

5.3.2.1. Deployment Process

Octopus deployment is the methodology you will use to install Confirmit onto the various machines in each environment. Refer to the Octopus section for an overview of the technology and process.

Deployment is the most common activity you will perform from the Octopus console.

Note: Deployment to environments must run in a specific order (as setup by your Confirmit technician). For example if you have Staging and Production environments, then your Octopus configuration will ensure that before a Confirmit version is deployed to your production environment it must first be deployed to your staging environment. This ordering of environments can be changed or removed, but this is an advanced topic.

5.3.2.2. Deploy Confirmit to the Staging Environment

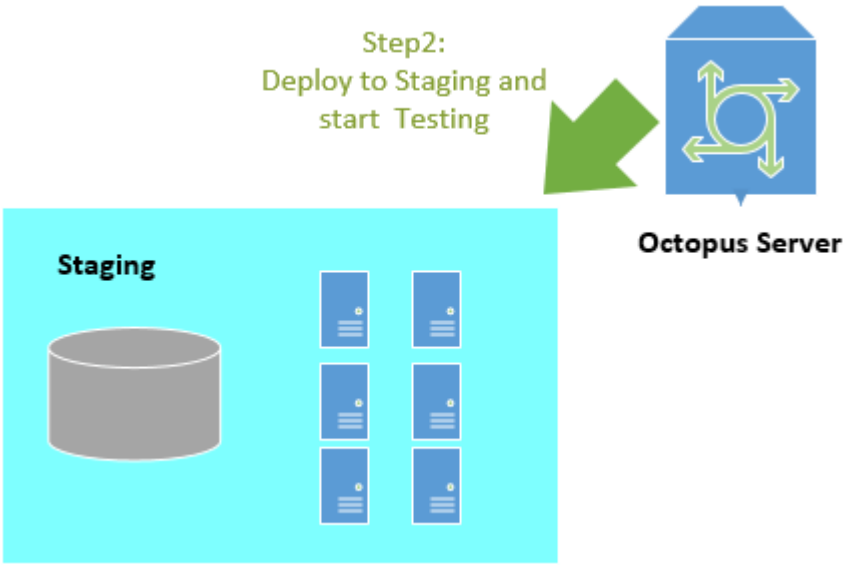


Figure 18 Deploying to the Staging environment

The aim here is to take a version of Confirmit that is available to the Octopus server, and then install it onto the Staging then Production Environments.

As stated in the previous section, the order of deployment to environments is strictly controlled by Octopus. New versions of Confirmit must be deployed on the Staging environment before they can be deployed to production.

5.3.2.2.1. Deploy to All Servers in an Environment

1. From the Octopus web interface - go to the top of the screen and click **Dashboard**.

Note: This is the default view when first logging into Octopus.

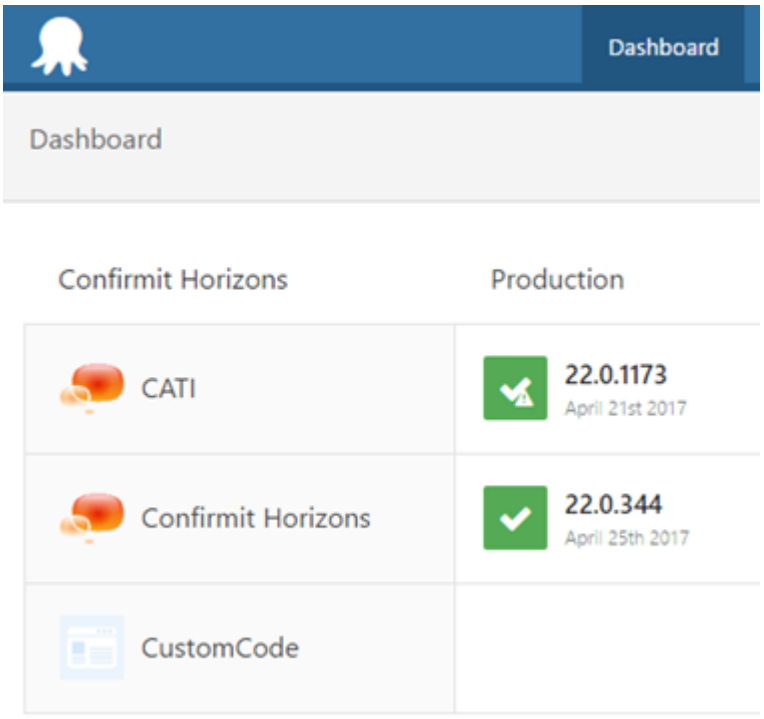


Figure 19 The Dashboard screen (this may differ from site to site)

2. Select **Confirmit Horizons** .
3. Select **Deploy** on the version you wish to install.

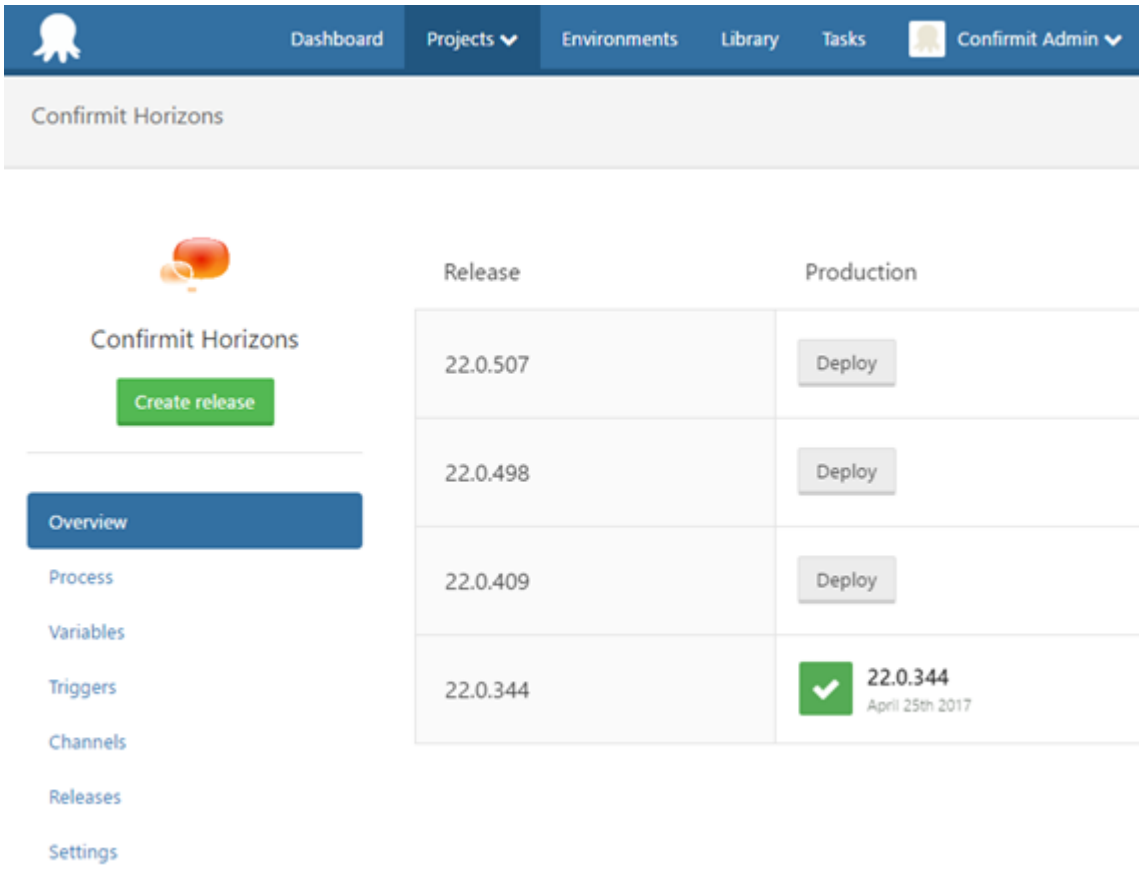


Figure 20 Click Deploy

The release deployment process window opens.

4. Verify you are deploying to the intended environment, scroll to the bottom and select **Deploy Release**.

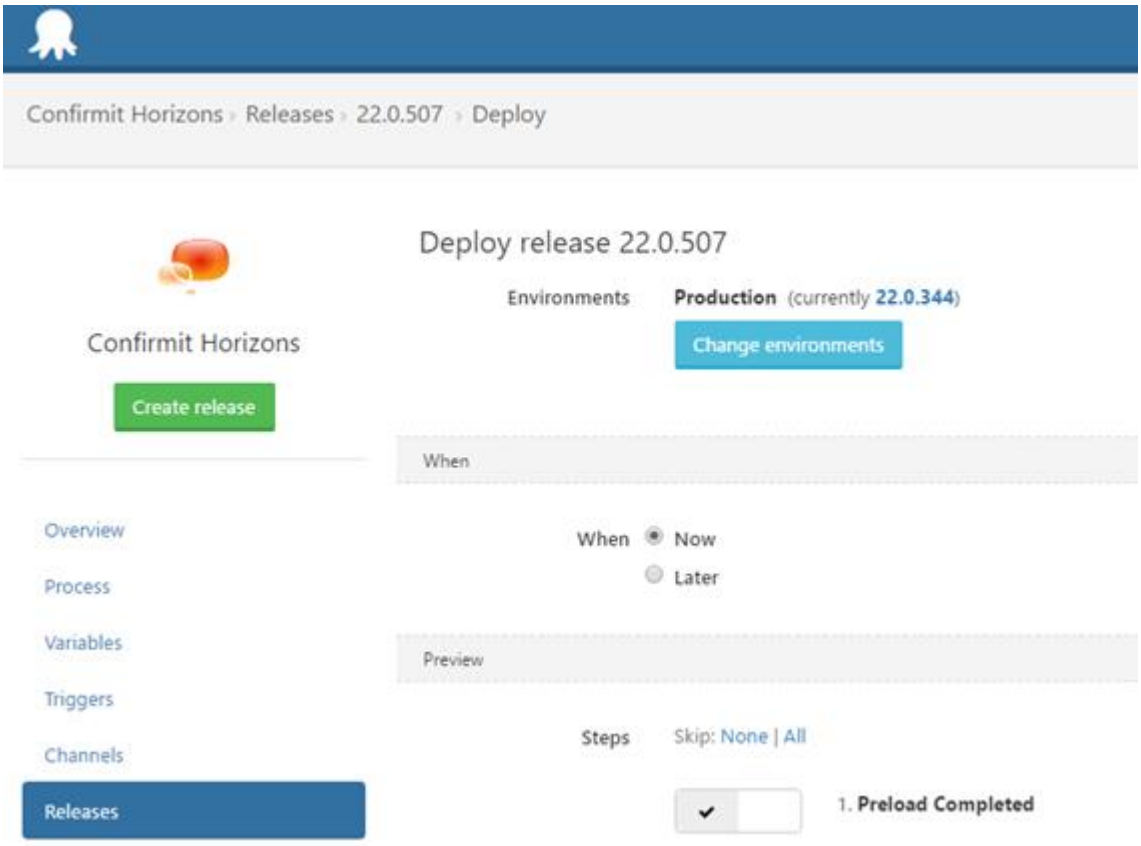


Figure 21 Deploying a release to the Production environment

This will begin preloading Confirmit nuget packages from the designated Confirmit repository.

5.3.2.2.2. Preload Step

Potential low bandwidth and latency issues between your site's Octopus server and Confirmit's Artifactory repositories may result in very long deployment times. The Confirmit release is approximately ~2GB, so downloading and then installing can impact installation times.

The Preload step (step 1) is a break-point in the Octopus rollout script; it stops the processing of the Octopus installation once the Confirmit packages have been downloaded. It then requires a manual confirmation to proceed. Once the pre-load step has completed, any subsequent installations of the release will make use of the locally cached files, rather than downloading the components again. Subsequent deployments of the same release to the same servers can be done by skipping this step.

After you have selected **Deploy**, your installation will stop at this point (after the Acquire Packages step).

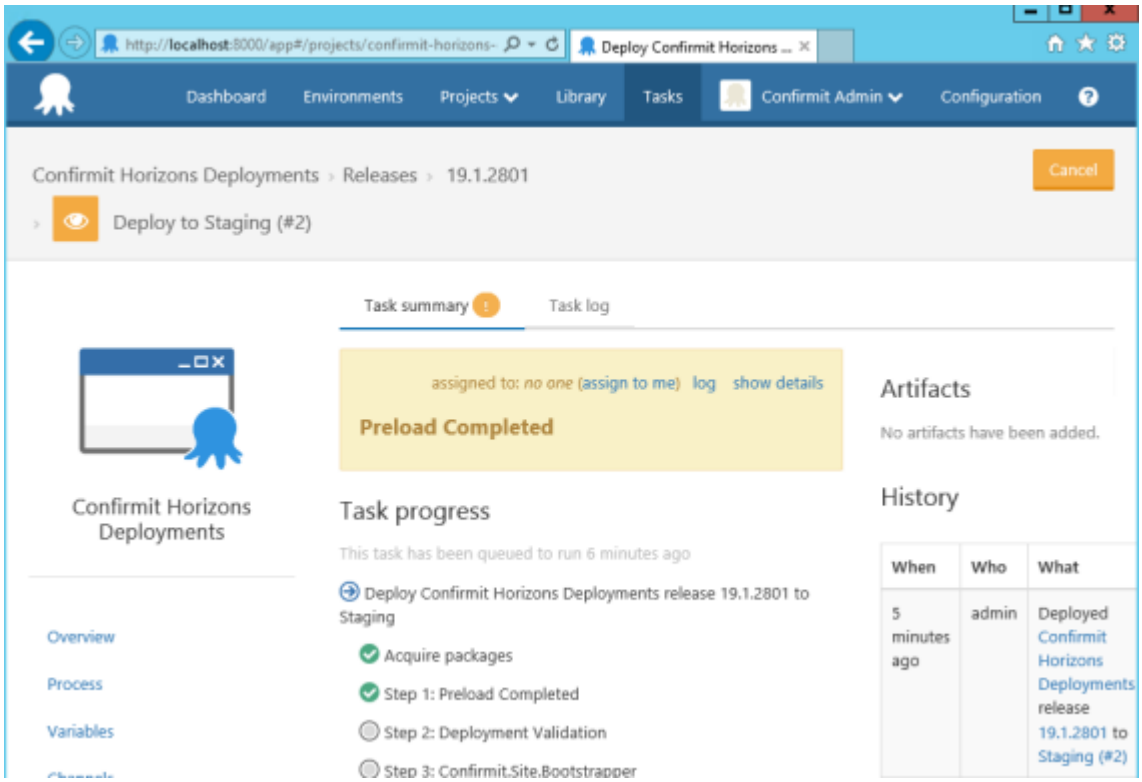


Figure 22 Preload complete

Click **Show details** to view the notification of 'Preload Completed'

To continue the installation click **Assign to me**. The following notification will appear:

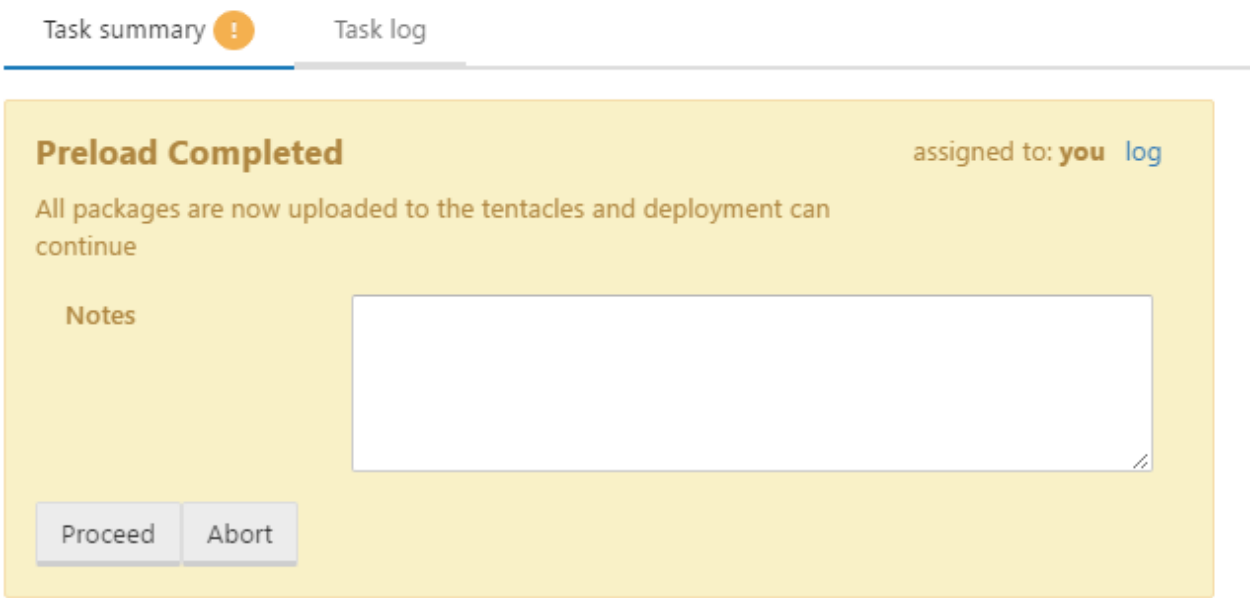


Figure 23 Preload completed

Click **Proceed** to continue with the installation.

Installation progress can be followed from here. The Task Summary tab displays the deployment steps. More detailed information can be obtained from the 'Task Log' tab.

5.4. Updating to a Newer Release

As part of the Octopus installation, a Windows Scheduled task is setup on the Octopus server that will regularly connect back to the Confirmit Artifactory server and pull any new releases of Confirmit. These new releases will be listed in the Octopus dashboard. The site can be upgraded by following the same installation steps as previously outlined.

Note: Be sure to check the release notes for any potential changes in new releases. These can be found on the Confirmit Extranet.

5.5. Confirmit and UAC (User Account Control)

5.5.1. When Installing Confirmit

Confirmit can not be installed properly while UAC is enabled. Follow these steps to disable UAC on your servers prior to installing Confirmit.

1. Run: **UserAccountControlSettings.exe**
2. Move the slider to **Never notify**.

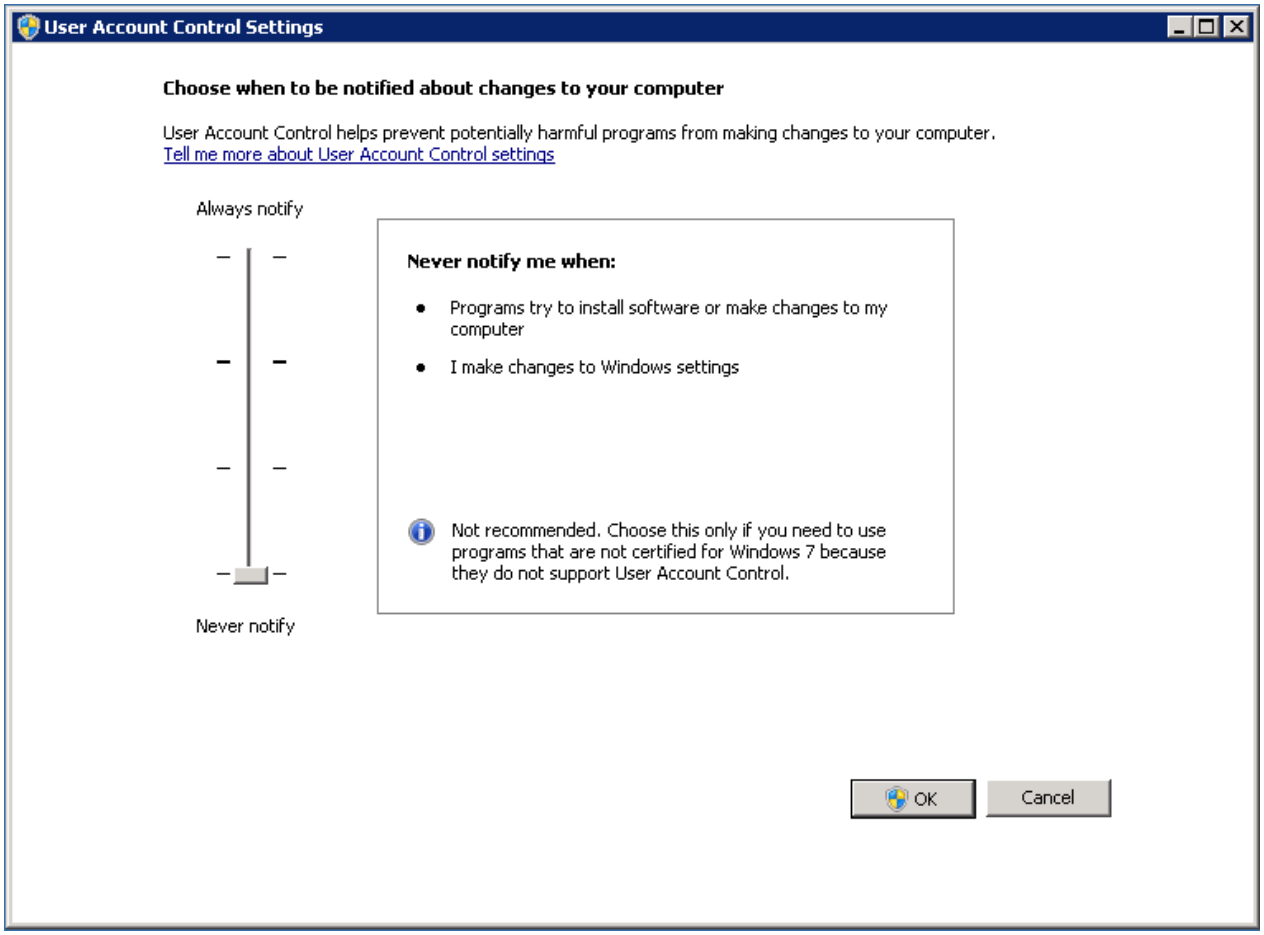


Figure 24 Disabling UAC

To enable UAC:

1. Run: **UserAccountControlSettings.exe**
2. Move the slider to the default setting (Default - Notify me only when programs try to make changes to my computer).

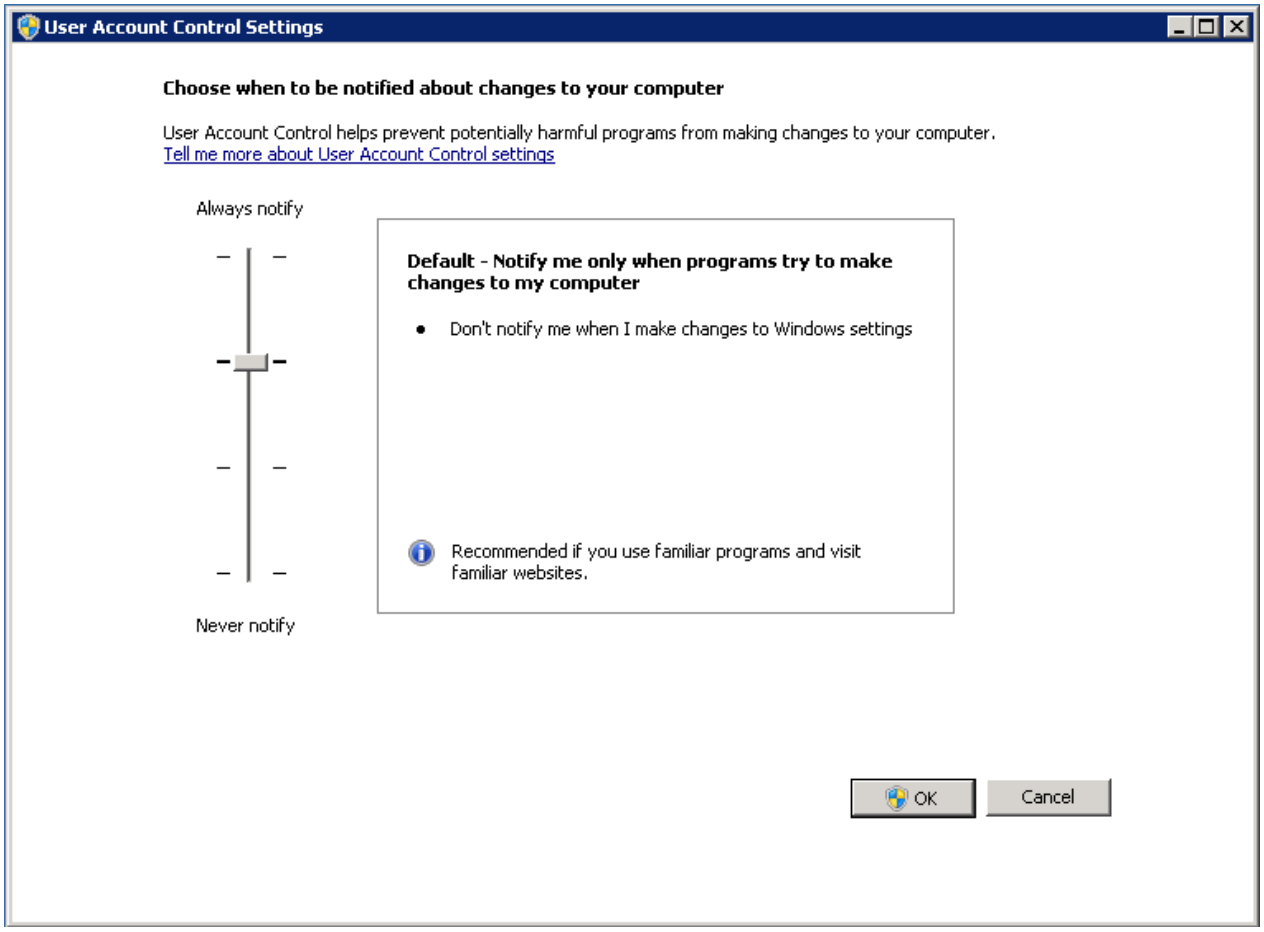


Figure 25 Enabling UAC

Note: Enabling and disabling UAC may require a reboot.

5.5.2. When Using Confirmit

If for security and policy requirements your site must have UAC enabled, you must perform the following steps to get Confirmit working:

ConfirmitDeploy cannot be member of local Administrators group, so

1. Add to **Remote Desktop User** so you can log on and perform required logins (For Batch servers).
2. Assign folder permissions.

   ConfirmitDeploy must have read/write permissions to Email server pickup directory (X:\Inetpub\mailroot\pickup).

   ConfirmitDeploy must have read/write/change permissions to Confirmit data directory (and sub directories).
3. Assign the same user name and password for ConfirmitDeploy on the other Confirmit servers in the site

This is also applicable if you have Shared data stored on a server that is not running Confirmit code.

Note: Confirmit will not work if you have UAC enabled and your ConfirmitDeploy user has Administrative rights. This is because when ConfirmitDeploy has Administrative rights, UAC will try to show the Raise Privileges overlay. This cannot be shown for a service account, and Confirmit will fail in performing the requested action.

5.6. Post-Installation Server Configuration

A number of procedures must be completed after the main installation phase to set the system up.

5.6.1. Obtaining and Installing a License File

The appropriate license files must be purchased and installed before Confirmit can be used.

5.6.1.1. Licensing Concepts

Confirmit requires a valid license file to function. The concept of license keys was introduced to protect the software against piracy. By default, Confirmit is not installed with a license.

Note that major changes to your SQL Server (such as migrating to a new server) might warrant having to issue a new license file for your site.

5.6.1.2. Add-ons

Confirmit functionality can be expanded with add-ons. Access to add-ons is controlled by the license file. Some add-ons are free of charge, while others are chargeable. Currently available add-ons include:

Add-on	**Chargeable**	**Requires 3rd-party software**
Spell Checker	Yes	No
Basic Panel	No	No
Translator	Yes	No
FTP for file transfer	Yes (included in Security Package)	Yes
PGP Encryption	Yes (included in Security Package)	Yes (included)
Reportal	Yes	Yes
CAPI	Yes	No
File Library	Yes	No
Reportal PDF Export	Yes	Yes
Kiosk	Yes	No
DomainKeys	Yes (included in Premium Emailing)	No

Dedicated IP	Yes (included in Premium Emailing)	No
Email Delivery Report	Yes (included in Premium Emailing)	No
Fixed Sender Domain	Yes (included in Premium Emailing)	No
Community Panels	Yes	No
Questionnaire Reviewer	Yes	No
ConcurrentSampling	Yes (included with Community Panels)	No
Community Snapshot Panel	Yes (included with Community Panels)	No
Data Processing	Yes	Yes
Sample Only	Yes	Yes
CATI Call Centers	Yes	Yes
Database Encryption	Yes	No
Flex Framework	Yes	No
SSO	Yes	Yes
Single Sign-On	Yes	Yes

Note: If you wish to remove the CATI Call Centers add-on, you must first remove all the call centers from the system manually.

5.6.1.3. Finding the Site ID for the License

Once the installation has completed, log on to Confirmit Authoring using the local machine: http://localhost/confirm.

1. Once logged on, go to **Admin > System Configuration** and open the License tab.
2. Beside Activation Info, click the **Copy to Clipboard** link.

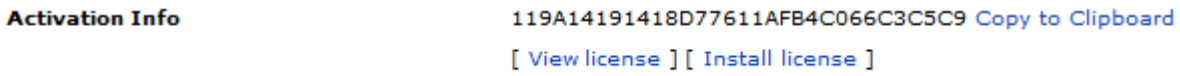


Figure 26 The Activation Info

3. Allow access to the Clipboard if prompted by the browser.

Save the information from the clipboard and send it to support@confirmit.com. Support will confirm that the request has been received and will issue a valid license file as soon as possible, and within two working days after receiving the request (if Confirmit is notified about the installation in advance, this step will typically be performed a lot faster). Once generated, the license file will be issued to the contact who initiated the request, provided this user is a designated support contact. Support will confirm delivery of the license file and follow up any requests for installation assistance if required. Note that the license file is valid for all servers within the same site, so it is not necessary to repeat the process for all servers.

5.6.1.4. Installing/Updating the Confirmit License

Note: The license file handling has been changed from v17.5 onward. Licenses must now be installed using the Confirmit Authoring GUI. Licenses can no longer be manually copied out to Confirmit servers

When the license file has been obtained from Support, log on to Confirmit Authoring:

1. Once logged on, go to **Admin > System Configuration** and open the License tab.
2. Click the **Install License** link.
3. Click **Browse** and locate the file obtained from Confirmit support.
4. Click the **Install license** button.
5. Examine the license overview and ensure it is correct, then click the **Close** button at the bottom of the page.

All issued license files have a specified expiry date. 30 days prior to the license expiry date, system administrators will receive a notification when logging on to Confirmit explaining that the license is about to expire. Confirmit maintains an updated list of license expiration dates, and will proactively provide new license files with extended expiry periods as long as the contract is not terminated. Renewed license files will be delivered to registered technical contacts no later than two weeks before the actual expiry date.

5.6.1.5. Confirmit SDK License

Customers using the Confirmit Web Services must install an additional license for the SDK if this has not been included in the regular license file. The procedure for installing the license file is the same, but the license is only required on servers where the Confirmit SDK is installed.

5.6.2. File System Permissions - Confirmit Data Share

Confirmit supports uploading from the user interface that requires files to be copied between different servers within the site, and also produces files itself that require deployment to remote servers (such as compiled survey packages, HTML reports, respondent files for importing, shared files in the File Library, etc.). In order to know which servers are supposed to receive specific types of files, Confirmit uses the role membership from the server setting (for example, production surveys go to servers with the Deployment role, while test surveys are deployed to servers with the Authoring role since that is where Authoring users are testing).

The share and related permissions are created automatically during installation in Octopus.

The Root Path setting in Confirmit should be changed to reflect the change. Either use the server NetBIOS name or the IP address, share name, and the trailing ‘web’ folder, i.e. \\servername\confirmitdatashare\web or \\192.168.0.10\data$\web.

5.6.3. Assigning Tasks to a Task System Server

When the Confirmit Task System service has been started on the task system server(s), the service will register itself in the Confirmit system database and allow tasks to be assigned to it. This can be done in the Confirmit GUI, in the **Admin > Batch > Server Config** menu, by selecting or deselecting a task type for the listed task system servers.

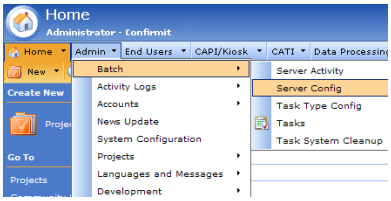


Figure 27 Going to the Admin > Batch > Server Config menu

For information about the task types in Confirmit, refer to the Confirmit Software Administrator Manual.

5.6.3.1. Task Isolation Level

Most tasks in Confirmit Horizons are written in C# and run in the context of the Confirmit Task System service (a few tasks still remain from older versions that are not converted to C#). All C# tasks can be configured to run in one of two isolation levels; Thread, or Process. The different settings can be configured in the **Admin > Batch > Task Type Config** area in Confirmit.

On this page, the various tasks can also be configured in a weighting system. Tasks are set up with a default configuration that has been tested to work in most configurations, with a suggested maximum number of concurrent tasks for that task type and the percentage-wise load a single instance of the task has on the server. If you experience task queuing on your system, especially for specific task types, then dedicating more servers to handling that task type or allowing more task instances to be run concurrently may be an option.

5.6.4. BitStream Server Operation Modes

The Confirmit BitStream service will be installed when the 'BitStream' role is selected. The BitStream service provides fast access to reporting data than SQL Server, and it provides a proprietary file format and query language developed by Confirmit. Confirmit BitStream Roles can be installed on one or more servers. Depending on requirements, this role is typically installed along beside the Task System role (Batch) or dedicated to its own server.

5.6.4.1. Managing BitStream Filesets

Bitstream files set are created on a 'per project' basis. All files will be created under the Confirmit Data directory / Bitstream. Based on 'Bitstreamweighting' file set will be split between servers with the Bitstream role when the first task runs to generate the file set. These can also be manually assigned or reassigned by going to **Admin > Projects > BitStream Server Configuration**, and tick the box corresponding to the chosen server:

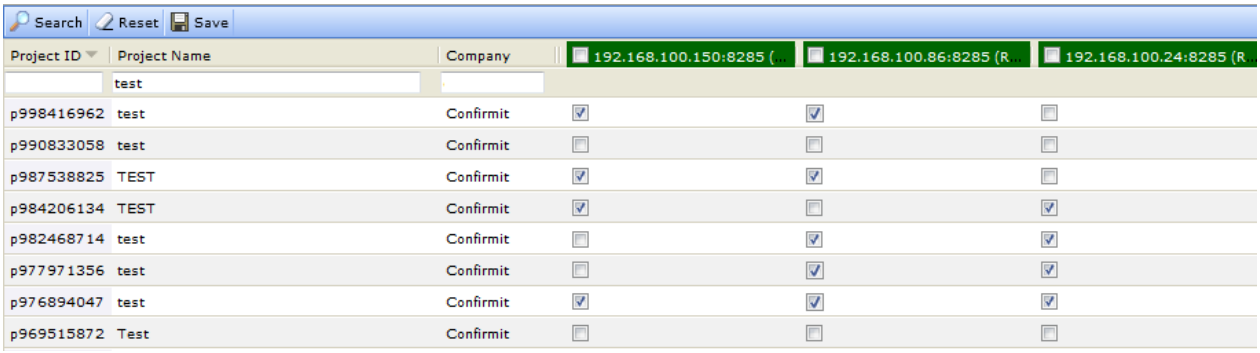


Figure 28 Project BitStream assignment across several servers

Note: If you have Reportal Reports with several Data Sources, and joins between these, you may receive a notice in Reportal about having to assign the BitStream file sets to the same BitStream server(s).

5.6.5. Synching the Confirmit File Library on a New Confirmit Authoring Server

Note: This only applies when introducing a new Authoring server, or exchanging the active Authoring server for a new one.

In order for new Authoring servers to be able to read file library data, the ContentUpload data store must be synchronized onto that server from the ShareData location.

Copy the <confirmit shared data path> \ContentUpload, including subfolders, to the <confirmitdata>\web\contentupload folder.

5.7. Upgrading From a Previous Version

The following tasks must be performed if you are upgrading your Confirmit installation from a previous version.

5.7.1. Assigning New Task Types to Task System Servers

New versions of Confirmit often include new task types, which will need to be assigned to one or more batch servers. This can be done in the Confirmit GUI, in the **Admin > Batch > Server Configuration** menu. If a task is not run on any of the servers, the text for the task will turn red.

5.8. Confirmit Configuration

The GUI-based Confirmit configuration can be accessed from the Confirmit Authoring module by a system administrator, and configuration settings are stored in an SQL database. When installing additional servers to a Confirmit environment, new servers will automatically register in the configuration system, with assigned roles based on the features that were selected during setup.

Important
There is one exception where servers will not automatically be added to the Confirmit configuration, specifically when setting up additional dedicated SQL Server instances or servers. Since there is no Confirmit code being run when installing additional SQL Server instances, adding database servers must be done manually (see Multiple SQL Server Instances on page 80 for more information). Contact support@confirmit.com if you need more information about this subject. For customers with the premium support package, configuration assistance for multiple SQL Server instances will normally be provided free of charge. Functionality to enable additional database servers from the GUI may be included in a future version of Confirmit.

Once the installation is completed for all servers, log on to Confirmit on an Authoring server as a user with the SYSTEM_ADMINISTRATE permission. The first logon should automatically redirect you to the system configuration page. If you need to access it later, it can be located in the **Admin** menu as shown below:

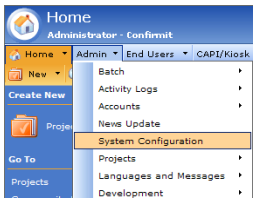


Figure 29 The Admin > System Configuration menu

5.8.1. Configuration System Concepts

Configuration settings have been arranged into categories that can apply to the whole site, for a group of servers assigned to a role, or per-server only for settings that can differ for each server. Descriptions for configuration settings are displayed when you allow the mouse pointer to hover over the setting name.

5.8.1.1. Site-Wide Settings

Site wide settings apply to all servers within the site. They can only be set on the site (top) level and cannot be overridden on a role or server level. The initial site wide settings are migrated from the first server to be installed or upgraded, and will not be overwritten by subsequent installs. Once set, new servers added to the site or upgraded from a Confirmit version with registry-based configuration will automatically inherit the site wide settings when added to the site.

5.8.1.2. Role-Wide Settings

Role-wide setting applies to the group of servers configured with that role. To assign a role to a server, double-click on the server in the **Servers** folder and select the checkboxes corresponding to the roles you want to assign:

Some role-wide settings apply to more than one role and will change for all roles if the setting is changed in one of the roles.

Note: Servers installed with specific features only cannot be adapted to running new features simply by assigning the role. This is because certain features must be installed during setup and cannot be added on the fly in the system configuration in the versions of Confirmit currently available. Selecting a role for a server will only enable configuration options for the server that are applicable to the role, and will make the server a deployment target for certain tasks (for example, selecting the 'deployment' role will make the server a target for launched production surveys but will not copy existing surveys on the system to the new server.) To change the role on a server, it is recommended to start by re-running the Confirmit Configuration Tool on the server, select the required role, apply the changes, then synchronize necessary data folders. Contact Support for details.

Role-wide settings can be overridden at the server level. When a role setting is overridden on a server, the setting value will be shown in a yellow background when viewing the server settings - see below. Also, a 'Revert to default' checkbox will be available (checking it and clicking **Save** will revert the setting to the value defined on the role level).

Figure 30 Role-wide settings

Figure 31 Role-wide settings overridden at the server level

5.8.1.3. Server-Wide Settings

Some settings (such as server name, UNC path for file deployment, etc.) are unique to each server and needs to be set on a per-server basis. These settings can be changed when a server is selected from the **Servers** folder.

5.9. The Command Line Management Tools

5.9.1. The ConfirmitAdministrator Tool

During the initial installation of the first Confirmit server within a site, a set of four SQL Server logins are created on the database server. Each of these four logins has permissions to the various databases that match their access requirements.

The SQL connection strings used by the ADO.NET component to connect different application features to the SQL Server databases are encrypted and stored in the configuration database. If the password for any of the Confirmit SQL logins is changed, the encrypted connection string must also be updated to reflect the setting. Since the encrypted connection string is unique to each server within the site, it would have to be changed on all servers where Confirmit is installed. This is the purpose of the **ConfirmitAdministrator.exe** tool. The tool also serves a second purpose – for updating the connection string used by the Confirmit Task System Helper service,

Run the tool from the command line to see a help screen explaining the various input parameters that can be passed into the tool.

```
ConfirmitAdministrator.exe /t:<type> [/r:<role>] /u:<user>
[/p:<password>] [/au:
<user> /ap:<password>]
```

5.9.2. The aspnet_setreg.exe Tool

.NET web applications spanning over several servers require that a logged on session is valid between the servers that run the application. To achieve this, the .NET Framework allows for two means of maintaining sessions. The first runs the ASP.NET State Service, installed with the .NET Framework, on one of the servers, and configures the .NET applications to connect to this service.

The second maintains sessions via a dedicated ASPState session database. If this is the preferred method, (it is how Confirmit applications are configured by default) each server serving the specific application must be able to read and write to this database. This is done using the last user account specified during Confirmit installation; the 'aspstate' account (if you used the suggested login name).

If the password for this SQL Server login is changed, the connection string used by the .NET applications will also have to be updated. This connection string is stored in an encrypted format in the system registry and must be changed using a tool normally installed with the .NET Framework SDK, **aspnet_setreg.exe**. The tool can be found in the **<confirmprog>\3rd party\Microsoft** folder.

The tool requires two parameters in order to insert the new encrypted password into registry:

- '-k:' = registry key location, relative to HKEY_LOCAL_MACHINE ('HKLM')
- '-c:' = the Sql Connection String.

The complete command string is as follows:

```
aspnet_setreg.exe -k:SOFTWARE\FIRM\ConFIRM\SessionState -c:"data
source=<sqlservername>;user
id=<ASPStateUserSQLlogin>;password=<ASPStateUserSQLpassword>"
```

The data source in the SQLConnectionString should match the name of the database (preferably the SQL Server Client Network Utility alias used during installation.)

If the suggested user name was used during installation, a typical syntax might look like this:

```
aspnet_setreg.exe -k:SOFTWARE\FIRM\ConFIRM\SessionState -c:"data
source=ConfirmitDB;user id=aspstate;password=MyPassword"
```

Note: If using the aspnet_setreg tool to update a changed password after the initial installation, permissions to the updated registry key are reset by the aspnet_setreg tool, and must be altered to allow Confirmit read access.

Note: The <confirmprog>\bin\SetRegPerm.exe tool will set the required permissions in registry automatically

Alternatively, start the registry editor on the server and browse to the 'SOFTWARE\FIRM\ConFIRM\SessionState' key. Right-click the key and choose 'Permission', and allow read access to this key for the Network Service system account. Also allow for permissions to be inherited by sub keys.

5.9.3. The ConfirmitAdmin Tool

ConfirmitAdmin.exe is a powerful tool that can be used for a variety of features on a server without having to log on to the Confirmit GUI, such as restoring soft-deleted projects, importing and exporting batches of survey definitions and many other practical operations. The various actions are documented in the tool itself; simply run the tool from the command line without any arguments to display a list of available commands. Confirmit Support can assist in the usage of the tool.

```
<confirmprog>\Confirmit.Admin.Tool\ConfirmitAdmin.exe /U:(username)
/P:(password) -command
```

5.9.4. Uninstalling Confirmit

To uninstall Confirmit from a server, run the setup tool and choose **Uninstall** to remove the server from the Confirmit configuration and remove the application files from the server. The Confirmit Data folder will not be deleted, but the CIFS share will be removed as part of the uninstall process.

6. Octopus Deploy

Octopus has been a great success in the SaaS Confirmit for some time where it has been used to support the Continuous Deployment model of software deployment - where code is checked in and subsequently pushed to internal test systems on a daily basis.

Although On-premise customers (in most cases) do not need the updates on the same frequency as the SaaS site, they can still benefit from the Continuous Deployment-style system which includes:

- Automation - Automated download of updates.
- Centralization - centralized management of all servers within an environment (staging/production).
- Simplified deployment - 1-click deployment of Confirmit to all servers in an environment (staging/production).

This has obvious benefits over the labour-intensive method that is required by the current MSI installer.

6.1. What is Octopus Deploy

Octopus Deploy is a third-party application, developed by a company with the same name. The application is used by Confirmit for orchestrating application package deployment. Where you in the past were required to log on to all servers and run a setup, Octopus Deploy will, via the use of agents running as services on all Confirmit Horizons application servers, ensure that deployment of new versions of Confirmit Horizons code is handled efficiently and securely.

6.2. Security in Octopus Deploy

In Octopus Deploy, as is the case with Confirmit Horizons, an application security review by an independent party is performed regularly, at a minimum on major releases. Major findings will be remediated as a part of this process. A report of this application test is available upon request. The Octopus Deploy server typically requires access to the package repositories hosted by Confirmit AS to download new versions of the Confirmit Horizons code. The access to these download sites are regulated via the use of Company-wide usernames and passwords. Outbound access from the server hosting the Octopus application is sufficient. Furthermore, the Octopus Deploy server needs to be able to reach all application servers running the Confirmit Horizons code to orchestrate the code deployment.

6.3. New Application Structure

Starting from v20, Confirmit Horizons will be made available via NuGet feeds, containing versioned NuGet packages, rather in one large MSI installer. The Octopus Server will have the information on which NuGet package versions need to be installed to make up a Confirmit Horizons release.

The underlying reason for this is that we want to be able to deploy new fixes and features more often, in a safe way. To achieve this, the Confirmit Horizons application has over a course of several years been broken into smaller, independent pieces, moving into a Microservice infrastructure.

A NuGet package is in effect a zip file. One NuGet package comprises everything necessary for running a Confirmit application.

6.4. Software Deployment Infrastructure

This is a connectivity diagram of the Octopus Server infrastructure.

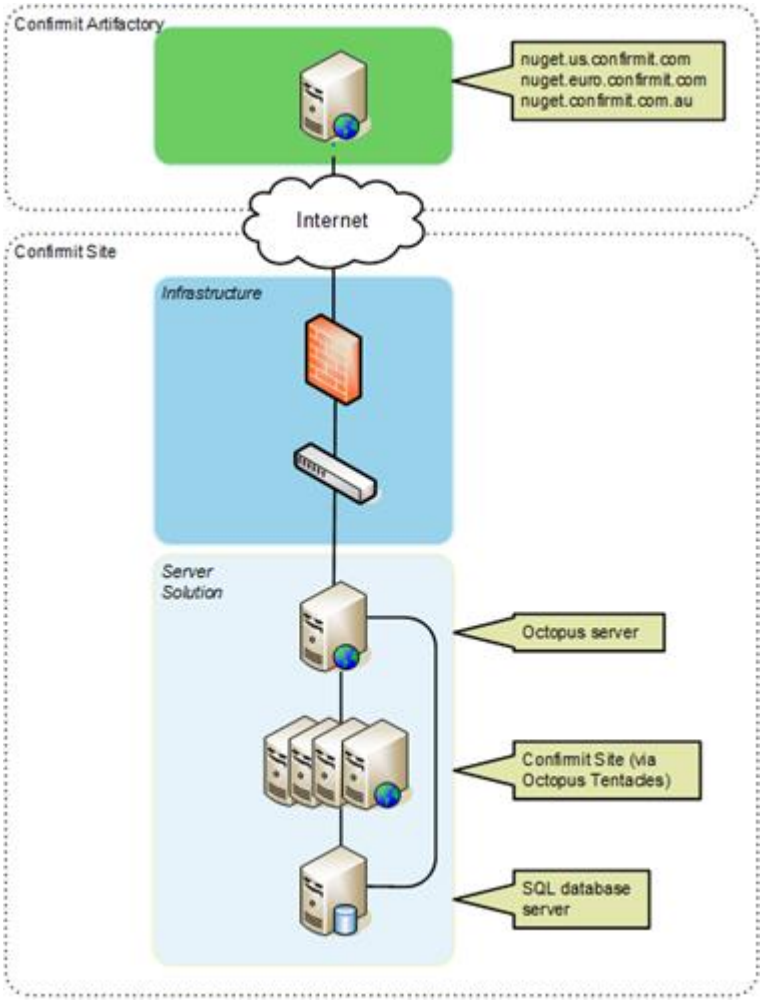


Figure 32 Diagram of Octopus infrastructure and required external connections

The Confirmit Artifactory section represents the Artifactory feed your Octopus server will connect to do download updated releases.

Note: If your Octopus server does not have direct access to Confirmit's Artifactory servers. Releases can be downloaded externally and copied to the Octopus server for installation.

6.5. Installation Workflow

This workflow assumes you have a two-site setup, that is Staging and Production sites, and that you will deploy Confirmit to your staging site first, where you do testing. Only after testing is complete will you deploy to production.

If you have a single site only, that is just production, then the methodology can be easily inferred.

Note: Setup of Octopus (including definition of environments and installation of agents) will already have been performed by Confirmit staff for you.

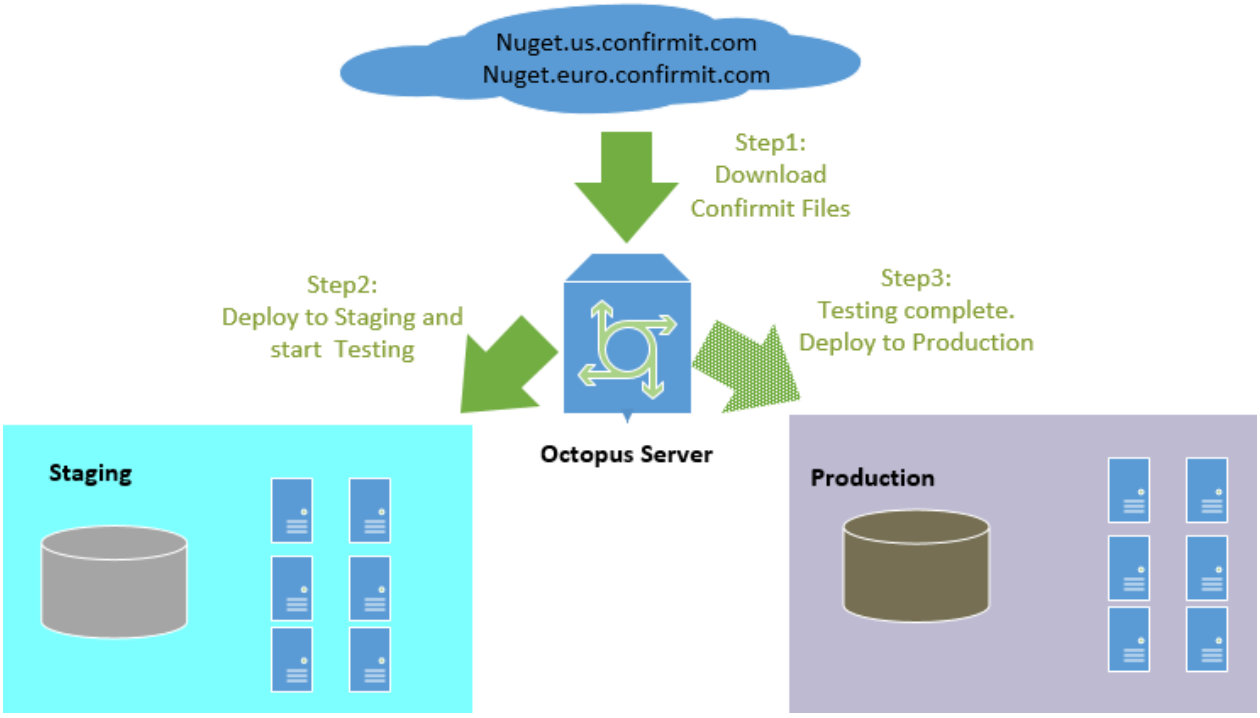


Figure 33 The installation workflow

6.5.1. Step 1 - Automated Release Updates

A scheduled task is pre-configured on the Octopus server that automatically contacts the Confirmit feed selected for your site (nuget.us.confirmit.com, nuget.euro.confirmit.com, nuget.confirmit.com.au) and will check for any Confirmit versions available for On Premise installation. These will appear in the release list in Octopus.

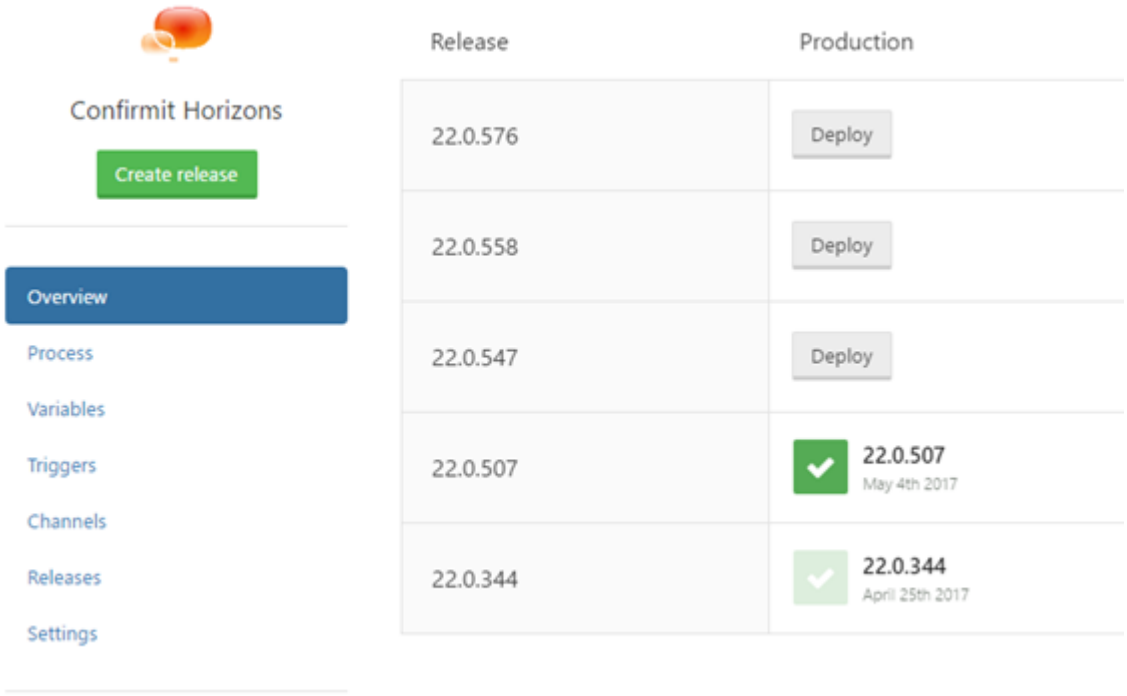
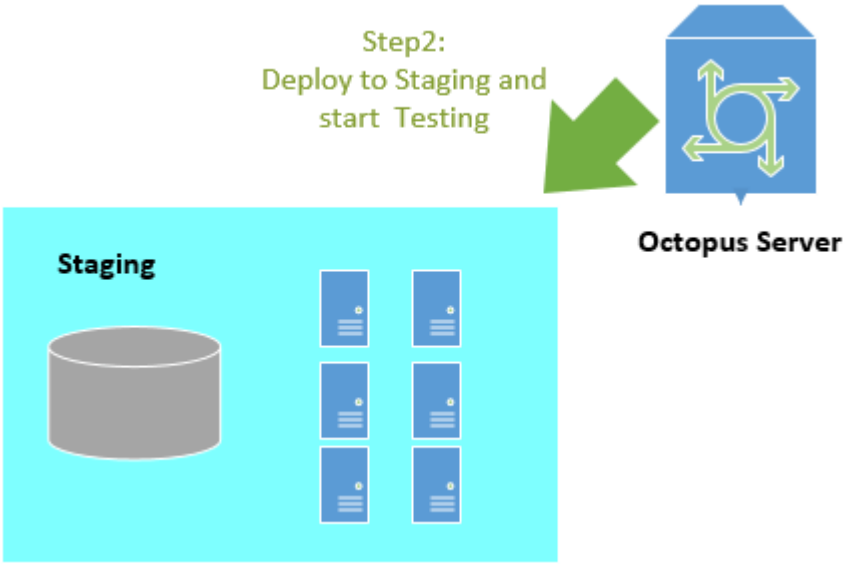


Figure 34 Available releases for installation to a production environment

6.5.2. Step 2 - Deployment to Staging

Decide which release you plan to install. Updated release notes are available via the extranet for each build.

In the Octopus web interface, choose that version and deploy it to your staging environment



Step 2 - Deployment to Staging

6.5.3. Step 3 - Testing

During this period you will test the software you deployed onto the Staging environment to ensure it is safe to be installed onto the Production environment.

Note: The scheduled task will run in the background, fetching the latest available versions of Confirmit. No downloading or installation to any of your environments will occur unless you choose to.

6.5.4. Step 4 - Deployment to Production

Once testing on the Staging environment has completed, deployment to production can be scheduled.

From the Octopus web console you can proceed to deploy selected version to your Production environment using the same method as staging.

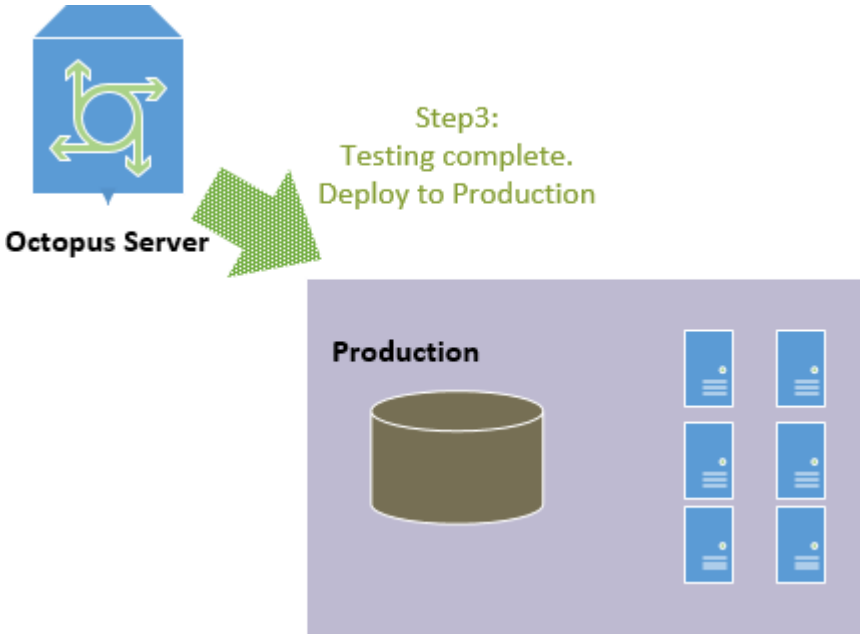


Figure 35 Step 4 - Deployment to Production

7. Confirmit Maintenance

This chapter contains information about recommended day-to-day maintenance routines, such as backup and logging, application monitoring, recommendations for server hardening and script examples for helping the system administrator set up scheduled maintenance jobs.

7.1. Backup and Recovery

7.1.1. System Files / Partitions

One advantage of having Confirmit installed on a dedicated partition/drive on the server is that required changes to the system such as a full reinstall do not affect Confirmit data. If C is your system drive, installing Confirmit application files in C and data files on a separate D-drive might be useful. This would allow you to set up the server exactly how you want it, and later using a partition imaging tool to back up the system drive completely. In the event of a system crash, the partition can be restored to a fully functioning state within a short timeframe. With data on a separate partition, the system partition size can be kept to a controllable level, and system restores would not require restoration of data.

7.1.2. Data Files Generated by Confirmit

The **<confirmdata>** folder contains files that have been generated and deployed to different servers by the Confirmit Task System service. The amount of data for each server will vary; servers belonging to the Deployment role will typically contain the largest amount of generated data.

Backing up the **<confirmdata>** folder is strongly recommended, and especially so for servers where only one server is a member of a specific role. If that particular server should crash completely, the data contained on it would be lost if no backups or additional servers containing the same data were present in the server park.

The Windows Server Backup tool included in Windows Server has shown itself to be useful when backing up the <confirmdata> folders. Details about that tool can be found here:

http://technet.microsoft.com/en-us/library/cc770757.aspx.

It works quite fast for large volumes of files and directories, creates a backup file in a general VHD format that is readable by many other backup tools, can be set up in a schedule and supports both differential and incremental backups. However, it does not support compression of backup data. Using WinZip (with the command line plug-in) or a similar compression tool allows for compression of the output Windows Server Backup file if this is required. WinZip can also be used for backing up the **<confirmdata>** folders directly, although it does not traverse directories as fast as Windows Server Backup and might even run into problems on very large systems with files in the millions (during initial traversing of the file system to determine the estimated time required for the operation..)

When backing up the **<confirmdata>** folders, **<confirmdata>\temp** can usually be omitted from the backup.

If a restore of any data from the **<confirmdata>** backup is required, NTBackup supports restore of single files or directories. If restored to their original location, Confirmit should become automatically aware of the data providing it was backed up from the same Confirmit version.

7.1.3. SQL Server Databases

Confirmit system and survey databases should be backed up on a regular basis to ensure the possibility of restoring the system or a survey to a previous point.

The recommended approach is using the SQL Server Agent to set up a scheduled backup job that runs online backup of all databases on the system without affecting availability of any databases on the system. The databases that should be included in a backup schedule are:

- **[ASPState] -** contains login states for authenticated users.
- **[Authentication] –** contains information about Confirmit logins.
- **[Confirm] –** – main system database. Contains task system data, Reportal reports, system configuration, survey information, end user system, etc.
- **[Confirm admin] -** contains information about survey databases on the instance.

- **[Confirmlog] –** contains company information and also logs from the survey monitor, Reportal, Authoring, and user activities.
- **[ConfirmitCache] –** contains Reportal data cache. Does not require regular backup but should be backed up at least once (for instance immediately after installation/upgrade/patch) in order to restore the schema if necessary.
- **[ConfirmitCATIV15] -** database master template for creating new Multimode instances.
- **[ConfirmitCATIV15 {N}] -** database holding Multimode instance data for companyid N.
- **[Confirmit.Reviewer.Service] -** data for CATI Reviewer micro service.
- **[Confirmit.RuntimeSurveyVariableCache.Service] –** data for RuntimeSurveyVariablesCache micro service.
- **[Confirmit Search] -** contains search queue information for Confirmit Searching Service.
- **[ConfirmitUrls] -** contains mappings between survey links and short URLs. This database may also be installed on SYSTEM instance.
- **[Custom_tables_design] –** contains hierarchy data in design mode.
- **[Custom_tables_runtime] –** contains hierarchy data in runtime mode.
- **[ConfirmitUrls] -** contains mappings between survey links and short URLs.
- **[DataCentral] –** contains temporary data for Data Central processing.
- **[Data_Integrations] -** database for Flex extension.
- **[Document Store] -** contains data used by the Rest (Metadata) API interface (primarily survey definitions, survey layouts, templates etc.).
- **[EmailOptOut] -** database for Flex extension.
- **[FileStorage] –** contains uploaded templates for Reportal PowerPoint exports.
- **[FraudDetection] -** database for Flex extension.
- **[Hierarchies[-** contains data for the Hierarchy Management module.
- **[Hub] –** contains hub definitions used for creating hub_xxx_default and hubs_companyx_prod_x databases. These databases can be recreated as long as the “Hub” database exists, so it’s really important that this is backed up.
- **[hub_{n}_default] –** database for SmartHub. N is hub number.
- **[hub_admin] –** common hub resources for a Report SQL Server instance. Mainly used for keeping record of detached databases.
- **[hubs_comany{n}_{variant}_{x}] –** databases for implicit hubs within a company, created for survey key metrics or from Create Reporting data in Authoring UI. N is company ID, x is a global counter, variant is prod for production data, test for test data and keymetrics for survey key metrics.
- **[hubs_companyresources_{n}] –** calendars for each company. n is a global counter. You will find company number as part of the schemas in the database.
- **[hubs_dbschemas_{n}] –** database designer objects. N is a global counter. You will find database designer schema ID as part of the schemas in the database.
- **[Identity] –** contains data for the Confirmit Identity microservice.
- **[ImageMap] -** database for Flex extension.
- **[Individualreports] –** contains individual report information (hit lists, single views, etc.).
- **[MobilePortal] -** database for Flex extension.
- **[Node Store] -** temporary data storage used by the Rest (Metadata) API. Contains unzipped data from the DocumentStore database.
- **[PanelManagement] –** contains community panel data.

- **[Resources] –** contains languages and system messages.
- **[SmsSurveys] -** database for Flex extension.
- **[Survey_p{N}] –** contains respondent and response data for a survey compiled in production mode (legacy or optimized format).
- **[Survey_p{N}_test] –** contains respondent and response data for a survey compiled in test mode (legacy or optimized format).
- **[Surveys_{companyid}_dp] –** contains required Reportal metadata for reporting on non-Confirmit data.
- **[Surveys_{companyid}_express] –** contains Confirmit Express surveys for a specific company (optimized format).
- **[Surveys_{companyid}_poll] –** contains Poll surveys for a specific company (optimized format).
- **[Surveys_{companyid}] –** contains optimized format surveys for a specific company where the 'reusable databases' option has been enabled.
- **[TextHighlight] -** database for Flex extension.
- **[VideoQuestion] -** database for Flex extension.
- **[WITemplates] –** contains web interview templates.

Note: Some of these databases are role-specific, meaning that they might not be present on your system if the related role isn't installed.

7.2. Monitoring and Systems Maintenance

7.2.1. SQL Database Server Maintenance

To maintain a healthy and well-performing environment, the database server should run a number of scheduled maintenance jobs. Some jobs are included during installation as stored procedures in Confirmit databases, but are not started by default, and others can be added to fit individual setups.

Note: Confirmit does not by default hard-delete deleted projects. You must start and schedule the Confirmit Database Cleanup task in the Authoring interface under Admin > Maintenance > Database Cleanup to permanently delete projects from SQL.

1. To adjust the number of days that deleted projects are to be kept as soft-deleted, tune the DaysToKeepSurveyDatabasesForDeletedProjects in the sitewide System Configuration settings.
2. Go to **Admin > Maintenance > Database Cleanup** and change from "ASAP" to "Schedule for later execution", then click **OK** and create an appropriate recurring schedule.

   Keep in mind that this task does put stress on the system, so it is highly recommended to run it out of high peak hours.

7.2.1.1. Restoring Survey Databases

If a Confirmit database is restored from backup, service broker must be set manually for the database so that Confirmit will function correctly. Run the following command after restoring a survey database to enable the broker on the database:

```
ALTER DATABASE <database name> SET NEW_BROKER WITH ROLLBACK IMMEDIATE (or
use 'ROLLBACK AFTER XX' seconds instead of 'ROLLBACK IMMEDIATE').
```

Note: Ensure that the ConfirmitDeployer SQL account is the database owner after restore. The current user will always take ownership when restoring databases.
The following SQL command will allow you to change database owner:
(use <database name>; exec sp_changedbowner ConfirmitDeployer;)

7.2.1.2. Automatic Detachment of Inactive Survey Databases

The traditional Confirmit database model operates with a separate database for each survey created on a site (two if a survey is compiled in test mode.) Although this provides a fully segregated database model and allows for individual maintenance of particular surveys, it would eventually become a problem if the server was loaded with more databases than it could effectively operate with in memory, potentially leading to an unstable server, and in the worst case, a server crash rendering all databases unavailable. This was not likely to occur until the number of databases reached over 1,000 unless running on database server with very little memory installed, but still something had to be done for the future of growing installations.

One of the issues that can be seen if the amount of databases on the system starts reaching the limit is errors when running certain tasks on the system, typically yielding errors like "There is insufficient memory to run this query". This warning indicates that the SQL Server's MemToLeave area is too small for the current amount of databases and transactions.

Microsoft SQL Server allocates at least one 64KB block used for formatting log records before they are flushed to disk. The allocation takes place when the first log record is generated for a database, following any INSERT, UPDATE or DELETE statement. Depending on the size of the generated log records and the activity on the system, additional 64KB blocks may, and will in most cases be allocated as well.

The situation can be somewhat relieved by configuring the SQL server to use the -g startup flag to instruct the server to leave more memory aside to prevent that the combination of log-related allocations and other allocations causes the SQL server to run out of virtual address space. The more databases are attached and active and on the system, the more memory would likely have to be reserved.

The Microsoft SQL Server Books online provides more documentation on this feature, and http://support.microsoft.com/kb/316749 references it in more detail as well.

Confirmit can detach survey databases that have not been accessed for a while. The [confirm_admin]..[surveydatabases] table stores and maintains information about the current status of every survey database on the system, as long as the project has not been deleted from the GUI. The table is initially populated during the installation of Confirmit, and is automatically updated with new rows when databases are compiled for new projects.

Included as a stored procedure in the [confirm_admin] database is usp_DeactivateIdleSurveyDatabases. The procedure takes one parameter; an integer specifying (in minutes) how long a database should be inactive before it's detached by the procedure, with 1440 (24hrs) as the default set up in a maintenance job created on the database server during Confirmit installation; "Detach idle surveydatabases". The job is always installed but not enabled by default, and must be enabled manually to run. It is recommended that it is enabled for most environments.

When the procedure runs, it will read through the [surveydatabases] table and store the LastConnect timestamp (which is updated dynamically by any Confirmit application module which uses the database in any way,) and detach any survey databases that exceed the set limit, thus freeing resources for the SQL Server process. It will also set the status of the database as being currently detached (inactive.)

If any part of Confirmit (a survey, report user, etc.) tries to access the database while it's detached, Confirmit will automatically reattach the database (provided the files are still available) and reset the status to attached (active.)

Important
The SQL Agent will reset NTFS permissions on the databases when you detach them. Therefore the SQL Server Agent MUST run under the same user account as the SQL Server, otherwise your SQL Server will not be able to attach the databases.

If for any reason there are databases that should not be detached from the system, the [surveydatabases] table can be modified, altering the donttouch column for the relevant row for a specific database. Setting the value of this column to '1' for required databases will make the detach procedure to overlook them in the detachment process.

Note: only Confirmit survey databases are affected by this procedure, system databases are never, and should never be detached.

Important
To prevent excessive reads to the control table to determine survey database status, each server caches a survey database's status for 60 minutes after a successful status check. Keep this in mind if detaching databases manually with a shorter time frame than 60 minutes, and recycle IIS on web servers and restart the Confirmit Task System on batch servers if there are inconsistencies – or simply wait for the issue to resolve itself within an hour.

7.2.1.3. Deleting Survey Test Databases

Administrators might also notice that survey test databases expand over time, and after a while can take up quite a lot of space on the server. To keep the number of test databases to a minimum, a maintenance job can be added on the database server to delete databases that have been inactive for a specified period. An example script to add a procedure for automating this job is available in the Server Admin Scripts section on the Confirmit Extranet: http://extranet.confirmit.com/download/AdminScripts.aspx. Note that you will need to login to the site. If you don't currently have the required access permission then there is a link on the login page that will open an email to the Confirmit support team so you can request the permission.

7.2.1.4. Clearing Contents of Log Tables

The databases [confirm] and [confirmlog] contain task history and survey response history tables respectively. On a busy system, these databases will quickly grow substantially in size, and over time performance might deteriorate when reading from or writing to these databases.

To control the growth of the databases, some tables can be managed on a scheduled basis to reduce the number of stored rows. Scripts are available in the Server Admin Scripts section on the Confirmit Extranet: http://extranet.confirmit.com/download/AdminScripts.aspx. Note that you will need to log in to the site. If you don't currently have the required access permission then there is a link on the login page that will open an email to the Confirmit support team so you can request the permission.

7.2.1.5. Defragmenting Indexes

Indexes on large tables may become large after a while, and if the tables are cleared regularly, indexes can contain chunks of white space that could be optimized by defragmenting the index. This can both lead to less disk space being used by the index, and increased performance due to reduced logical reads. In our SaaS environments, we use Ola Hallengren's scripts for Index and Statistics maintenance. We highly recommend those to our clients as well. The scripts can be found here:

https://ola.hallengren.com/sql-server-index-and-statistics-maintenance.html

7.2.1.6. Shrinking Databases

In most cases we do not recommend shrinking databases as this often just leads to more autogrow events and reduced performance when the database has to grow back up later. However, sometimes it makes sense to do it, particularly on databases that are huge but have lots of free space inside them. Typically, this can happen if you haven't performed any cleaning in the confirmlog database previously and have just started doing it, or have been using DocumentStore/NodeStore for a year without enabling the DocumentStore/NodeStore cleanup tasks. In cases like this, you can consider shrinking those databases. Note that the shrink itself can cause performance problems in the database when it's executed, so off-peak hours is the best time for doing it.

7.2.1.6.1. SQL Server LOB Compaction

SQL Server offers functionality to compact pages that are sparsely populated. Run the following statement to de-allocate unused pages:

```
use [survey_pXXXXXXXX]
go
DBCC SHRINKFILE (1, 1)
WITH (lob compaction=ON)
```

Note that we do not recommend shrinking unless you:

- Are certain the database isn't going to grow back up again.
- Are in a situation where you don't have much choice.

7.2.1.7. Design Log Cleanup

This task does not have any parameters, but uses the DesignerLogCleanupPeriod setting in System Configuration. This is set by default to 180 days, so Design log entries older than 180 days will be deleted.

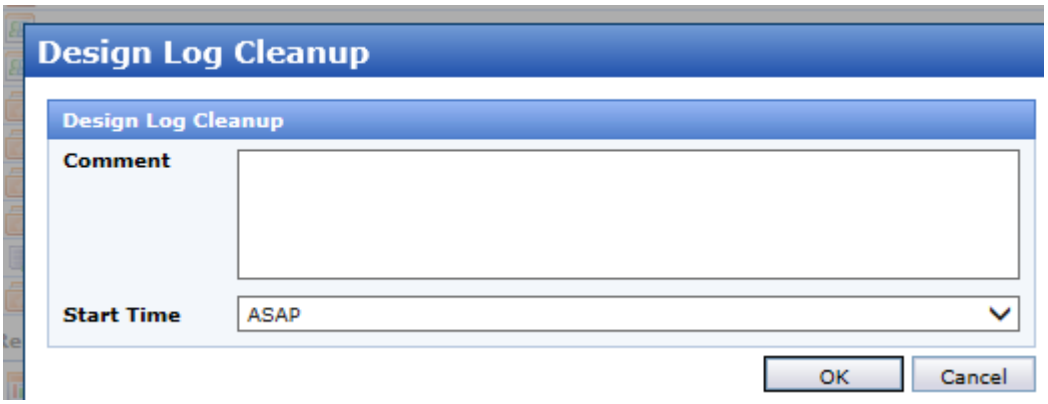


Figure 36 The Design Log Cleanup overlay

Design log entries for deleted projects will not be removed with the DB cleanup task, but wait for the Design Log cleanup to do its work.

7.2.1.8. DocumentStore Cleanup Task

The DocumentStore cleanup task will delete all document versions that are marked as expired. Version expiration is set based on the system configuration setting "MaxSurveyVersions". The latest x (MaxSurveyVersions) will be kept; the rest will be marked as expired (and thus cleaned whenever the DocumentStore cleanup task runs).

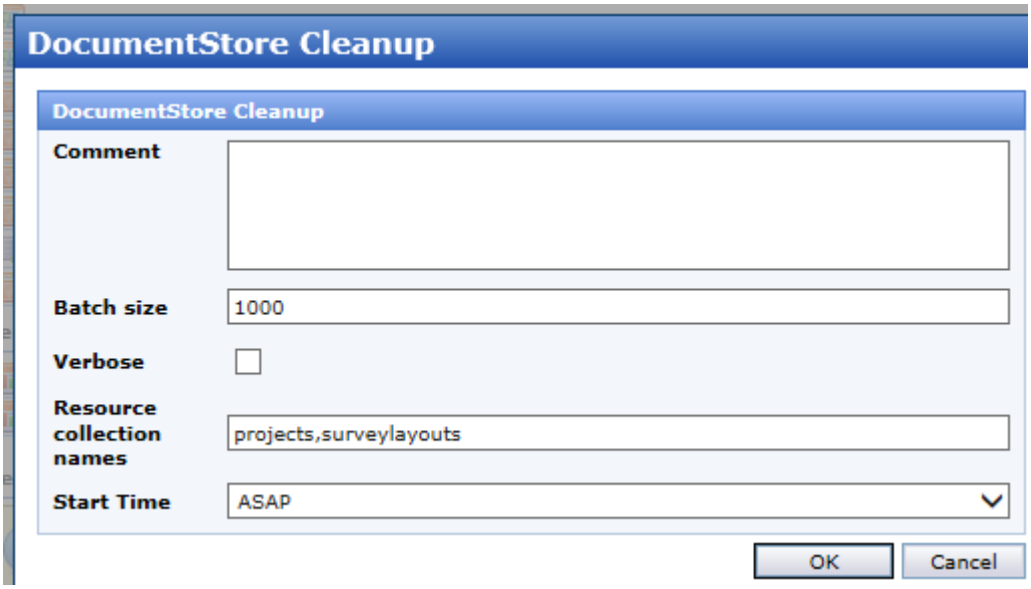


Figure 37 The DocumentStore Cleanup overlay

- **Batch size -** defines how many versions will be deleted every time the task runs. So if you have 50 000 expired versions and the batch size is 1000, the cleanup task will delete 1000 versions and then stop.
- **Verbose** - check to get details of individual records that have been deleted. For example:

  If Verbose is not activated -

  - 1000 records deleted
  - 2000 records deleted

- 3000...

If Verbose is activated -

- p1347856284 v13 has been deleted
- p1237495869 v7 has been deleted
- etc.

Note that Verbose is useful for trouble-shooting, but can quickly lead to very large task logs.

- **Resource collection names -** defines what will be deleted. I.e. Survey layouts and/or projects.
- **Start Time -** defines when the task is to run. Select Schedule for later execution to set up a recurring task.

7.2.1.9. NodeStore Cleanup Task

The NodeStorecleanup task will remove all data for the configured “resource collection names”, that is older than the “expiry days” setting configured in the NodeStore cleanup task definition.

Note: All documents exist in the DocumentStore. No data is lost when this cleanup job executes.

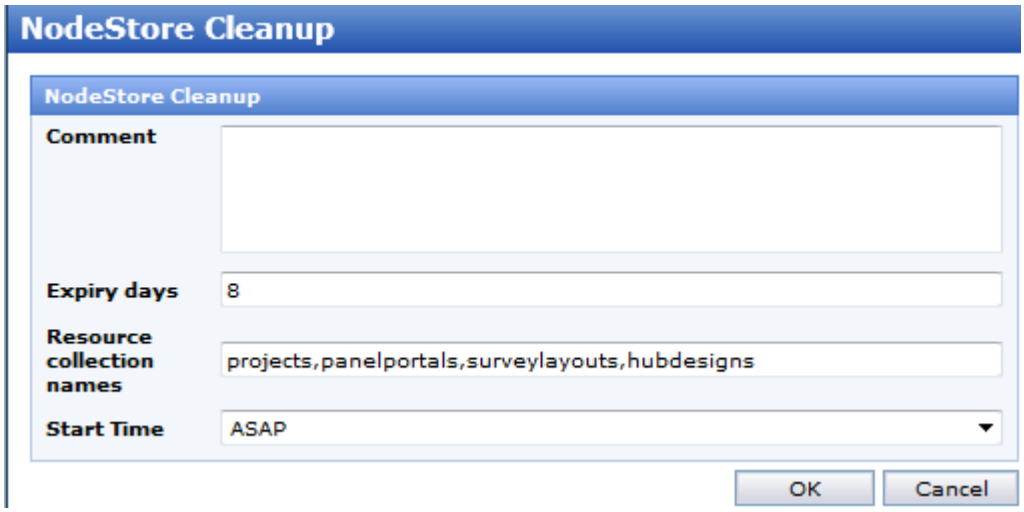


Figure 38 The NodeStore Cleanup overlay

- **Expiry days -** defines the number of days NodeStore entries are to be kept (days to hold). Once the task is set to recurring, any changes to a property will be included for the next task run. The recommended expiry days setting for this task is 8 days.
- **Resource collection names -** defines what will be deleted. Ie. Survey layouts and/or projects.
- **Start Time -** defines when the task is to run. Select Schedule for later execution to set up a recurring task.

The following properties are located in System Configuration. Under "Searching":

- **NodeStorageIndexingEnabled** - If the indexing of changes in the node store is enabled - this is related to the index used by the Confirmit Searching service (not related to the NodeStore cleanup task).

And under "Miscellaneous":

- **NodeStoreExpiry -** This is the number of days before versions will expire from NodeStore. Note that the versions will still be available and compressed in DocumentStore after being removed from NodeStore.

7.2.1.10. Searching and Indexing History Cleanup

Figure 39 The Searching and Indexing History Cleanup overlay

The ...history cleanup task removes old entries in the ConfirmitSearch database. The Days to expire... value specifies the number of days the history is to be kept for each task before it is deleted.

The task parameters specify how long internal activity tracking data is kept. This data is really used for extended troubleshooting, so unless you are in the process of troubleshooting something related to the Confirmit Searching service, the default values can be used when scheduling this task.

7.2.1.11. Hub Cleaning

Hub Cleanup can be performed periodically to permanently remove deleted hubs, sources and related objects from SmartHub. When a hub is deleted from the user interface, it will initially be “soft-deleted” (marked for deletion). The database will not be removed until the hub cleanup task runs after a specified number of days have passed.

There are two configuration settings in “System Configuration” for this:

- DaysToKeepHubDatabaseObjectsAfterSoftDeletion controls the number of days that must expire before the database holding data in the hub column store can be dropped. Note that as the definition of the hub is not deleted it will be possible to restore a deleted hub after the database is deleted. The hub loader will be able to recreate the database in the hub column store from the original sources (unless they have been removed).
- The definition of the hub will be deleted when more than the number of days specified in the DaysToKeepHubDesignObjectsAfterSoftDeletion config setting has passed. This can be set higher than the DaysToKeepHubDatabaseObjectsAfterSoftDeletion, since the definition is small compared to the actual data.

7.2.1.12. Single Page Survey Cleanup

This task cleans up any files created by the single page survey generation task where the files are older than 1 month.

7.2.2. Anti-Virus Recommendations

Servers with file deployment and/or upload (task system and author servers) should have anti-virus software installed to protect the system from infected content. There are two important considerations with regards to performance of a Confirmit server when installing an anti-virus application:

- Real-time file scanning of survey packages (read/write/access) might slow down interviewing, file deployment (WI and report generation) even if **<confirmdata>** folders are added to the exclusion list (files still need to be checked against the exclusion list.) In some cases, anti-virus software might even cause unexpected errors in Confirmit (particularly in task system logs, referencing file deployment.)
- Script blocking might render some parts of the application unusable as this might refuse some of the COM objects to run (i.e. components accessing the file system on behalf of the web server.) If possible, script blocking should be disabled or not installed at all.
- For best performance and to reduce the possibility of corrupted operating system files, Confirmit recommends you adhere to the official Microsoft Anti-Virus Exclusion list: https://social.technet.microsoft.com/wiki/contents/articles/953.microsoft-anti-virus-exclusion-list.aspx

7.2.3. Time Synchronization

Over time, internal server clocks may drift at slightly different speeds. Running a multi-server installation may eventually lead to servers being out of sync. This can usually be diagnosed by delays in task execution, or in variations of recorded time-stamps from surveys. This can be avoided by updating the server clocks regularly, either by using an Internet time server, a domain server or synchronizing manually.

7.2.4. Horizons Logging Format

Starting with features introduced to General Availability in Confirmit Horizons v22 Continuous Deployment version, some application and service logs are stored in a new location and format. Where logs traditionally were stored in SQL, they are now logged directly to disk using a built-in logging framework. The log directory for these logs is determined by the **Confirmit.Site.LogPath** variable in Octopus. As is the case with IISlogs, there may be reasons why logs need to be kept in an unchanged format for an extended time, so these logs are not cleaned up by the Horizons application. We strongly recommend that you determine your requirements for keeping logs for audit purposes, and clean them as necessary.

Some background for the change: The advantages of changing to this format are that it significantly reduces the cost of introducing rich logging to new features (thereby reducing the time for the feature to become available), it reduces strain on the back-end SQL clusters, and it reduces strain on SQL when viewing logfiles.

The structure of the logging directory will be **<Confirmit.Site.LogPath>\services\<service name>\<log>.json**. The **<Confirmit.Site.LogPath>** path is a site variable defined in Octopus.

Note: For troubleshooting purposes, Confirmit Support are able to parse logs created by your installation. This will require you making them available to Confirmit Support. Compressing and encrypting logs first is recommended.

The Confirmit Horizons logging syntax uses standardized JSON format that can be read by various log processing products such as Splunk, Graylog, Sumo Logic or ELK. Should you have a logging system already available for other monitoring purposes, this can be used for Horizons logs.

7.2.4.1. Log Processing

For a complete logging solution, a system should be used for picking up, processing and centralizing the produced logs automatically. As mentioned in the previous section the generated log files can be processed using various products, however Confirmit has experience with and is using the ELK stack for our own test and SaaS environments. NXLog (https://nxlog.co/, an agent running on each Horizons server) monitors the log directories and writes the log-entries to Logstash for indexing in an Elasticsearch cluster. Kibana is then used for visualization of the data. Elasticsearch, Logstash and Kibana is part of an open source software stack (ELK) which can be downloaded and installed free of charge.

Note: Confirmit are working on an easier distribution model for downloading, setting up the cluster and applying the appropriate configuration via Octopus deploy. If you require help setting up a solution based on ELK before the distribution model has matured, please start by contacting Confirmit Support.

To set up an Elasticsearch cluster manually, follow the steps here: https://www.elastic.co/guide/en/elasticsearch/reference/current/windows.html. Please refer to documentation by Elastic for assistance with the ELK components. Confirmit Support will be able to assist you with providing appropriate configuration to digest the logs created by Confirmit Horizons.

Note: For larger installations we recommend getting a support agreement in place with Elastic, or a company providing such service.

7.2.4.2. Log Types and Formats

Service log file. Expected file name: service-<date>.json

This log type is used for service-related information, such as providing metrics for when a task has been processed, containing the task type, task duration, task delay before being picked up etc.

Json FieldName (* = mandatory)	Format	Description
CorrelationId	string	Unique request identifier. Identifier of the client request used to relate log records from different sources and infer causality. Passes from the client in X-Confirmit-Correlation-Id header. If header is empty service should generate an globally unique id.
Service	string	Name of the application / service that is doing the logging. The name format is: Confirmit.X.Y. (Example: Confirmit.Dictionary.Api, Confirmit.SurveyDesigner.Client).
ReferrerService	string	Name of the client application / service. (The service calling this service). Generated by client and passed in X-Confirmit-User-Agent.
InitiatingService	string	Name of the service that initiated the request. Typically the first service will set it's own service name to InitiatingService. Passes from the client in X-Confirmit-Initiating-Service header.
EventTime*	ISO8601 with offset (e.g. 2015-11-11T12:12:00+01)	Time of the logging event.
Severity	one of Fatal/Error/Warn/Info/Debug/Trace	Level of the logging event.
Logger	Namespace+ClassName	Full type name(namespace + class name) of the caller issuing the logging request.
Message	string	Custom message.
ExceptionType	Namespace+ClassName	Full type of the exception.
Exception	string	Should be equivalent of Exception.ToString method. Exception type, message, inner exceptions, stack trace.
Username	string	Optional: The name of the user that is executing the service.
Company	string	Optional: The name of company of the user that is executing the service.
ServiceVersion	string	Optional: Version of the application / service.
ConfirmitSite*	lowercase string	Name of the site in lowercase.
SourceModuleName*	string	Logical module or service, lowercase with “-“ (e.g. “discovery-analytics”, “confirmit-identity”).
SourceModuleType*	one of service/access/iis	Ref the log format specification.
TimeTakenMicros	long, micro seconds	Optional: The time it took to process the request/task/job.
{CustomField}	*	Optional: Services can add any number of custom fields.

Json FieldName (* = mandatory)	Format	Description
Custom fields used for activity logging		
Activity	string	Activity name.
ResourceCompany	string	The name of the company which owns the activity performed on a resource.

Access log file: Expected file name: access-<date>.json

This log type logs service access events. This is useful for gauging traffic patterns, amount of hits to one service from another.

Json FieldName	Format	Description
CorrelationId		See service log.
Service		See service log.
ReferrerService		See service log.
InitiatingService		See service log.
EventTime		See service log.
Severity		See service log.
Logger		See service log.
Username		See service log.
Company		See service log.
ServiceVersion		See service log.
ConfirmitSite		See service log.
SourceModuleName		See service log.
SourceModuleType		See service log.
RequestUri	string	The uri of the request.
ResponseStatusCode	int, any valid http status code	The response status code of the request.
RequestHttpVerb	uppercase string, any valid http verb	The http verb of the request.
TimeTakenMicros	long, micro seconds	The time it took to process the request.
{CustomField}		See service log.

IIS Logs

The ELK system can be used to index IIS logs to give better insight into your traffic. This is most easily achieved by updating IIS to log to the same logging directory as the Confirmit.Site.LogPath Octopus variable, for example in a subfolder called 'IIS'. The log fields in IIS have to match the log fields processed by Logstash in the logstash config. Confirmit SaaS sites use the log fields outlined below.

Name	Field
Date	Date
Time	time
Client IP Address	c-ip
User Name	cs-username

Name	Field
Server IP address	s-ip
Server Port	s-port
Method	cs-method
URI Stem	cs-uri-stem
URI Query	cs-uri-query
Protocol Status	sc-status
Protocol Substatus	sc-substatus
Win32 Status	sc-win32-status
Bytes Sent	sc-bytes
Byte Received	cs-bytes
Time Taken	time-taken
Protocol Verions	cs-version
User Agent	cs(User-Agent)
Referer	cs(Referer)

7.2.5. Event Logs

Event logs are used for investigating issues with applications or the system itself, and for auditing system usage. For example, if a Confirmit installation fails it will log an error in the application log stating why it failed. Confirmit will also create its own event log on servers where it is installed, where errors from modules using the .NET Framework will be logged (in addition to being sent via email to the configured EmailRecipient.)

The size of the event logs can be increased to provide a longer backlog for events on the system.

7.2.5.1. Secure Event Log Viewing

Default configuration allows guests and null logons the ability to view event logs (system, and application logs.) The Security log is protected from guest access by default, though it is viewable by users who have “Manage Audit Logs” permission. The Event log services use the following key to restrict guest access to these logs:

```
HKEY_LOCAL_MACHINE\SYSTEM
CurrentControlSet\Services\EventLog\[LogName]
RestrictGuestAccess: REG_DWORD: 1
```

7.2.6. IIS Configuration

7.2.6.1. IIS Hardening

IIS is relatively secure with its base configuration. As it is module based, only required modules need to be installed, thereby limiting the surface area for potential attacks against the service.

7.2.6.1.1. Moving the Web Root

By default, IIS installs its web root folder on the system drive. If an attacker were to be able to hack a page residing on the system drive, using folder traversal could potentially give the attacker access to system files. Confirmit recommends moving the web root folder to a different drive, and installing Confirmit on a different drive than the operating system as well. Scripts for performing an IIS folder migration can be found around the Internet. Alternatively, the official steps from the IIS blog may be used:

http://blogs.iis.net/thomad/archive/2008/02/10/moving-the-iis7-inetpub-directory-to-a-different-drive.aspx

7.2.6.2. IIS Log Files

IIS supports different log file formats; its own Microsoft IIS log file format, the NCSA Common and the W3C Extended log file formats. The W3C extended log file format is the most versatile as it offers customizable output to accommodate individual needs.

Confirmit recommends setting up IIS to store log files on a different partition than the operating system or setting up a log file archival process if possible to avoid disk space problems if the logs outgrow the available partition space.

7.2.6.2.1. IIS Log File Archiving

Web server and SMTP logs can grow substantially in size on busy systems. Two easy ways of controlling log size are by enabling NTFS compression on the log directory and archiving logs.

Enabling NTFS compression on the log directory will reduce the physical space required to store logs, but can add some overhead to the I/O performance on the system, specifically on very busy servers.

Archiving logs for external storage is a more performance/cost effective solution and can be customized to accommodate most needs. Log archiving can be set up as a scheduled task on the servers. An example script (written for IIS6 but also works with IIS7 when IIS6 WMI compatibility is enabled) can be found in the script repository on the Confirmit Extranet.

7.2.7. IIS Monitoring

Monitoring of the web server health can help resolve situations where the web server becomes unresponsive or unavailable by alerting technical personnel about the server status.

Several 3rd party products can be used for monitoring web service health, amongst others: Mercury SiteScope, HP OpenView, Paessler PRTG, Solarwinds IPMonitor, KeyNote, etc.

7.2.7.1. Using a Basic Page to Monitor Web Server Availability

Alternatively, a basic .aspx page can be used to monitor the availability of the web service. The following script can be saved as an aspx page in the web server root directory. Accessing the file through a monitoring system can monitor the basic health of the IIS server.

```
<%@ Page Language="C#" %>
<!-- IsAlive.aspx
Check that server is alive
-->
<html>
<body topmargin="0"  bgcolor="FFFFFF">
<%
Response.Write("<CENTER><BR>");
Response.Write("<table width=250 bgcolor=green><tr
align=center><td><strong><font color=white>Page loaded
successfully</font></strong></td></tr>");
Response.Write("</table></CENTER>");
%>
</body>
</html>
```

7.2.7.2. Using a Survey to Monitor Survey Engine Availability

Monitoring a dummy survey is a good method of ensuring that your Confirmit server environment is working properly. The approach below will only "cost" one complete transaction, and we avoid the additional work of "flagging" certain projects in the transaction logs. The recommended approach for creating and monitoring a Confirmit survey is as follows because it will only generate one complete:

1. Create a short survey; one page with some static text should be sufficient. Add some text to the page that is easily recognized by the monitoring software such as "Survey Loaded" or similar.
2. Compile the survey as a limited survey, allowing respondents to reenter the survey upon completion.
3. Upload one respondent to the survey and find the respondent’s unique link from the Respondent Editor (or send it as an invitation email from Confirmit if a respondent was uploaded with an email address.)
4. Use the personalized URL to monitor the environment; i.e. http://server/wix/p12345678.wix?r=1&s=XXXXXX

If the 'EncryptSystemRequestParameters' configuration setting is enabled, the alternative link can be used. Following the steps above should enable monitoring of Confirmit end-to-end, as failures on either the front-end or back-end will fail the survey. For best coverage you should monitor all deployment servers internally, and any load-balanced public IP-addresses externally. You will then be notified in the event of a failure in the network infrastructure, and if there are any problems reaching the servers.

Note: This model works as long as the survey is completed multiple times within 30 minutes. If more than 30 minutes is allowed to pass between each survey completion with the same respondent, the transactions are counted as individual completes.

8. Advanced System Configuration

This chapter describes the more advanced setup and settings in the system configuration.

8.1. IIS Advanced Configuration

8.1.1. IIS Application Pool Performance Tuning

While monitoring your servers you may see increased memory consumption due to the surveys or services being loaded into memory. As *32 processes (i.e. any 32-bit process running on a 64-bit Windows OS) cannot allocate more than 4GB of memory, you may have to recycle the IIS process from time to time; especially for busy environments. This limit is noticeable for .NET Framework applications as the effective memory limit becomes approximately 50 – 60% of the maximum process memory size.

Confirmit operates three Software-as-a-Service sites. All sites have 20GB memory installed in the web servers, and the following recycle recommendations are adapted from our current environments. Other recommendations may apply in environments with different memory specifics.

8.1.1.1. Configuration

How to set up the ConfirmitWIXAppPool recycle settings:

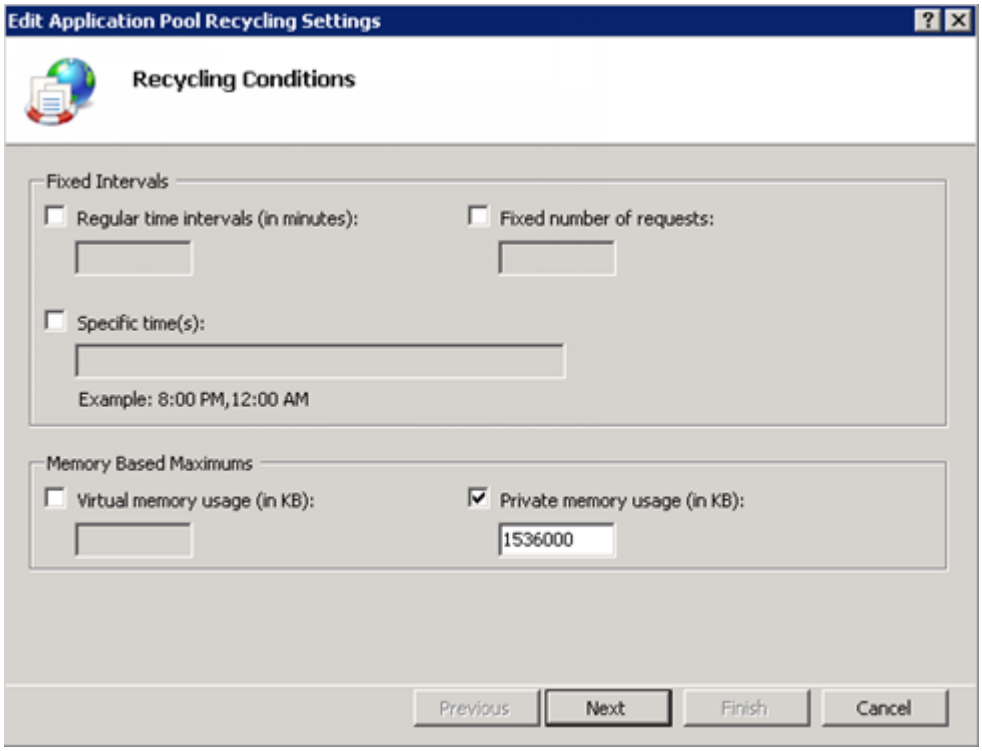


Figure 40 The Recycling Conditions page

1. Disable Regular time interval recycling.
2. Enable Private memory usage recycling, and set to 1536000 (which equals 1546MB).

Note: you should monitor each server's virtual bytes and private bytes for the w3wp.exe processes to ensure that they are not recycled too soon or too late. If the server runs into 'OutOfMemory' exceptions, reduce this value. If the server recycles frequently without any errors, consider increasing this value and decreasing the SurveyPackageSlidingExpiration value in Confirmit.

3. Click **Next** to continue.

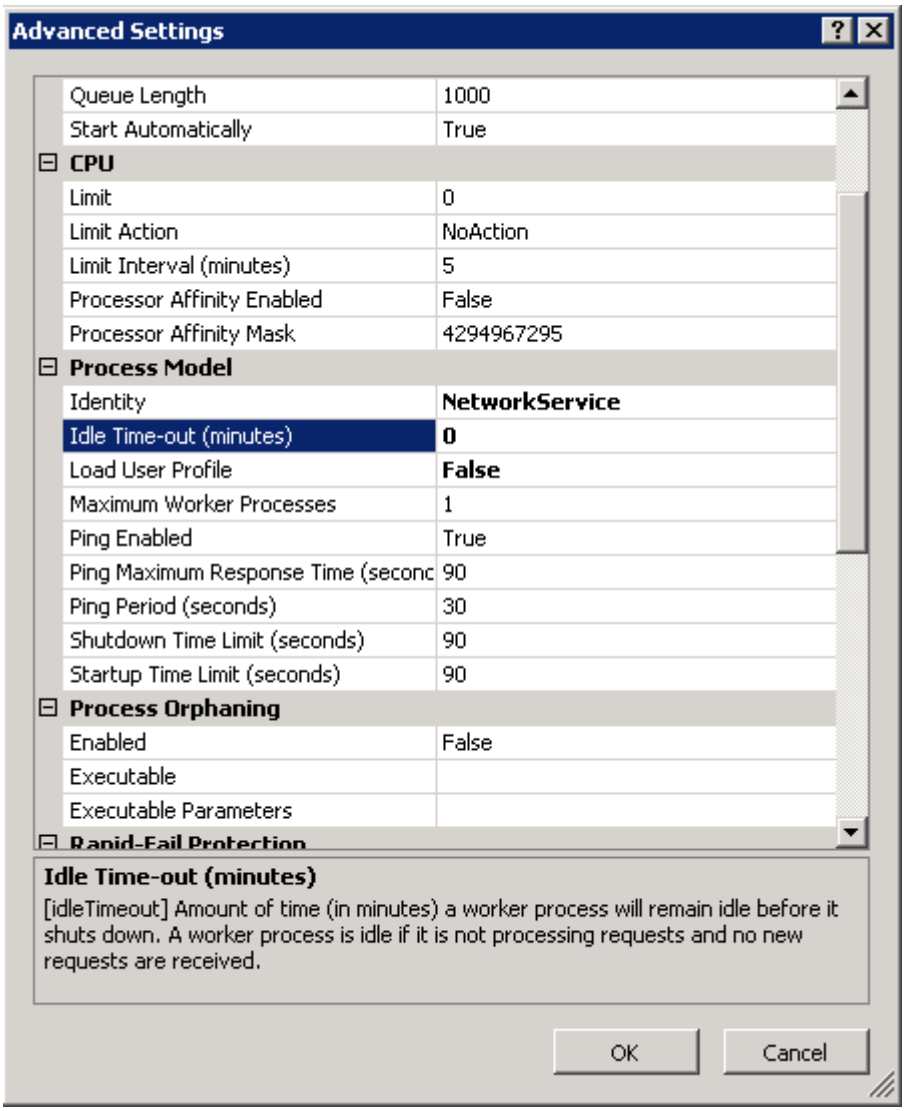


Figure 41 The Advanced Settings page - 1

4. Disable Idle Time-out (minutes) to prevent worker processes from shutting down if they have no work to do.

   This option could have some small performance effects, but these will be most likely for systems with little traffic (most usual for other application pools than those handing surveys). Setting the value to 0 will disable idle time-outs.

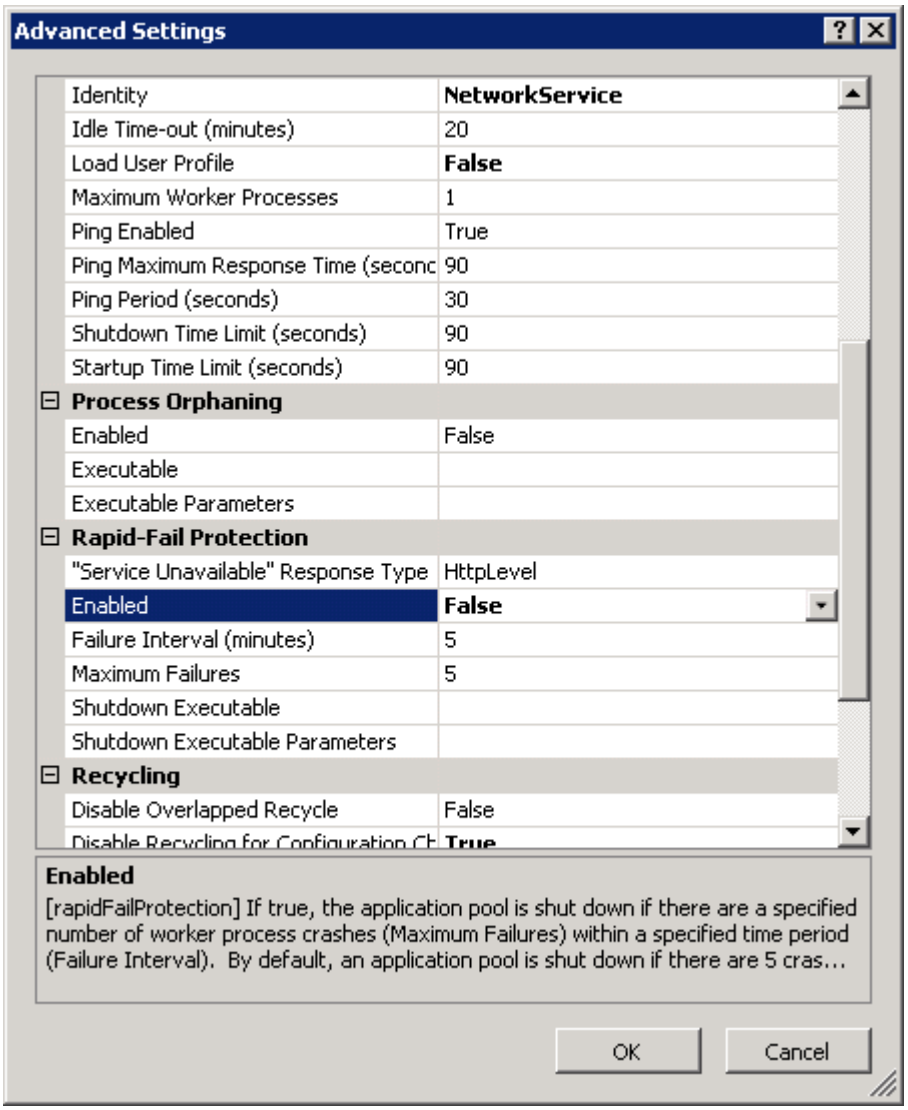


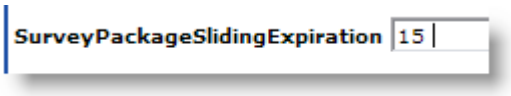
Figure 42 The Advanced Settings page - 2

5. Disable rapid-fail protection to prevent the application pool from shutting down automatically in the event of crashes.

   With the setting disabled, a new worker process will be started repeatedly. If the worker process is shut down, all surveying will stop until the process is manually restarted.

Note: Disabling the fail protection merely bypasses the internal protection feature within IIS; it does not correct the cause of the errors.

6. Within Confirmit we set the SurveyPackageSlidingExpiration to 15 minutes site-wide to prevent idle surveys from occupying memory unnecessarily. Go to **Admin > System Configuration > Deployment**.



7. For each survey server under, reset the SurveyPackageSlidingExpiration to the default value to make it inherit the role setting (which is now set to 15 in the step above).

8.1.1.2. Other Application Pool Recommendations

Authoring Servers:

- ConfirmitAuthorAppPool
  - Idle Time-out (minutes): 0 (original 20)
  - Recycling
    - Private Memory Limit (KB): 1536000
    - Regular Time Interval (minutes): 0

Reportal Servers:

- ConfirmitReportalAppPool
  - Idle Time-out (minutes): 0
  - Recycling
    - Private Memory Limit (KB): 1536000
    - Regular Time Interval (minutes): 0

Web Service Servers:

- ConfirmitApiAppPool
  - Idle Time-out (minutes): 0
  - Shutdown Time Limit (seconds): 300
  - Recycling
    - Private Memory Limit (KB): 1536000
    - Regular Time Interval (minutes): 0

8.1.2. IIS Compression Tuning

Some pages from Confirmit can become very large, and with the built-in compression support found in most browsers, content delivery time could potentially be reduced with a fair amount of IIS compression is enabled for IIS. Compressed pages are transferred faster to the client browser requesting the page, and also reduce the bandwidth requirement for transmission of the same amount of content. Compressed pages are automatically decompressed on the client so the process is transparent to the end-user. If the browser does not support compression the server will send pages uncompressed, so the only backside for this is a slight performance overhead on the server for handling the compression.-

By default, IIS 7 compression is only enabled for static content, not for dynamic content. Dynamic content compression needs to be enabled in the server role configuration before it can be set in the IIS administration console.

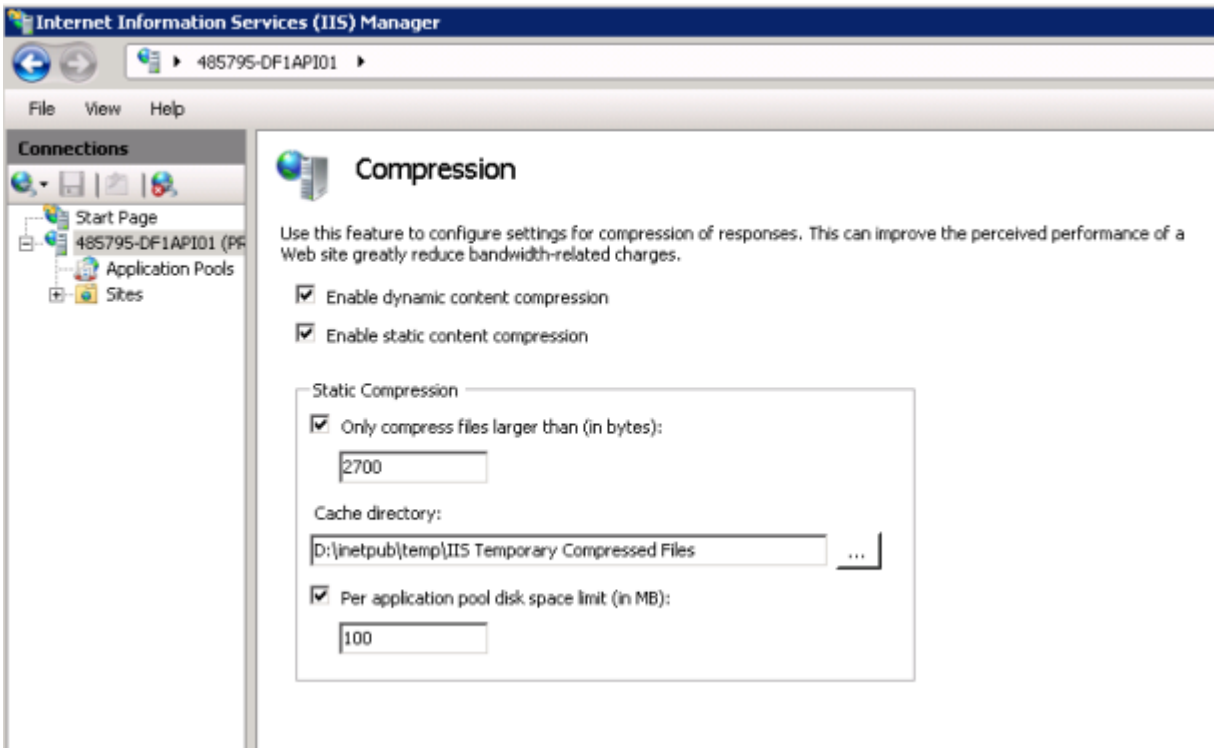


Figure 43 Compression settings for servers

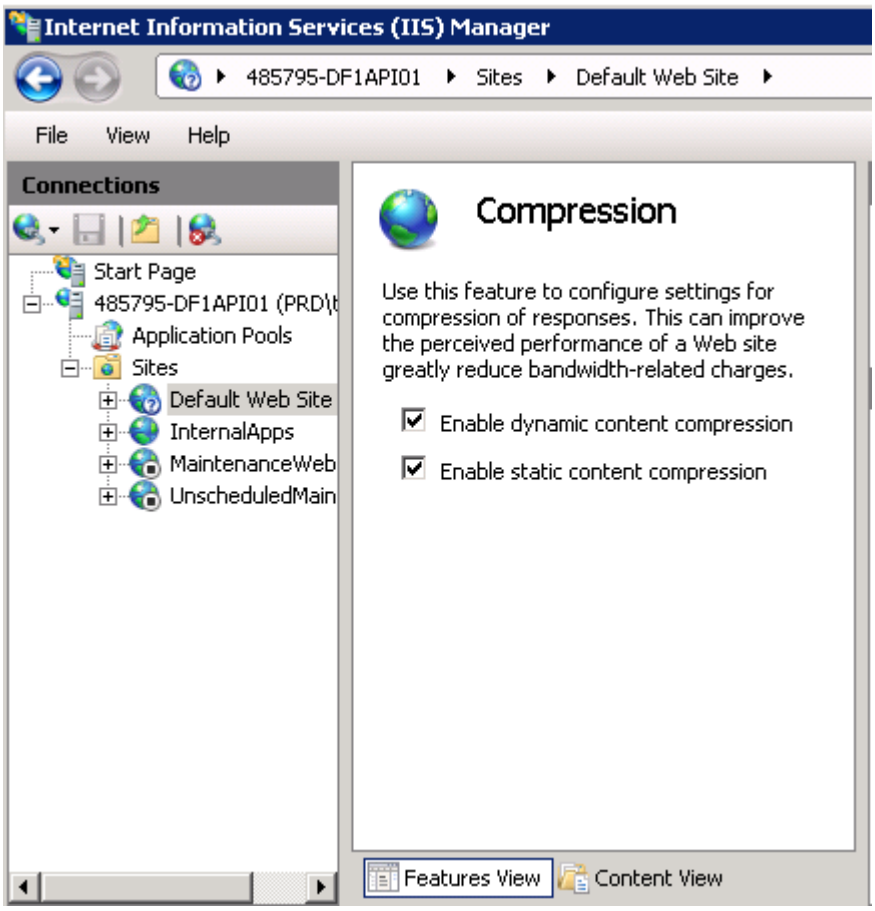


Figure 44 Compression settings for web site

IIS supports many additional compression settings that are not visible in the GUI, these can be set using the appcmd tool or by editing the applicationHost.config file found in the **c:\windows\system32\inetsrv\config** folder.

Most options are documented in the IIS.Net config reference, found at: http://www.iis.net/ConfigReference.

Warning
Confirmit strongly recommends that you perform a back-up of the existing configuration before making any changes.

For example, the following excerpts from the **applicationHost.config** file on Confirmit's servers have been shown to work quite well with compression. More detail is available at:

http://www.iis.net/ConfigReference/system.webServer

In the **<system.webserver>** section, add the following:

```
<httpCompression directory="%SystemDrive%\inetpub\temp\IIS Temporary
Compressed Files" minFileSizeForComp="256">
  <scheme name="gzip" dll="%Windir%\system32\inetsrv\gzip.dll" />
    <staticTypes>
      <add mimeType="text/*" enabled="true" />
      <add mimeType="message/*" enabled="true" />
      <add mimeType="application/x-javascript" enabled="true" />
      <add mimeType="application/x-javascript charset=utf-8"
enabled="true" />
      <add mimeType="application/atom+xml" enabled="true" />
      <add mimeType="application/xaml+xml" enabled="true" />
      <add mimeType="*/*" enabled="false" />
    </staticTypes>
    <dynamicTypes>
      <add mimeType="text/*" enabled="true" />
      <add mimeType="message/*" enabled="true" />
      <add mimeType="application/x-javascript" enabled="true" />
      <add mimeType="*/*" enabled="false" />
  </dynamicTypes>
</httpCompression>
```

8.1.3. Preventing Search Engine Indexing of Surveys

To prevent indexing of surveys by search engines, copy the text below into a file called **robots.txt** and place it in the web server root folder on deployment servers.

```
User-agent: *
Disallow: /
```

8.1.4. Avoiding Issues with IIS Request Filtering or URL Scan

If you have activated Request Filtering or URLScan software, you will need to allow the following verbs:

- GET
- HEAD
- POST

Note that the Horizons deployment process will clear existing IIS request filtering rules and define the necessary file extension and verb rules for each sub-application during deployment. If modification of request filtering rules has been disallowed on the IIS root level, you will see errors like this when accessing affected applications:

Figure 45 Example of an error due to filtering rules being locked

To avoid this, make sure that you remove any **lockitem=”true”** flags set in the applicationHost.config file for IIS. Example below:

```
<verbs allowUnlisted="true" applyToWebDAV="true">
<add verb="TRACE" allowed="false" lockItem="true" />
</verbs>
```

8.1.5. Using Performance Counters to Troubleshoot Survey Problems

The Confirmit survey engine also installs a number of performance counters that can be used to monitor and troubleshoot the web server. The counters are installed under the performance object “Confirmit Survey Engine .Net”. The following performance counters are registered:

Category	Counter name	Description
Cache	# SurveyPackages	Number of survey packages currently cached
Cache	SurveyPackage Load Time	Average time in sec used to load survey packages
Cache	SurveyPackage Turnover Rate	Number of additions and removals to the SurveyPackage cache per second
Monitor	CPU milliseconds	Average total CPU time in milliseconds used by a request
Monitor	CPU milliseconds Usercode	Average usercode CPU time in milliseconds used by a request
Monitor	Page errors/sec	Average number of survey page errors per sec
Monitor	Page hits/sec	Average number of survey page hits per sec
Monitor	Real milliseconds	Average total real time in milliseconds used to execute a request
Monitor	SQL Compile CPU milliseconds	Average CPU time in milliseconds used to compile statements
Monitor	SQL Compile Elapsed milliseconds	Average elapsed time in milliseconds used to compile statements
Monitor	SQL Execution CPU milliseconds	Average CPU time in milliseconds used to execute statements
Monitor	SQL Execution Elapsed milliseconds	Average elapsed time in milliseconds used to execute statements
Monitor	SQL Logical Reads	Average number of pages read from the data cache
Monitor	SQL Physical Reads	Average number of pages read from disk
Monitor	SQL Scan Count	Average number of scans performed
SPGen	Alter Save Proc/sec	Rate showing how many store procedures that are altered per sec
SPGen	Alter Select Proc/sec	Rate showing how many select procedures that are altered per sec

SPGen	Create Save Proc/sec	Rate showing how many save procedures that are created per sec
SPGen	Create Select Proc/sec	Rate showing how many select procedures that are created per sec
SPGen	Exec Save Proc/sec	Rate showing how many save store procedures that are executed per sec
SPGen	Exec Select Proc/sec	Rate showing how many select procedures that are Executed per sec

8.1.6. Using Confirmit BitStream Performance Counters

The Confirmit BitStream service contains several performance counters which can be utilized to troubleshoot BitStream issues. The following performance counters are registered:

Category	Countername	Description
Monitor	Write - # Currently Runinng	Current number of BitStream writing operations.
Monitor	Write - # Total Started	Total number of BitStream writing operations since service start.
Monitor	Write - % Success	Current percentage of successful BitStream write operations.
Monitor	Write – Generation Time	Current BitStream write generation time.
Monitor	Query - # Currently Running	Current number of BitStream queries running.
Monitor	Query - # Total Started	Total number of BitStream queries run since service start.
Monitor	Query - % Success	Current percentage of successful BitStream queries.
Monitor	Query – Execution Time	Current BitStream query execution time.

8.2. Using Confirmit with a Load Balancer with SSL Acceleration

When the '**SSLAuthorCertificate**' config setting is enabled, Authoring logins will be redirected to SSL even if the '**EnforceSSLOnSite**' is not - the former only allows SSL to be used - and enforces it for the entire session for all users. If the user reaches the server via HTTP, a redirect will occur to HTTPS for the login screen, and once authenticated, back to HTTP again. If reaching the server via HTTPS in the first place, the session will remain encrypted. This will also be the case if the user has 'Always use SSL' selected in the user settings, or if this is enforced on the company level as well.

The redirect happens on the server side, which will only work if the server itself decrypts the SSL traffic. If SSL traffic is handled by an external load balancer/SSL accelerator, two configuration settings have been added to support this mode in Confirmit: the SSLAcceleratorMode and SSLAcceleratorPort settings. In order to use these settings you can follow these steps:

1. Add a second listening port for HTTP in the IIS manager for the web site on which Authoring runs.

This can be any port except port 443 - Confirmit uses port 81 for our systems (so both ports 80 and 81 are listening for HTTP traffic.)

2. Configure your load balancer/SSL accelerator module to translate SSL requests and forward the decrypted transactions to the port you set up in step 1.
3. Set the SSLAcceleratorPort in the Confirmit settings to the port number you set up in step 1.
4. Enable the SSLAcceleratorMode setting.
5. **Save**.

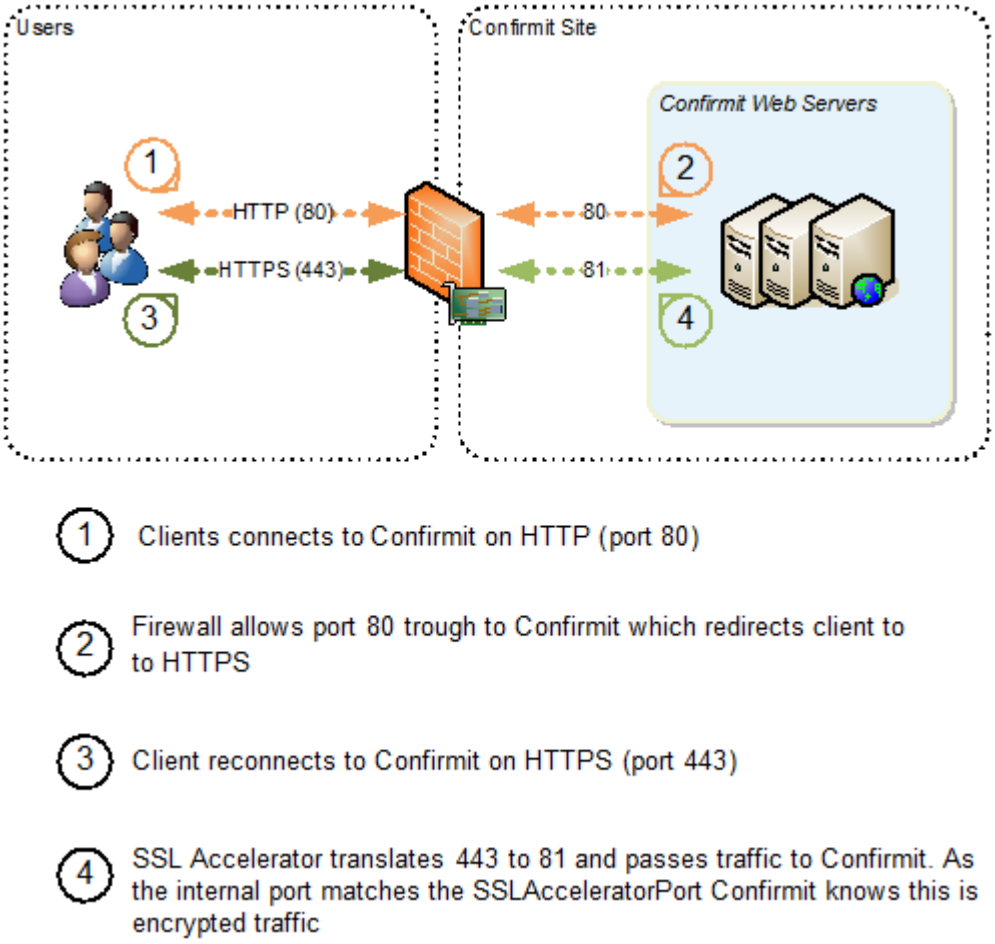


Figure 46 Using Confirmit with a load balancer

8.2.1. Automated Load Balancing During Deployment

Octopus has the ability to take your servers in and out of the load balanced pool during the installation of features that will drop user sessions.

There are no configuration options for this Load Balancing management functionality and it must be implemented exactly as described below.

The load balancing management functionality has been implemented for the following features:

- Reportal
- Authoring
- Web Services

8.2.1.1. Configuring Automated Load Balancing

This functionality can only be configured in one specific manner.

Your load balancer will need to be configured to examine an isalive.htm file in the default Webroot of the Confirmit application on the server.

During the configuration / installation of the three features listed, the Octopus will:

- Look up the IIS webroot on that server.
- Rename isalive.htm file - your load balancer will see that this file doesn't exist, hence taking it out of the load balanced pool.
- Install the feature.
- Rename the isalive.htm file back to the correct name - your load balancer will see this file exists and will add it back to the load balanced pool.

8.3. Database System Redundancy

8.3.1. SQL Server in Windows Cluster

SQL server can be set up as a cluster resource if necessary. This would add an extra layer of redundancy if set up correctly, as a cluster resource can be failed over to a secondary server if the primary should fail. Clustering also allows for easier upgrades of the operating system on database servers as a cluster node can be upgraded and rebooted if necessary without affecting SQL Server performance. Once the update is complete, database resources can be failed over to the updated server and the process can be repeated for the other cluster node(s) in the cluster.

In order for a SQL Server cluster to have any use, the server must also be set up with a clustered IP address for connections, and with a cluster resource for database storage such as a DAS or SAN device.

8.3.2. Multiple SQL Server Instances

Confirmit supports multiple SQL Server instances handling survey databases. This allows administrators to set up additional SQL resources if the performance of the primary server should be insufficient. The additional server instance can be installed as a new named instance on the same physical server as the primary instance, or on a separate physical server.

Some manual steps are required to attach additional SQL Server instances to Confirmit, and some points should be taken into consideration before starting.

8.3.2.1. SQL Server Collation

Confirmit features that require reading or writing from multiple databases simultaneously will assume that the SQL collation is identical on all instances. Once SQL Server has been installed, the collation cannot be changed, so make sure to select the same collation for additional servers as the primary. This is equally important both for adding new instances to an existing environment, and when installing a SQL Server to migrate the existing SQL Server databases to.

Note: If a new SQL Server instance has been installed with the wrong collation, it is necessary to re-install SQL Server. If the databases are put into use, collation conflicts may render the site in an unusable state.

8.3.2.2. SQL Server Logins

Confirmit will use the same encrypted login credentials stored in the system configuration database when connecting to secondary servers, so SQL Server logins must be created with identical user names and passwords as the primary server. On the primary server this is done automatically during installation of Confirmit, but for secondary servers it must be done manually by the database administrator. We recommend using Microsoft's own procedures for migrating the logins while retaining the user SID's so that no remapping of these are needed afterwards if databases are moved between the instances: https://support.microsoft.com/en-us/help/918992/how-to-transfer-logins-and-passwords-between-instances-of-sql-server.

8.3.2.3. Linked Servers

In order for Confirmit to read or write data from multiple instances, the servers must be mutually set up as linked servers to provide server awareness between the instances. These linked server objects are created automatically by the deployment steps during installation of Confirmit Horizons, but needs to be taken into consideration for SQL migrations.

8.3.2.4. System, Survey and Hub Databases

If installing on a system that already has more than one instance of SQL Server intended for Confirmit usage, the first server (the one given during Confirmit installation) must always host the Confirmit system databases. Subsequent SQL servers are typically added to host additional survey and hub databases.

8.3.2.5. Creating Aliases for the New Database Server

If the primary database server uses an alias (for instance: 'ConfirmitDB'), setting up a new alias on all application servers with a similar alias (i.e. 'ConfirmitDB2') would be recommended. If IP addresses or NetBIOS names are used for connecting to SQL Server, this step is not required. Note that we strongly recommend configuring Confirmit Horizons to connect to SQL using local SQL aliases or DNS CNAME's. The reason for this is that connection strings are hard coded into survey packages, so if you at a later time migrate to a new database server with a new IP or name, you need to relaunch all your surveys in order for them to work.

8.4. Database Encryption

Confirmit Horizons v15 Patch 02 introduced a new Database Encryption feature which allows Confirmit Authoring users to encrypt survey databases "at rest" on the database server. The add-on is chargeable and requires an updated license file in order to be used.

If your Confirmit environment has more than one SQL Server Instance (see Multiple SQL Server Instances on page 80 for more information) you can select which instance(s) you want Database Encryption activated on.

Note: Database Encryption requires SQL Server Enterprise Edition.

8.4.1. Configure Database Encryption

After purchasing the add-on and installing a new license file, the following steps are required to prepare your Confirmit site for Database Encryption:

8.4.1.1. Preparing SQL Server for Database Encryption

To prepare the SQL Server(s) you have to create a certificate for encryption on your DB server. This is done by running the following statement in the SQL Server Management Studio (replacing values where required):

```
USE_master;
GO
CREATE_MASTERKEYENCRYPTIONBYPASSWORD='EncryptionPassword';
go
CREATE_CERTIFICATE EncryptionCertificate WITHSUBJECT='Encryption
Certificate Confirmit site';
GO
```

If enabling Database Encryption on multiple SQL Server instances, the certificate must be created with the same name on all instances. Once the Certificate is created, you are strongly recommended to take a backup. This can be done by running the following command (replacing values where required):

```
USE master
GO
BACKUPCERTIFICATE OnDemandEncryptionCertificate
TOFILE='[file location]\[EncryptionCertificateName].cer'
WITHPRIVATEKEY (FILE='[file location]\[EncryptionCertificateName].pvk',
ENCRYPTIONBYPASSWORD='EncryptionPassword')
GO
```

We strongly recommend that you backup and store the certificate in a secure location.

8.4.1.2. Preparing Confirmit for Database Encryption

Once the certificate is created, Database Encryption can be enabled in Confirmit:

1. Open Confirmit System Configuration.
2. Expand the Security settings section.
3. Enter the name of your certificate in the EncryptionCertificateName box.

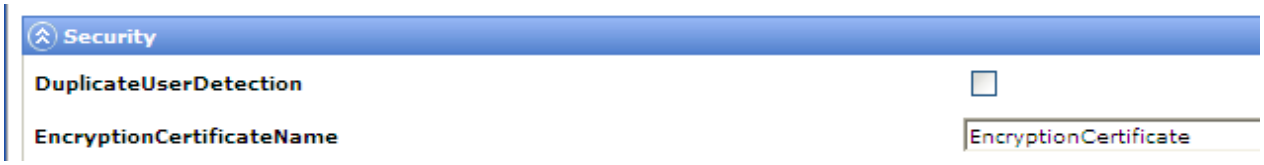


Figure 47 Entering the certificate name

4. Expand the SqlServer role, select the SQL server(s) where the certificate was created, and enable the SQLServerUsedForEncryption checkbox to enable encryption for the server.

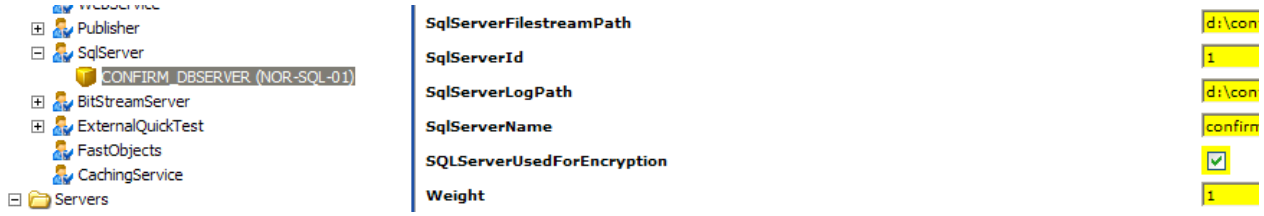


Figure 48 Checking the SQLServerUsedForEncryption checkbox

Database Encryption is provided on a per company basis in Confirmit. To enable the feature for a company:

5. Open the **Admin > Accounts > Companies** menu.
6. Select the desired company.
7. Navigate to the Add-Ons tab, and add the Database Encryption add-on to the Selected list.

Figure 49 Adding the Database Encryption add-on

8. To create an encrypted database, the 'Enable Database Encryption' checkbox must be selected when launching a survey.

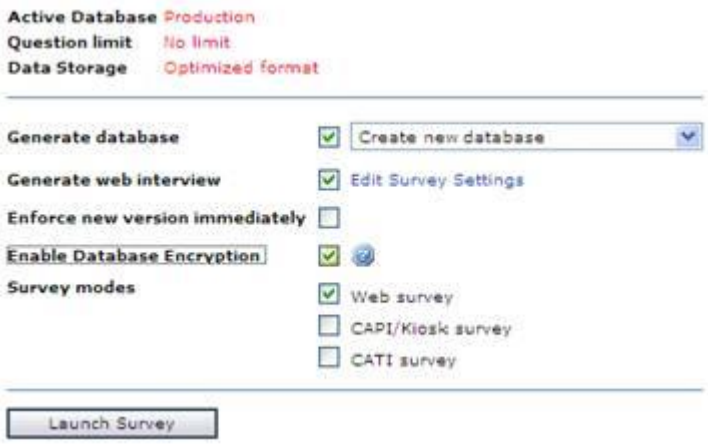


Figure 50 Selecting Enable Database Encryption

Database Encryption only applies to new databases, or databases that are re-launched with the 'Create new database' option.

Important
Selecting 'Create new database' will cause a new database to be created and will result in loss of data if the data is not backed up.

9. Monitoring Confirmit

Confirmit is a complex application suite that should be monitored and maintained closely by trained professionals. A number of self-monitoring features are built into the application itself and can be accessed from the Admin menu after logging on by users with appropriate permissions to the system. These features are described in detail in the Confirmit Administrator Manual.

The built-in features are useful for ad-hoc monitoring and for troubleshooting specific problems, but for day-to-day tracking, additional features should be deployed.

9.1. The Confirmit Monitoring Utility

The Confirmit SaaS Operations team has built a companion utility for the built-in monitors that extracts data to a database on an aggregate level and presents the results in a graphical display along with warning and alert features. The system can be provided along with installation instructions to On-Premise customers.

Note: As this utility is not part of the Confirmit main product line, no support is available for troubleshooting/update requests. The product will be updated at a pace to be determined solely by the Confirmit SaaS Operations team.

The graphical report is available as a .NET application and looks as shown in the example below:

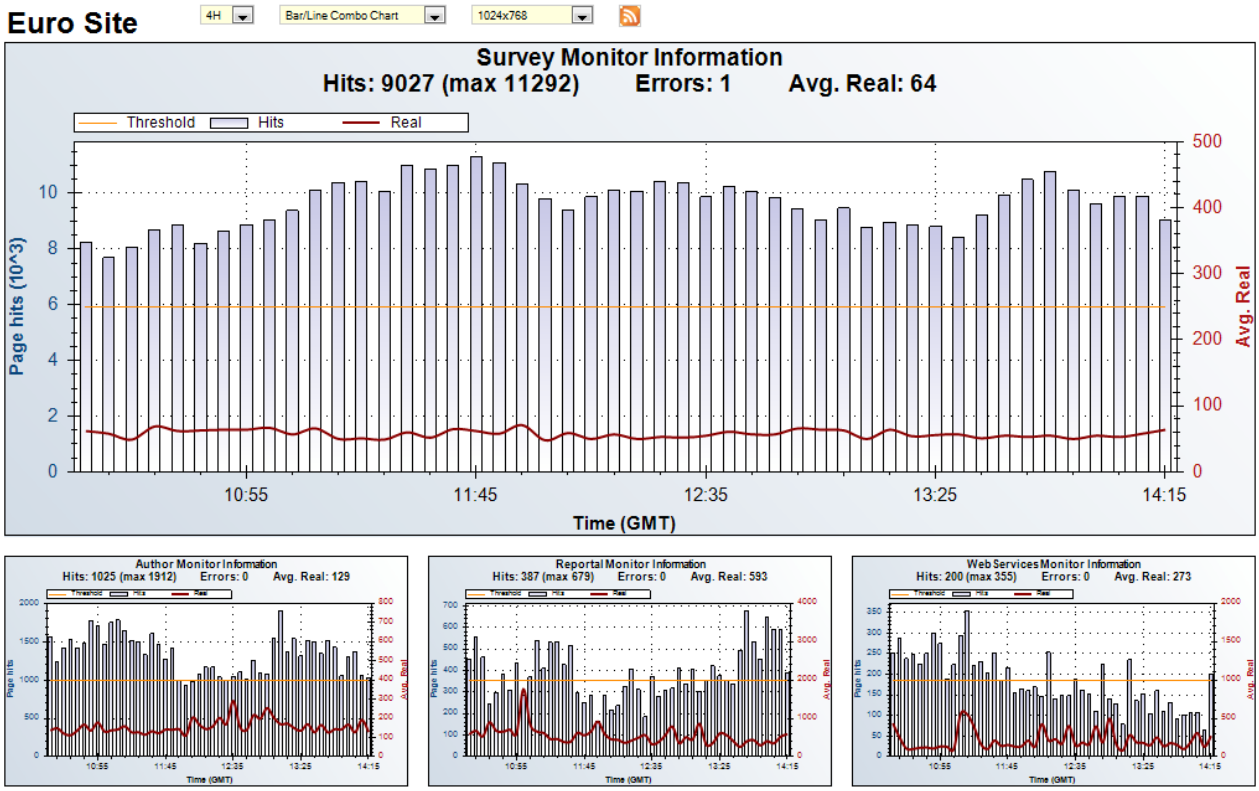


Figure 51 Example of the monitoring utility screen layout

9.2. Generic Server Monitoring

Confirmit recommends using a monitoring application utility with a wide array of capabilities in order to monitor the applications as widely as possible. Some of the more generic items that Confirmit recommends to be included are:

- CPU utilization % levels for server devices (warn/alert on prolonged % thresholds).

- Memory utilization (warn/alert on % thresholds).
- Disk space utilization (warn/alert on % or MB/GB thresholds).
- Availability of web server, database server and mail server applications.
- If HTTPS is used: Website SSL Certificate expiration.

9.3. External Location Monitoring

In addition to internal monitoring systems, Confirmit also uses external services for monitoring performance and availability of web applications from various worldwide locations. This can be used for Confirmit Authoring, Reportal, Surveys, REST API and Web Services.

Note: If you wish to monitor Authoring/Reportal logins, this will require supporting a login transaction.

9.4. Confirmit-Specific Monitoring and Recommended Sensors

Monitoring of certain Confirmit components should be set up for production installations. For this purpose Confirmit recommends monitoring applications capable of running:

- SQL Server queries.
- WMI queries.
- Custom VBScript/Jscript/PowerShell/Executables returning status values.

The recommended monitor sensors are as follows:

9.4.1. Confirmit System Service Monitoring

Confirmit installs various system services when installed based on certain roles. Availability of these services should be monitored to ensure systems depending on them are functioning properly. Services include:

- **Confirmit BitStream Service -** Used for creating and queries against Confirmit BitStream files, which are used by many application components. Reportal, Rapid Results and Panel sampling are some features that may not function if this service is not running.
- **Confirmit Caching Service -** Used by Reportal for providing fast access to recently used page contents. If the service is not running, Reportal performance will be drastically reduced and may not function at all.
- **Confirmit Task System Service -** Runs tasks from the queue requested by users such as data exports, launch survey and respondent emailing. If the service is stopped, tasks remain in the queue until the service is started again.
- **Confirmit Task Scheduler Service –** Runs on Batch servers and monitors SmartHUB enabled projects set to continuous synchronization. The service schedules Hub Confirmit Project Loader tasks if changes between data in a Hub and its data sources are detected. If the service stops, Hubs set to continuous synchronization will not be updated automatically.
- **Confirmit Agent Controller Service –** Runs on servers with the Task System role and manages the agent processes that process requests made by the RabbitMQ queue. If the service is stopped, requests remain queued in RabbitMQ until the service is started again.
- **Confirmit Searching and Indexing Service -** This service indexes survey designs for fast searching. The Service monitors changes in the Confirmit Authoring GUI and re-indexes changes. If the service stops, searching from the menu bar does not return any results.

9.4.2. SQL Query for Monitoring Surveys Returning Errors

The following SQL statement will return the amount of distinct surveys on the system returning errors to respondents in the last five minutes:

```
select count(distinct(projectid))
from [confirmlog]..[surveymonitor]
with(nolock)
where errorflag<>0
and performed >dateadd(mi,-5,getdate())
and test=0;
```

The monitoring system could run this at a frequency of 5 minutes or less. If there are many live surveys on the system (more than 100) then warning/error thresholds could probably be set between 5 and 10. Setting a lower value introduces the risk of "over-alerting" on survey errors generated by script code – errors that project owners also receive.

9.4.3. SQL Query for Monitoring Errors from Authoring

Similar to monitoring errors from surveys, the following query may be used to report on the number of errors recently generated by Confirmit Authoring:

```
select count (id) as count
from [confirmlog]..[reportmonitor] with (nolock)
where errorflag<>0
and performed > dateadd(mi,-10,getdate())
and context=2
```

Changing the context value from 2 to 3 will give results for Confirmit Express while omitting the context altogether will return both Authoring and Express results.

9.4.4. SQL Query for Monitoring Errors from Reportal

The following query may be used to report on the number of errors recently generated from Reportal:

```
select count(id) as count
from [confirmlog]..[reportalmonitor] with(nolock)
where errorflag<>0
and performed >dateadd(mi,-10,getdate())
```

9.4.5. SQL Query for Monitoring Errors from Web Services

The following query may be used to report on the amount of errors recently generated from XML Web Services:

```
select count(id) as count
from [confirmlog]..[webservicemonitor] with(nolock)
where errorflag<>0
and performed > dateadd(mi,-10,getdate())
```

9.4.6. SQL Query for Monitoring Delays in Task Queue Execution

The following SQL statement returns the number of tasks in queue that are delayed by more than ten minutes:

```
Select count(id) as QueuedTasks
from [confirm].dbo.taskinstance (nolock)
where status=0
and datediff(mi,schedule,getdate())>10;
```

If a single project is queued up with many different tasks and this is expected behavior that could be ignored as long as it doesn't affect other projects, the above query may be modified to return a count of distinct projects having delayed tasks in queue instead of the total number of tasks:

```
select count(distinct(td.projectid)) as QueuedTasks
from [confirm].dbo.taskinstance ti (nolock)
left join [confirm]..taskdefinition td (nolock)
on td.id = ti.taskdefinitionid
where ti.status=0
and datediff(mi,ti.schedule,getdate())>10;
```

9.4.7. SQL Query for Monitoring Failing Launch Survey Tasks

If users are unable to launch new or updated surveys, the production status of the system is pretty much at a standstill. The query below will return the number of Launch Survey tasks that have failed recently:

```
select count(distinct(td.projectid))
from taskinstance ti (nolock)
join taskdefinition td (nolock)
on ti.taskdefinitionid = td.id
where ti.tasktypeid=43
and ti.status not in (0,1,4)
and ti.starttime>dateadd(mi,-15,getdate());
```

9.4.8. SQL Query for Monitoring the Number of Attached Survey Databases

Confirmit installations with many active surveys should utilize the automatic survey database detachment/attachment feature. It is also recommended to monitor how many survey databases are attached at a given time if you suspect your database server is nearing capacity – detaching inactive databases frees up resources for live ones, and reduces time spent for failover if you are running an active-passive SQL cluster. The following query returns the amount of attached survey databases on a particular SQL Server instance:

```
select count(*)
from confirm_admin.dbo.surveydatabases
where active = 1;
```

9.4.9. SQL Server Audit for Surveys

This option enables SQL Server auditing at the survey database level. Refer to the Confirmit Administrator Manual for details on how to turn it on. When enabled, all surveys created for a company will have database auditing enabled. This audit log is not accessible through the Confirmit user interface. To access the audit logs, go to the **SQLAudit** folder configured in **System Configuration**. You can also query the SQL Audit logs by using:

```
SELECT * FROM sys.fn_get_audit_file ((SELECT audit_file_path FROM
sys.dm_server_audit_status WHERE name = 'Audit-SQL'),default,default)
```

9.4.10. SQL Query for Monitoring REST API Server Errors

The REST API servers are the backbone of a lot of the services in Confirmit. The logs are IIS based, meaning that all errors in the 500 range are server error messages.

```
select count(id) as count
from [confirmlog]..[qMetadataApiMonitor] with(nolock)
where statuscode >=500
and timestamp >dateadd(mi,-10,getdate())
```

9.5. Monitoring Example

This overview is provided to give an example of which systems should be monitored, broken down per server role. Keep in mind that in cases where servers are set up resilient, all hosts should have separate monitoring. Setting up an "isalive" webpage so monitoring systems can verify IIS connectivity is highly recommended.

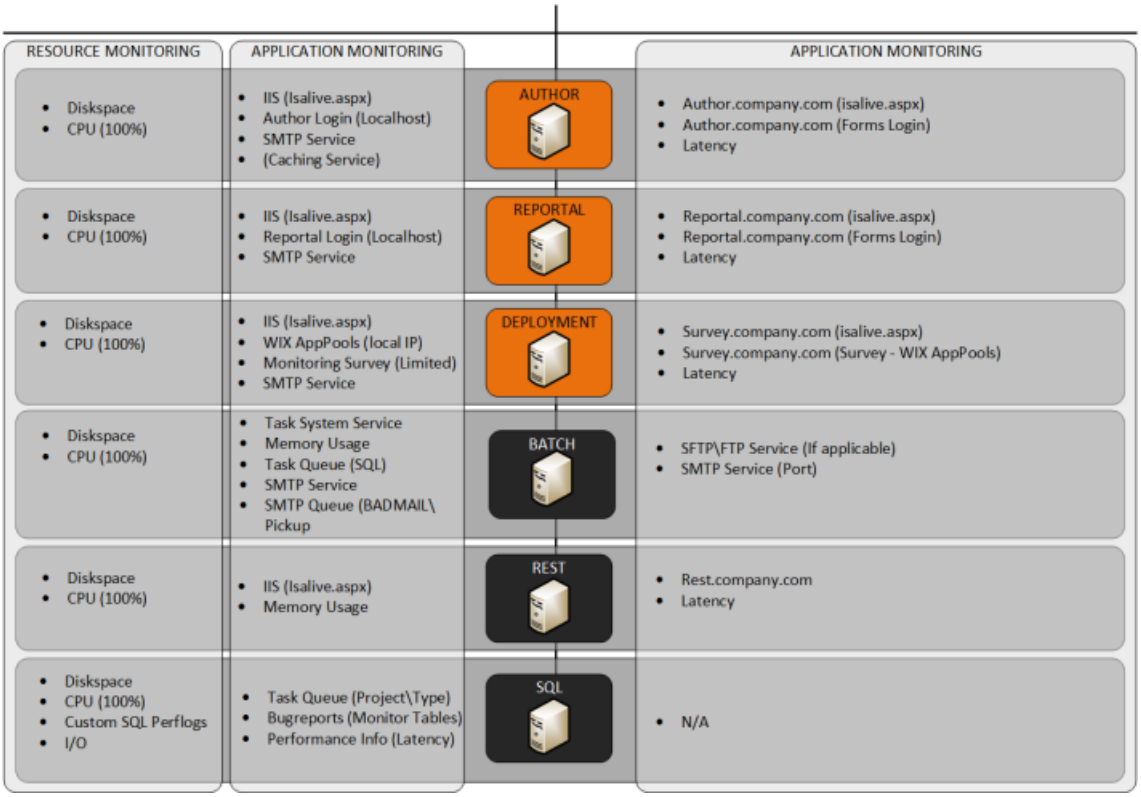


Figure 52 Example of the systems to be monitored

The above is only displayed as a "best practices" example; more systems may have to be monitored to ensure that a Confirmit site is healthy.

Appendix A - Configuration Setting Descriptions

This table lists the configuration settings available to the server administrator, and gives a brief description for each.

ConfigName	ConfigDescription
AbortedTasksLimit	Number of aborted tasks allowed before the recurrence pattern is disabled.
ActionManagementCaseEmailServiceLink	AM Case EmailServiceLink
ActionManagementSqlServer	Action Management SqlServer
ActionManagementURL	Action Management URL
ActiveDashboardSqlServer	Active dashboards SQL server instance
ActiveDashboardURL	Active Dashboard URL
AggregateSQLStatistics	When enabled, will collect CPU statistics for page hits for .NET web application in the surveymonitor table.
AllowActivePdfEngine	Per-site option. Indicates if the ActivePdf engine is allowed for PDF export
AllowEssentialObjectsEngine	Per-site option. Indicates if the EssentialObjects engine is allowed for PDF export
ArchivingBatchSize	Number of projects checked/archived before sleeping
ArchivingEnabled	Specifies whether archiving is enabled on the server. Archiving must be initiated separately and is not started simply by enabling this option.
ArchivingSleepTime	Sleep time between each batch of checking and archiving
AscribeFtpLogEnabled	When enabled, FTP uploads to the Ascribe server will be logged verbosely in the task log.
AscribeFtpServer	Specifies the IP address or DNS server name of the remote Ascribe FTP server.
AudioUploadRootFolder	Audio upload root folder (UNC).
AuthorC	NULL
AuthorN	NULL
BatchTimeout	Sets the timeout limit for executing tasks. (Seconds)

ConfigName	ConfigDescription
BitStreamClientConnectionTimeout	Connection timeout for remoting calls.
BitStreamClientQueryTimeout	Timeout in milliseconds for remoting calls.
BitStreamPath	Specifies the local path to Confirmit BitStream files for the BitStream service. Normally <confirmdata>\BitStreams.
BitStreamQueryTimeout	BitStream query timeout in seconds
BitStreamRead	When checked, the server will register as a BitStream Reader.
BitStreamSyncRepositoryPath	Shared UNC path for the BitStream file set repository. If set, all BitStream servers will use this for synchronizing local file sets.
BitStreamURL	IP/Servername:Port used by other Confirmit servers to connect to the BitStream service on this server.
BitStreamWeight	Indicates weight ratio for BitStream project file set deployment on BitStream servers (used in multi-server systems). Value "0" indicates that no file sets are created on the server. Weights for multiple servers if set differently on the server level are relative to each other. Out of (Sum weight across all servers), (Weight on server x) number of file sets are created on server x.
BitStreamWrite	When checked, the server will register as a BitStream Writer (will produce BitStream files.)
BugReportEmailingLevel	A setting to control bugreport emailing from the batch system
BugreportSender	Email address to use as the sender address for error messages generated by Confirmit.
CachingServiceInstalled	NULL
CachingServiceServerName	Used during installation/upgrade to set the Caching Service Server Name in relevant application configuration files locally on each server. (IP/Windows server name)
CachingServiceServerPort	Used during installation/upgrade to set the TCP port number for the Caching Service.
CapiAndroidAppCurrentVersion	Specifies which version of CAPI Android App is the current. Must be updated after upgrading CAPI Android App download repository.
CapiAndroidAppMandatoryUpdate	When checked, app users will not be able to synchronize with an app version lower than the CapiAndroidAppMinVersion.
CapiAndroidAppMinVersion	This is the minimum app version that is supported. Updates to at least this version can be enforced by checking CapiAndroidAppMandatoryUpdate.

ConfigName	ConfigDescription
CapiConsoleCurrentVersion	Specifies which version of CAPI is the current. Must be updated after upgrading CAPI console download repository.
CapiIosAppCurrentVersion	Specifies which version of CAPI iOS App is the current. Must be updated after upgrading CAPI iOS App download repository.
CapiIosAppMandatoryUpdate	When checked, app users will not be able to synchronize with an app version lower than the CapiIosAppMinVersion.
CapiIosAppMinVersion	This is the minimum app version that is supported. Updates to at least this version can be enforced by checking CapiIosAppMandatoryUpdate.
CapiServerCertificateId	Stores generated CAPI certificate data used for issuing activation files for CAPI clients and authorization of installed consoles.
CapiSynchronizationServerUrl	Configures the Web Service URL used by CAPI clients for synchronizing survey contents and data.
CapiSyncLogExportRowsLimit	Maximum number of entries that can be exported from CAPI synchronization log
CapiWindowsAppMandatoryUpdate	When checked, app users will not be able to synchronize with an app version lower than the CapiWindowsAppMinVersion.
CapiWindowsAppMinVersion	This is the minimum app version that is supported. Updates to at least this version can be enforced by checking CapiWindowsAppMandatoryUpdate.
ChangeTrackingRetentionHours	Specifies retention hours used when enabling change tracking on a survey database.
CodingExportMaxNoOfResponses	The maximum number of responses exported using the Online Coding Administration export action.
CommandTimeout	Specifies how long a survey can attempt to run a SQL command before timing out. (Seconds)
ConcurrentReportViewRequestLimit	Maximum number of report viewers that can request a report page at the same time (The maximum number of concurrent requests for report view). Does not apply to preview and exports.
ConfirmArchive	Sets the target path for archived Confirmit projects. (UNC-path on archiving server)
ConfirmData	Specifies the local target path for generated Confirmit data files.
ConfirmitCookieSuffix	Sets a custom suffix that will be applied to the Confirmit login cookie (used for separating sessions on different sites on the same domain).
ConfirmitDeployer	NULL

ConfigName	ConfigDescription
ConfirmitURL	Specifies the URL for reaching the Confirmit installation on this site. Required by the Translation XML Import feature to validate translated XML files before upload.
ConfirmProgram	Specifies the path where Confirmit application files are installed.
ConfirmSecurity	NULL
Connect.Horizons2SFDC.Salesforce.x509	Connect Salesforce Application x509 certificate
Connect.Horizons2SFDC.SalesforceClientID	Connect Salesforce Application Client Id
Connect.Horizons2SFDC.SalesforceClientSecret	Connect Salesforce Application Client Secret
ConnectionTimeout	Specifies how long a survey can attempt to connect to the survey database before timing out. (Seconds)
ControlKey	NULL
CookieWarning	Cookie warning for the login page.
DashboardStayLoggedInTimeout	Indicates the lifetime of the permanent cookie for the Dashboard (in hours). Default value is 48
DatabaseCleanupBatchSize	Number of surveys deleted before sleeping.
DatabaseCleanupSleepTime	Sleep time (in seconds) between each batch.
DataCentralHorizontalMergeBuildIndexesLimit	Limitation on number of indexes built by DCS in the Horizontal Merge operation.
DataCentralHorizontalMergeMaxInputSurveys	Maximum number of input surveys used with the Horizontal Merge operation.
DataCentralHorizontalMergeMaxOutputSurveys	Maximum number of output surveys provided by the Horizontal Merge operation.
DataCentralHorizontalMergeRunQueryLimit	Time limit on a query run by the Horizontal Merge operation. (Seconds)
DataCentralMaxInputSurveys	Maximum number of input surveys to Data Central
DataCentralSurveySizeLimit	Maximum file size for one input survey. (MBytes)
DataProcessingTripleSMaxVariables	Determines how many variables can be used in Data Processing for Triple-S formats. (Number)
DataTransferRespondentLimit	Specifies how many rows of respondent data that should be

ConfigName	ConfigDescription
	transferred per cycle when using the Web Services.
DataTransferResponseLimit	Specifies how many cells of response data that should be transferred per cycle when using the Web Services.
DaysToKeepHubDatabaseObjectsAfterSoftDelete	Days before hub cleanup task will drop database objects for soft-deleted elements in hub designs.
DaysToKeepHubDesignObjectsAfterSoftDelete	Days before hub cleanup task will drop design model objects after they are soft-deleted.
DaysToKeepSurveyDatabasesForDeletedProjects	Specifices how long the survey database should be protected from permanent deletion on the database server after the project has been soft-deleted in the GUI (Days)
DebugStationInstallationPath	Sets the URL where Confirmit Debug Station can be downloaded from.
DedicateBatchServerToCompany	Dedicate batch server to specific companies (comma separated list of company ids).
DefaultDomainKeysCertificateIdentifier	Used by DomainKeys to add valid header info in outgoing emails signed with DomainKeys. (Certificate key)
DefaultDomainKeysSelector	Used by DomainKeys to add valid header info in outgoing emails signed with DomainKeys. (DomainKeys selector)
DefaultEmailBouncebackDomain	Determines the default email domain to use for sender addresses if the Email Delivery add-on is activated. Can be overridden on the company level.
DefaultOptimisticQuotaTimeout	Gets the default timeout (in minutes) for inactive interviews to be considered as expired and corresponding live achieved counters for optimistic quotas to be decremented. Default value is 5 minutes.
DefaultSqlCommandTimeout	Controls how long SQL commands/queries can execute before they will be aborted with a timeout. (Seconds)
DeployC	NULL
DeployN	NULL
DeployURL	Links generated in Confirmit will use this URL base unless any are overridden with the feature-specific links. Should be a FQDN prefixed by the http:// or https:// protocol, i.e. http://survey.confirmit.com, or using an IP address, i.e. http://192.168.1.1.
DesignerLogCleanupPeriod	The number of days to keep survey designer log entries.
DisableLegacyTableEngine	Hide table engine selection option in new reports, always use extended engine

ConfigName	ConfigDescription
DisplayEndLinkRedirectInformation	If turned on respondent will see information about upcoming redirect next to OK button and help link
DisplayLoginOverlayTimer	Specifies how long an inactive session is valid until the Login Overlay is displayed (Minutes)
DuplicateLogonEmail	Specifies recipient(s) of duplicate logon notifications. Multiple email addresses can be entered, using using the semicolon character as a separator between each address.
DuplicateUserDetection	When enabled, Confirmit will only allow one concurrent logon per username (unless they are from the same originating machine,) and send a notification email to the address(es) specified in the DuplicateLogonEmail field about the event.
ElevatedPermissionsList	The permissions that require 2-step authentication to be applied to a user/group/role
EmailFooter	This text is added as a footer to emails sent from Confirmit containing data. This does not apply to invitation nor to survey engine generated emails.
EmailPostfix	HTML tags to add in emails before user-generated content.
EmailPrefix	HTML tags to add in emails after user-generated content.
EmailRecipient	Speficies recipient(s) of Confirmit error messages. Multiple email addresses can be entered, using using the semicolon character as a separator between each address.
EmailSender	Exports from Confirmit tasks will be sent from this email address.
EmailSendoutLimit	Specifies the maximum number of emails that can be sent from a single email invite task from Confirmit Express. Default value is 100.
EmailServer	When enabled, all emails from Confirmit will be sent through the specified server. Valid formats are FQDN or IP address.
EnableMultimodeWebservicesIPLockdown	When enabled, will lock down access to Multimode Webservices to internal IP addresses.
EnableNotification	Enable Notification processing
EncryptionCertificateName	Database Encryption Certificate Name
EncryptSystemRequestParameters	When enabled, encrypts system parameters in respondent email URLs to prevent tampering.
EnforceSslOnSite	When enabled, all connections to pages on all servers in the site will be run over HTTPS.

ConfigName	ConfigDescription
ExceptionLogFile	NULL
ExceptionLogSource	NULL
ExceptionResourceBaseName	NULL
ExporterPadPrecodes	When enabled, ASCII file exports will be padded with "0".
ExportFileSizeLimit	Limits size (in MB) of generated export files.
ExportFixedWidthFileWithFixedMultiCodeAnswer	Controls the Answer Code value for multi questions in the fixed width definition file. Unchecked=Precode, Checked=1
ExpressExternalHelpUrl	Link to external help file or web site for Confirmit Express users
ExpressTrialInvalidDomains	Only applicable to Confirmit On-Demand environment - Specifies domains the user is not able to use for a registration email address.
ExpressTrialMaxEmails	Only applicable to Confirmit On-Demand environment - Specifies maximum number of emails a Confirmit Express user can email.
ExpressTrialMaxQuestions	Only applicable to Confirmit On-Demand environment - Maximum number of questions in a Confirmit Express survey.
ExpressTrialMaxSurveys	Only applicable to Confirmit On-Demand environment - Maximum number of surveys a Confirmit Express user can create.
ExpressUserActivationDisabled	When enabled, disables activation requirement for Confirmit Express users.
ExternalQuickTestURL	Link to quick test for External debug station users. Server must run Confirmit and have sticky sessions enabled, and the ExternalQuickTest role must be enabled.
FileLibraryEnhancedFileTypes	Specifies allowed file types for upload for companies with Enhanced file library setting. Companies with Enhanced setting will also inherit Standard file types.
FileLibraryMultimediaFileTypes	Specifies allowed file types for upload for companies with Multimedia file library setting. Companies with Multimedia setting will also inherit Enhanced and Standard file types.
FileLibraryStandardFileTypes	Specifies allowed file types for upload for companies with Standard file library setting.
FileLibraryUploadMaxFileSize	Sets a file size limitation on the file library upload function. Note: Whichever is lower of this setting and the MaxRequestLength in the "httpruntimesection" in the web.config file for Confirmit authoring will take precedence. (File size in KB)

ConfigName	ConfigDescription
FTPEnabled	Enables FTP option in Export/ Import parameters.
FTPRootPath	Configures the root UNC path of the FTP server. Used by Confirmit to find import and export files relevant to company IDs.
GenericSsoTimestampTimeout	Indicates how long a timestamp is valid when using generic SSO login. (Seconds)
GoogleAnalyticsExternalAccountId	Google Analytics tracker id for external pages (surveys etc)
GoogleAnalyticsInternalAccountId	Google Analytics tracker id for Confirmit (Authoring, Reportal etc)
GridviewMax	Specifies the maximum number of questionnaire nodes shown simultaneously in the gridview.
HideConfirmitMarketplace	When enabled, links to Confirmit Marketplace will not be displayed.
HierarchiesRestApiURL	Specifies the URL for the HierarchiesRestApi servers. This URL is used for all communication with the Hierarchies Rest Api.
HTMLEncode	Specifies whether question text, title and comment should be HMTL encoded to prevent XSS.
HubBulkLoadBatchSize	Number of rows in each bulk-loading batch during hub data load.
HubDatabaseSourceLimit	Number of sources allowed in a hub database.
HubMaxFieldsInCsiTables	Max number of fields in tables using Column Store Indexes.
HubMaxFieldsInVerbatimTables	Max number of fields in hub tables without Column Store Indexes.
HubResponsePartitionSize	Number of responses put into same database partition in the hub.
HubSparseStorageTreshold	Treshold on minimum numbers of fields in a variable before the variable is stored in sparse format in the hub.
IADatadownloadLimit	Amount of exported records in Instant Analytics
IdentityProviderEnabled	Determines whether Identity provider is enabled
ImageUploadMaxFileSize	Max survey image upload file size (Kb).
ImageUploadMaxHeight	Max image height (resize to).
ImageUploadMaxWidth	Max image width (resize to).
ImageUploadRootFolder	Image upload root folder (UNC).
ImageUploadThumbSize	Thumbnail image width (and height).

ConfigName	ConfigDescription
ImageUrlExpirationTime	Image URL expiration time (minutes).
IndexingEnabled	If the indexing is globally enabled.
IndexLocation	The default location for new index files.
IndexLocationThresholdItems	The new index creation threshold in indexed items.
InstallationID	Unique server name used to identify server in logs and error messages.
InstantAnalyticsReportMaxResponses	Max number of records allowed when creating or updating an Instant Analytics Report
InstantAnalyticsReportMaxVariables	Max number of variables allowed to create or update an Instant Analytics Report
InternalProxyCacheURL	Internal Proxy cache URL. Calls to internal REST services will use this for caching if specified
IPLockdownAllowedIPNetworks	Specify the list of IP ranges that will be allowed access to internal services(REST APIs). Single IP addresses, IP Ranges, CIDRs and combinations may be used, separating valid ranges with the comma character. Both IPv4 and IPv6 addresses are supported. Example: 10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16. Example: 192.168.1.50-192.168.1.150, 192.168.1.200, 172.16.1.7-172.16.1.24
IPLockdownEnabled	Enable lockdown of internal services(REST APIs) to only be accessible from clients with an ipaddress within one of the ranges specified in the IPLockdownAllowedIPNetworks config setting
IsServerInstallation	When enabled, users will have access to the Database source and target in PanelRule as well as exporting a panel rule to disk on the server instead of email.
KeepIndexingHistoryDays	The indexing history length in days.
KeepMaintenanceTaskHystoryDays	The maintenance task history in days.
KeepSearchHistoryDays	The searching history in days.
KeepSearchingResultMinutes	The searching result timeout in minutes.
LdapAuthenticationType	1 = Anonymous 2 = Delegation 3 = Encryption 4 = FastBind 5 = None 6 = ReadonlyServer 7 = Sealing 8 = Secure 9 = SecureSocketsLayer 10 = ServerBind 11 = Signing
LdapBypassUser	Specifies Confirmit user that can be used to bypass LDAP logon.
LdapLogOnEnabled	Enables LDAP directory authentication will be used when logging on to Confirmit. If disabled, normal Confirmit authentication will be

ConfigName	ConfigDescription
	used.
LdapNewUserCompany	Specifies which company in Confirmit new users authenticated by LDAP should be add to. Company must already exist in Confirmit.
LdapNewUserLanguage	Specifies new language users authenticated LDAP should have by default. Must be a valid language code.
LdapNewUserLevel	Specifies "level" of the Confirmit user that is created during LDAP logon. Must be a valid level code.
LdapNewUserRole	Specifies "role" of the Confirmit user that is created during LDAP logon. Must be a role that has already been defined in Confirmit, if specified.
LdapNewUserType	Specifies "type" of Confirmit user that is created during LDAP logon. Must be a valid type code.
LdapPath	Path to LDAP server, starting with ldap://
LdapServerType	Directory server type. 1= MS Active Directory 2 = Open LDAP
LdapSyncPassword	When enabled, the Confirmit user will be given the same password as the LDAP directory user.
LdapSyncUserSettings	When enabled, the user settings (first name, last name, email) will be kept in sync with the values found in the LDAP directory.
LdapUserDomain	Specifies the domain of the directory user. Not always required depending on server type.
LdapUserEmailAttribute	Specifies directory attribute that is used to provide the email address of the Confirmit user that is created during LDAP logon.
LdapUserFirstNameAttribute	Specifies directory attribute that is used to provide the first name of the Confirmit user that is created during LDAP logon.
LdapUserIdAttribute	Specifies the directory attribute that is used when searching for user in directory.
LdapUserLastNameAttribute	Specifies directory attribute that is used to provide the last name of the Confirmit user that is created during LDAP logon.
LoaderMaxCommandTimeoutMinutes	Command timeout in minutes for potentially long-running commands used by Hub loader.
LoginKeyTimeout	Specifies session validation length for web service users. Also controls how long users can be idle before the Login Overlay is displayed. (Minutes)
LoginSysAdminToConfig	If enabled, the system configuration page will be displayed after each logon if the logged on user has the

ConfigName	ConfigDescription
	SYSTEM_ADMINISTRATE permission set.
LogThresholdBitStreamQueryTime	Threshold for logging BitStream queries. Unit is seconds.
LogThresholdFastObjectsQuery	Threshold for logging FastObjects queries. Unit is seconds.
LogThresholdHttpCollectionKeys	Log threshhold for number items in HttpCollectionKeys.
LogThresholdQueriesPerRequest	Threshold for logging number of SQL queries per request. Unit is number of queries.
LogThresholdQueryFrequencyPerTask	Threshold for logging query frequency per task. Unit is queries/second.
MaxAggregatedTables	Determines the maximum number of aggregated tables that can be inserted into a Reportal report page.
MaxAnswersAllowedForChart	Maximum number of answers allowed for questions used as source for Chart objects in questionnaire.
MaxBitStreamIncrements	Determines the maximum number of incremental updates that can be made to a BitStream file set before a new full file set must be created.
MaxConcurrentThreads	Specifies the number of concurrent threads that can run on one server. Default value is 1. Currently only used by Confirmit BitStream service.
MaxDynamicCategories	The maximum number of categories to be created for expanded Open Text or numeric questions in aggregated tables
MaxFileSize	Sets the maximum allowed (precalculated) file size for exports.
MaxLogonAttempts	Defines how many subsequent unsuccessful logon attempts will be allowed before the user is locked out.
MaxNumberOfEmailsPerTask	Maximum number of emails allowed per Batch Emailing task
MaxNumberOfRowsInMultivariateStatistics	Maximum number of rows allowed in Multivariate Linear Regression and Factor Analysis
MaxPoolSize	The maximum number of connections allowed in the pool used by survey engine to connect to survey databases. Default value is 100.
MaxReportalExportSize	Controls the maximum number of exports which are allowed in a single export order. NOTE: The amount allowed is the total number of hierarchy levels which would be generated from the export request. A single hierarchy level might be generated more than once if other languages than the export language are required. This would be the case if the "Send to Endusers" checkbox in the export dialogue has been checked and an enduser with this hierarchy level uses a different language than the export

ConfigName	ConfigDescription
MaxRequestLengthForWebServices	The maximum number of request length to web service.
MaxSegmentsInCombinedDimension	Sets the maximum number of allowed segments when combining multiple dimensions in a matrix. Default value is 3000.
MaxSurveyVersions	Number of runtime filess versions produced during survey launch that are stored. Older versions of such files are used to ensure that a respondent is not affected in the middle of taking an interview by a survey launch even when the definition of the new survey has changed. This setting can be overridden at a per company level.
MaxXmlSurveyVersions	Number of versions of the survey definition that are stored. Old survey definitions are used to support versioning, ie. ability to compare the current design mode survey definition with the survey definition at the time of previous survey launches. This setting can be overridden at a per company level.
MaxZipFileSize	Sets the maximum allowed file size for an export after compression.
MessageBrokerHost	Message broker endpoint. Include port using colon if other than default (5672). Separate nodes in cluster with comma.
MetadataApiMonitorEnabled	Enable logging of all Metadata API requests to MetadataApiMonitor
MetaDataRestApiURL	Specifies the URL for the MetaDataRestApi servers. This URL is used for all communication between Confirmit servers and the MetaData Api.
MinutesIdleTimeBeforeDetachingHubDatabase	Minutes of idle time for a hub database before it can be detached while running the Hub Cleanup task.
MobileDashboardBitStreamUpdateLimitMinutes	How often users are allowed to update BitStream files in the Mobile Dashboard (Report screen)
MultimodeBaseURL	The url to be used for accessing the CATI management component. The full URL to the location of the server hosting the CATI management components should be supplied. I.E http://server/cp/
MultimodeCAPIConsoleDownloadURL	Used in the CAPI - Download CAPI Console page to display and optionally email the url to users. I.E. http://server/download/capiconsole/
MultimodeCATIConsoleDownloadURL	Used on the CATI - Download CATI Console page to display and optionally email the url to users. I.E. http://server/download/caticonsole/
MultimodeCATISupervisorPlayerDownloadURL	URL for downloading the CATI supervisor console. I.E. http://server/download/catisupervisorconsole/
MultimodeWebServiceURL	The full URL to the location of the Web service used for CATI management. I.E http://server/BVFMWS/Service.asmx

ConfigName	ConfigDescription
NewsEmailBodySuffix	Text suffix used in news email body.
NewsEmailSubjectPrefix	Text prefix used in news email subject.
NodeStorageIndexingEnabled	If the indexing of changes in the node storage is enabled.
NodeStoreExpiry	Number of days before expiry from NodeStore
PanelPortalSlidingExpiration	Specifies how long a Panel Portal will remain in server memory after last being accessed.
PasswordExpiryDays	Number of days before password expires. Default value is 0 (never expires).
PasswordHistory	Controls how many previous passwords Confirmit will remember for each user (preventing the user from reusing any of those passwords).
PasswordLinkTimeout	The number of minutes a passwordlink should remain valid after it has been generated
PasswordMinimumAge	The user will have to wait x minutes after changing password before being allowed to change it again.
PasswordMinRequiredLength	Minimum password length (overrides company password validation script).
PasswordMinRequiredNonAlphaCharacters	Minimum required non-alpha characters.
PasswordMinRequiredNonAlphanumericCharacters	Minimum non-alphanumeric characters in password required.
PasswordMinRequiredUppercaseCharacters	Minimum required uppercase characters in password.
PasswordStrengthRegularExpression	Regular expression for validating password strength (overrides company password validation script).
PdfBatchServerIPs	Specify IP addresses of batch servers with ActivePDF installed (servers that are allowed to access Reportal for export purposes). Separate multiple entries with semicolon (i.e. 192.168.0.10;192.168.0.11)
PdfDebug	Enable debug mode in PDF exports for verbose logging. This will affect PDF engine performance and should therefore only be enabled for short periods during troubleshooting.
PdfPort	Specifies port on which ActivePDF is listening on the batch servers.
PdfReportalServer	Specifies IP address/FQDN of the Reportal server ActivePDF

ConfigName	ConfigDescription
	should use when running export from (i.e. reportal.confirmit.com).
PDIConnectionTimeout	NULL
PDIEnabled	NULL
PDINotificationEmails	NULL
PDIProbingTimeout	NULL
PGPEnabled	Enables PGP encryption as an option for exports/imports (Requires addon).
PGPPassPhrase	If PGP is enabled, the passphrase for the private key on the server should be set here.
PortalRememberMeLoginExpiration	Determines how long user will be logged on in case he checked 'Remember Me' (in minutes)
ProjectInfoCollectorTracing	Enables trace logging when running Project Info Collector tasks. Used for debugging purposes.
ProjectListResponseLimit	Maximum number of projects returned at a time on a project search using Web Services. If the result of a query returns more hits than the ProjectListResponseLimit all the hits can be returned, however split into several parts and controlled by use of tokens.
PublishDataBlockSize	Determines the block size for Reportal report publisher data (publishing large reports can time out if the block size value is too large)
PushServiceRestApiUrl	Sets the base URL of PushService Reset Api.
QueryTimeout	Sets a limit for how long a query can run in online reporting before the server times out. (Seconds)
QuestionnaireReviewerLinkExpiryDays	Specifies how long questionnaire reviewer links should be valid for. (Days)
QuotaGridMaxNumberOfCells	The maximum number of cells allowed for a quota grid. If the grid is larger than this number, the page will not open (a warning is shown.)
QuotaListAddAllMaxNumberOfRows	The maximum number of rows the Add All button will generate in the quota list. If the rows exceed this number, the "add all"; will not be performed.
QuotaListMaxNumberOfColumns	The maximum number of columns for the quota list mode. If the quota exceeds this number, some columns will automatically be hidden.
RapidResultsTableLimit	Maximum number of tables displayed in Rapid Results.

ConfigName	ConfigDescription
RapidResultsTimeout	Determines how long a printer friendly/excel export in Rapid Results can run before it is aborted with a time out. (Seconds)
RdgMaxResponses	Limits the maximum number of responses from the Random Data Generator.
RdgMinResponses	Limits the minimum number of responses from the Random Data Generator.
RdgSleepTime	Defines for how long the Random Data Generator should pause before generating a new response. (Seconds)
RealtimeStatusingDisabled	When enabled, surveys linked to panels will not update surveyhistory status in the panels in realtime. Instead, a panel rule must be used for these updates to be performed periodically.
RedisConnectionString	Redis URL. To turn off the redis listening clear the setting
RegisterSmtpEventSinks	Register SMTP Event Sinks (should be unchecked if Confirmit MTA notification service is to be used)
RemoveClientScripts	When enabled, scripts in surveys will not be parsed during launch.
ReplicatedVariablesMaxFieldWidthOfOpenText	At launch time CATI enabled surveys that contain open text variables that have the 'Available as CATI filter' property checked are verified that they do not exceed this maximum width value.
ReplicatedVariablesMaxNumber	At launch time for CATI enabled surveys, a count of all variables that have the 'Available as CATI filter' is performed, this count cannot exceed the supplied value.
ReportalCopyrightText	Specifies text to show at the bottom of Reportal report pages.
ReportalDisableEmailBranding	When enabled, Reportal report export emails will not include Confirmit branding.
ReportalEndUserURL	Sets the public URL used for emailing end users Reportal links. The URL should be in the format http://<reportalserver>/Reportal/login.aspx.
ReportalEventLogEnabled	Reportal eventlog logs actions performed by users in Reportal and makes it possible to track changes to a report
ReportalHitListTimeout	Controls search timeout in Reportal hitlists (Seconds).
ReportalPPTAddInDownloadURL	Url to download PPTAddIn
ReportalScriptCPULimit	Timeout control for scripting in Reportal (CPU time in milliseconds).
ReportalURL	If Reportal is running on a different server than Authoring, this value can be used for allowing users to open Reportal directly from Authoring without having to log on twice. Single sign-on requires

ConfigName	ConfigDescription
	that the SetConfirmitCookieDomain setting is enabled, and the servers must reside in the same domain (i.e. confirmit.com.) The value should link directly to the Reportal: http://<reportalserver>/Reportal/default.aspx
ReportalUseCachingService	If enabled, Reportal will cache reports in order to improve performance. Requires that the Confirmit Caching Service is running and configured correctly.
ReportC	NULL
ReportingActivityOverviewTimeout	Used to limit the time a request of activity overview executes towards database (Seconds).
ReportingApiURL	ReportingAPIURL
ReportingSqlServerId	The unique reporting database server ID number. The first should be "1", second "2", etc.
ReportingSqlServerWeight	Indicates weight ratio for reporting database deployment ratio on database servers (used in multi-database server systems). Value "0" indicates that no new reporting databases are created on the server. Weights for multiple database servers if set differently on the server level are relative to each other. Out of (Sum weight across all servers), (Weight on server x) number of databases are created on server x.
ReportingTimeout	Used to limit the time a report executes individual queries towards the survey database (Seconds).
ReportN	NULL
ReportPackageSlidingExpiration	Specifies how long a Reportal report will remain cached in server memory after last being accessed (Minutes).
RestApiURL	Specifies the root URL for all the REST APIs. This is used as the base path for all APIs
Root Path	Sets the root UNC path for deployment of files from remote servers, normally \\<servername>\confirmitdatashare\web. Used by batch servers to deploy files to servers belonging to specific roles (i.e. production survey packages to deployment servers.), and for distributing files to file library.
SamplingLoggingEnabled	This setting enables logging to file to disk for sampling tasks (applies to BitStream files for sampling, coffey reports and panelist weights.) This setting replaces the "DumpBitStreamToFile" setting which only applied to sampling BitStream files.
SchemaCacheTimeout	Specifies how long an object will remain in the MetaDataCache (Seconds).

ConfigName	ConfigDescription
SearchingAndIndexingServiceUrl	The Searching & Indexing service Url.
SearchingEnabled	If the searching is globally enabled.
SelfCompletionAndroidAppMandatoryUpdate	When checked, app users will not be able to synchronize with an app version lower than the SelfCompletionAndroidAppMinVersion.
SelfCompletionAndroidAppMinVersion	This is the minimum app version that is supported. Updates to at least this version can be enforced by checking SelfCompletionAndroidAppMandatoryUpdate.
SelfCompletionIosAppMandatoryUpdate	When checked, app users will not be able to synchronize with an app version lower than the SelfCompletionIosAppMinVersion.
SelfCompletionIosAppMinVersion	This is the minimum app version that is supported. Updates to at least this version can be enforced by checking SelfCompletionIosAppMandatoryUpdate.
ServiceDiscoveryEnabled	Enables usage of Service discovery service on the site.
ServiceDocumentationEnabled	Enables the service documentation for all the REST APIs on the site.
SetConfirmitCookieDomain	Used to allow single sign on to Reportal for authoring users if Reportal is running on a separate server, and link to Reportalserver uses same DNS domain address as the authoring server (i.e. author.confirmit.com and reportal.confirmit.com). This setting is required if multimode (CATI) is enabled.
SharedData	Sets the path to the shared data area where the Confirmit Task System service picks up files for importing, and Confirmit servers looks for shared data files (UNC path).
ShortURL API URL	Address for the short URL service
ShortURL Redirect URL	Domain name or URL with redirect service
ShowHelpVideos	When enabled, clickable video icons will be displayed where help videos are available (Note: Help videos must be downloaded separately, and made available under /HelpVideos on the web server.)
ShowSiteIdInTitle	Show the site ID in the client browser title bar.
ShrinkDatabaseFreeSpaceTreshold MB	Treshold of free database space in MB before hub loader should do a database shrink.
SinglePageSurveyCleanupPeriod	The number of days to keep generated single page surveys after last access.
SingleSignOn	Specifices whether the server should use Single Sign On for authentication (requires SSO configuration.)

ConfigName	ConfigDescription
SiteID	Sets the Site ID for identifying the current site. (Note: not related to the SiteID used in the license file.) The value will be displayed in the lower right corner of Confirmit, and will also be used for identifying the site when sending periodic transaction logs to Confirmit.
SMTPServerPickupDirectory	Used to specify where new emails should be created so the local SMTP service can find the emails and send them.
SPSSDataModelInstalled	Should be activated if SPSS Data Model is installed on server. Used by DataCentral projects. Must be activated to be able to upload a DC project that has SPSS operations.
SQLArchivingDataPath	Specifies UNC path for survey database data (mdf) files on the database server that can be reached from the batch server running the Archiving task (required only if Archiving is enabled).
SQLArchivingLogPath	Specifies UNC path for survey database log (ldf) files on the database server that can be reached from the batch server running the Archiving task (required only if Archiving is enabled).
SQLAuditExcludedPrincipalNames	Comma separated case insensitive list of principal names to exclude audit.
SQLAuditFolder	Specifies UNC path to sql audit folder.
SQLEngine	NULL
SQLServerDataPath	Specifies the local target path for new survey database data files on the database server.
SqlServerFilestreamPath	Specifies path for survey database FILESTREAM files.
SqlServerId	The unique database server ID number. The first should be "1", second "2", etc.
SQLServerLogPath	Specifies the local target path for new survey database log files on the database server.
SQLServerName	The unique database server name. The server name must be accessible by the application servers as a NetBIOS name, an IP address, or a SQL Server client alias.
SQLServerUsedForEncryption	Check to use this SQL instance to store encrypted databases.
SSISColumnLimit	Limits initialization time of SSIS packages.
SSISRuntimeVersion	Hints about main version of SQL that SSIS is running towards. 5=SQL2016, 6=SQL2017.
SSLAcceleratorMode	Enables SSL accelerator mode in Confirmit. Should ONLY be enabled if SSL certificates are loaded onto an external SSL accelerator rather than using local certificates installed on servers. Enabling this setting without proper load balancer configuration will

ConfigName	ConfigDescription
	interrupt access to the site, including access to revert this setting! Consult the Confirmit Server Manual before enabling this setting.
SSLAcceleratorPort	Used along with the SSLAcceleratorMode, this setting specifies which port on the server the SSL accelerator should forward decrypted requests to. Do not configure this unless using an external SSL accelerator. Also see separate chapter on SSL Acceleration in the Confirmit Server Manual.
SSLAuthorCertificate	When enabled, Confirmit logons will be encrypted via SSL (requires SSL certificate installed on the server(s) where authoring users log in.
	Note that if SSLAcceleratorMode is enabled, the SSL certificate only needs to be installed on the SSL accelerator).
SSLEnduserCertificate	When enabled, SSL may be used for end users (requires SSL certificate installed on the server(s) where endusers log in.
	Note that if SSLAcceleratorMode is enabled, the SSL certificate only needs to be installed on the SSL accelerator).
SSLSurveyCertificate	When enabled, survey links will include an SSL option, also enables support for the ^secureslink^ primitive in survey invitation emails (requires SSL certificate installed on the server where respondent accesses surveys.
	Note that if SSLAcceleratorMode is enabled, the SSL certificate only needs to be installed on the SSL accelerator).
SSLWebServiceCertificate	When enabled, SSL may be used when calling the Web Services (requires SSL certificate installed on the server(s) where the APIs are installed.
	Note that if SSLAcceleratorMode is enabled, the SSL certificate only needs to be installed on the SSL accelerator).
SurveyDataAPIURL	Specifies URL to SurveyData API if used on a separate server.
SurveyDeploymentThroughDatabase	Enables deployment of surveys through SQL database rather than deploying files from the batch server. Can be used on systems where file deployment is not allowed between servers. Note: Can not be reverted, consult the server documentation before enabling this setting.
SurveyEngineLogThresholdHttpCollectionKeys	Log threshhold for number items in HttpCollectionKeys for the survey engine.
SurveyExport70Enabled	NULL
SurveyLoginPageLinkTimeout	A setting for controlling the timeout (in minutes) of links in limited login surveys.

ConfigName	ConfigDescription
SurveyN	NULL
SurveyPackageSlidingExpiration	Specifies how long a survey package will remain in server memory after last being accessed. (Minutes)
SurveyPageCPULimit	Controls how much CPU time a script in a survey is allowed to use before it is aborted. Can be overridden on per-server level to set different limits for servers. The rolewide setting is shared between the Authoring, Deployment and ExternalQuickTest roles. (CPU time in milliseconds)
SurveyPageJavaScriptCPULimit	Controls how much CPU time a script in JavaScript language in a survey is allowed to use before it is aborted. Can be overridden on per-server level to set different limits for servers. The rolewide setting is shared between the Authoring, Deployment and ExternalQuickTest roles. (CPU time in milliseconds)
SurveyRouterGetAvailableSurveyRetryAttempts	The number of retry attempts to find an available survey to be routed to when previously selected surveys have already been participated in.
SurveyRouterMaxChainLength	The maximum number of surveys that can be chained together using survey router routing for a single respondent.
SurveyRouterMaxNoOfSurveys	The maximum number of surveys that can be registered in a single Survey Router.
SurveyRouterMaxQuestionsForSurveySelection	The maximum number of questions that can be supplied as criteria for survey selection.
SurveyRouterMaxRegisteredQuestions	The maximum number of questions that can be specified for registration in a Survey Router.
TaskAbortedSender	If a scheduled Confirmit task fails, and the user that set up the task has requested to be notified when tasks fail, this address will be used as the sender address for the notifications. (Email address)
TaskExecutorPort	The port used to host the WebApi within the Task Executor Host.
TaskQueueTimeout	The time (in minutes) before an assigned task with status queue is un-assigned.
TaskSchedulerLoadActionManagementSourceTimeInterval	Time (in seconds) determining how often the Task Scheduler service should search for changes and schedule load tasks for Action Management Program sources.
TaskSchedulerLoadCompanyCalendarTimeInterval	Time (in seconds) determining how often the Task Scheduler service should search for changes and schedule load tasks for company calendars.
TaskSchedulerLoadDbDesignerSchemaSourceTimeInterval	Time (in seconds) determining how often the Task Scheduler service should search for changes and schedule load tasks for db designer schemas.

ConfigName	ConfigDescription
TaskSchedulerLoadProjectSourceTimeInterval	Time (in seconds) determining how often the Task Scheduler service should search for changes and schedule load tasks for project source data.
TaskSchedulerTryGetLockTimeInterval	Time (in seconds) determining how often an instance of the Task Scheduler service should try to obtain an exclusive lock.
TaskSchedulerUpdateLastActivityTimeInterval	Time (in seconds) determining how often an instance of the Task Scheduler service should indicate that it holds an exclusive lock.
TextAnalyticsBatchSize	Number of records that will be transferred in one batch/task.
TextAnalyticsIntellegoRestApiBaseUrl	The base URL to be used for accessing the Text Analytics REST API
TextAnalyticsWorkflowRestApiUrl	Text Analytics Workflow REST API URL
TouchedResponsesReloadTreshold	Number of touched responses that triggers full Hub reload instead of incremental load.
TranslatorUrl	If the Confirmit Translation addon is enabled, and the module is running on a different server than the authoring feature, this value can be used to override the ConfirmitURL value. If left blank, the ConfirmitURL will be used as the base URL for Translator as well
TripleSRecordSizeLimit	Limits size (in characters) of a triple-s data record.
UniqueSiteIdForTextAnalytics	Sets the unique Site ID for identifying the current site when pushing data for Text Analytics processing. Note! Do not change once data have been pushed.
UseIERendererForPdfExports	Whether PDF exports on this server should use the old IE engine rather than the new x64-compatible WebKit engine
UserAgentRegExDesktop	Regular expression determining user agent of the client to be a desktop computer
UserAgentRegExTouchDevice	Regular expression determining user agent of the client to be a compatible touch device browser.
ValidIpRangeForInternalWebServices	Limit access to the Internal Web Services to a predefined IP range. Single IP addresses, IP ranges, or combinations may be used, separating valid ranges with the comma character. Example: 192.168.1.50-192.168.1.150,192.168.1.200,172.16.1.7-172.16.1.24.
ValidIPRangeForSysAdmin	Limit admin to accounts with system_administrate to a predefined IP range. Single IP addresses, IP ranges, or combinations may be used, separating valid ranges with the comma character. Example: 192.168.1.50-192.168.1.150,192.168.1.200,172.16.1.7-172.16.1.24.

ConfigName	ConfigDescription
VideoUploadRootFolder	Video upload root folder (UNC).
WebServiceBaseUrl	The base URL to be used for accessing the Confirmit Web Services on this site (e.g. http://ws.confirmit.com).
Weight	Indicates weight ratio for survey database deployment ratio on database servers (used in multi-database server systems). Value "0" indicates that no new survey databases are created on the server. Weights for multiple database servers if set differently on the server level are relative to each other. Out of (Sum weight across all servers), (Weight on server x) number of databases are created on server x.
wwwroot	NULL

Appendix B - Installing and Configuring RabbitMQ

Use this guide in addition to the official windows installation guide that can be found here:
https://www.rabbitmq.com/install-windows.html

Note: RabbitMQ is a third party component required for Survey Designer that will be used by other roles in the future. RabbitMQ is not supported by Confirmit. However to aid installation the following guide has been created.

Download Files

Description	Download
Installer for Windows systems (from rabbitmq.com)	https://www.rabbitmq.com/install-windows.html (Version 3.7.7)
Erlang Windows Binary File	http://www.erlang.org/download.html (Version 21.0.1)

Note: Verify the supported version to download by using the pre-requisite checker.

Setup

Before proceeding uninstall any previous versions of RabbitMQ

Installing the Server

1. Download and install the Erlang Windows Binary File.
2. Run the installer, **rabbitmq-server-<version number>.exe**.

   The installer will set RabbitMQ up and running as a service, with a default configuration.

Customize the RabbitMQ Environment

If you need to customize names or locations, it is easiest to configure environment variables in the Windows dialogue: **Start > Settings > Control Panel > System > Advanced > Environment Variables**. Then create or edit the system variable name and value.

Note: For environment changes to take effect on Windows, the service must be re-installed. It is not sufficient to restart the service.

Add Environmental Variables

Add the following environmental variable:

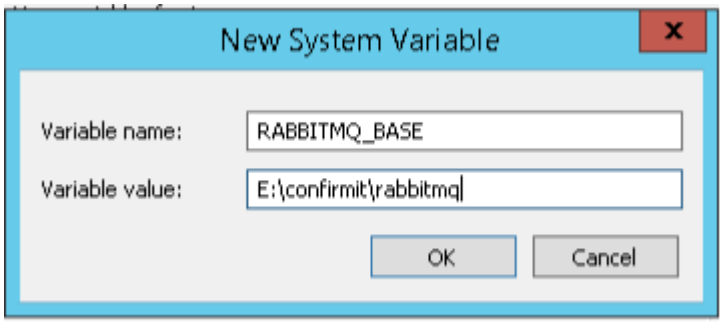


Figure 53 The New System Variable dialog

Name	Default value	Suggested value
RABBITMQ_BASE	%APPDATA%\RabbitMQ	E:\Confirmit\RabbitMQ (Suggested path, try to avoid keeping RabbitMQ data on system drive)

Add RabbitMQ Installation Directory to PATH Variable

Add **C:\Program Files\RabbitMQ Server\rabbitmq_server-<version number>\sbin**, or where RabbitMQ was installed in the previous step, to the Path variable.

This will give you access to RabbitMQ command line tools from any folder.

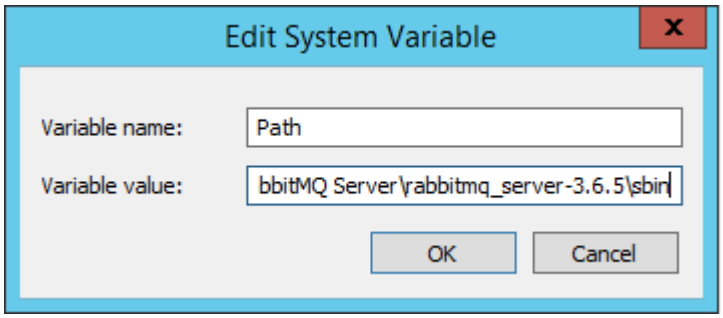


Figure 54 The Edit System Variable dialog

Configuration File

1. Create the RABBITMQ_BASE folder (E:\Confirmit\RabbitMQ or the same value as in the previous step).
2. Create a file called **enabled_plugins** inside the folder with the following content:

```
[rabbitmq_management,rabbitmq_management_agent,rabbitmq_shovel,rabbitmq_shovel_management,rabbitmq_tracing].
```

For more information on these plugins, read https://www.rabbitmq.com/plugins.html.

Remove and Reinstall the RabbitMQ Service

The following commands must be run in the command prompt or powershell, running with administrator permissions.

Note: If you have not added the rabbitmq.sbin folder to your Path variable, you need to CD into the sbin folder to access the "rabbitmq-service" command line tool.

```
rabbitmq-service remove
rabbitmq-service install
```

Wait for 30 seconds before executing next command.

```
rabbitmq-service start
```

Clustering

Note: Skip this section if you are not going to run a clustered RabbitMQ environment.

To create a cluster of RabbitMQ servers for performance and high availability, follow the following steps in addition to the previous steps on all servers.

Environment Variables

Add the following system environment variables (**Start > Settings > Control Panel > System > Advanced > Environment Variables**):

Name	Default value	Suggested value
RABBITMQ_CTL_ERL_ARGS	None	-setCookie 'xxxxx'

Important
This cookie value is an arbitrary string value that must be the same on both variables on all the servers that will join the cluster.

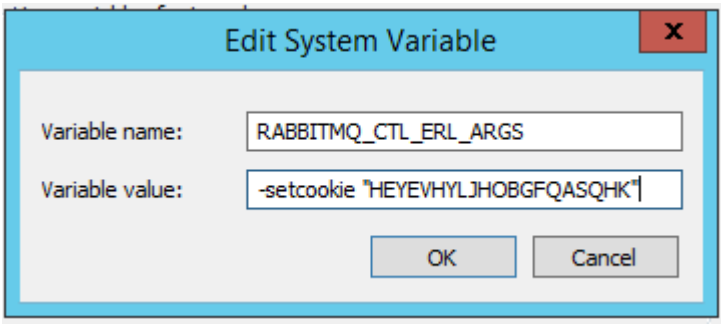


Figure 55 Example of the variable

Erlang Cookie File

RabbitMQ service will read this erlang cookie value from a file that is stored in

```
C:\windows\.erlang.cookie
```

If there is no file present, RabbitMQ will generate a file with an arbitrary string. This file is set to read only by default. Unticking the read-only option in the file properties will allow editing of this file. The text editor needs administrator privileges to edit this file.

Set the contents of the ".erlang.cookie" file to be the same on all nodes you want in the cluster, as well as in the environment variables from previous step.

Restart the RabbitMQ service to pick up any changes to the erlang cookie.

Joining the Cluster

```
Run the following command and note down the nodename on the first server
rabbitmqctl status
```

This command will return the following

```
Status of node 'rabbit@123456-SERVER1' ...
```

Log on to all the other servers that you want jo join into a cluster and run these commands on all of them:

```
rabbitmqctl stop_app
rabbitmqctl join_cluster 'rabbit@123456-SERVER1'
rabbitmqctl start_app
```

Verifying the Configuration

Note: The RabbitMQ service must be running at this point, and you must have enabled the management GUI by creating the file called enabled_plugins.

Login to the web management GUI interface at http://localhost:15672 on any of the servers. Log in with default credentials:

Username: **guest**

Password: **guest**

Note: The guest user can only login from localhost. You may disable the user or change the password in the next section.

Once you login you will see all your cluster nodes listed:

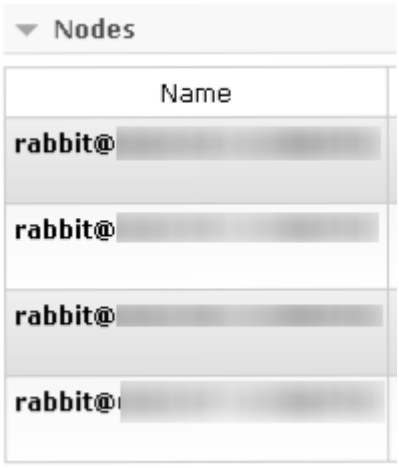


Figure 56 Example of the list of cluster nodes

RabbitMQ Users

This section describes creating and setting up users.

Logging In to the Management Interface

Note: The RabbitMQ service must be running at this point, and you must have enabled the management GUI by creating the file called enabled_plugins.

Login to the web management GUI interface at http://localhost:15672 on any of the servers. Log in with default credentials:

Username: **guest**

Password: **guest**

Note: The guest user can only login from localhost. You may disable the user or change the password in the next section.

Creating a User

Navigate to the **Admin** menu, and scroll down to the **Add a user** section.

Figure 57 The Add a user fields

After you have created the user you will see it in the users list, but it will have no rights yet. Click the username and then click the **Set permissions button** to give the user permissions.

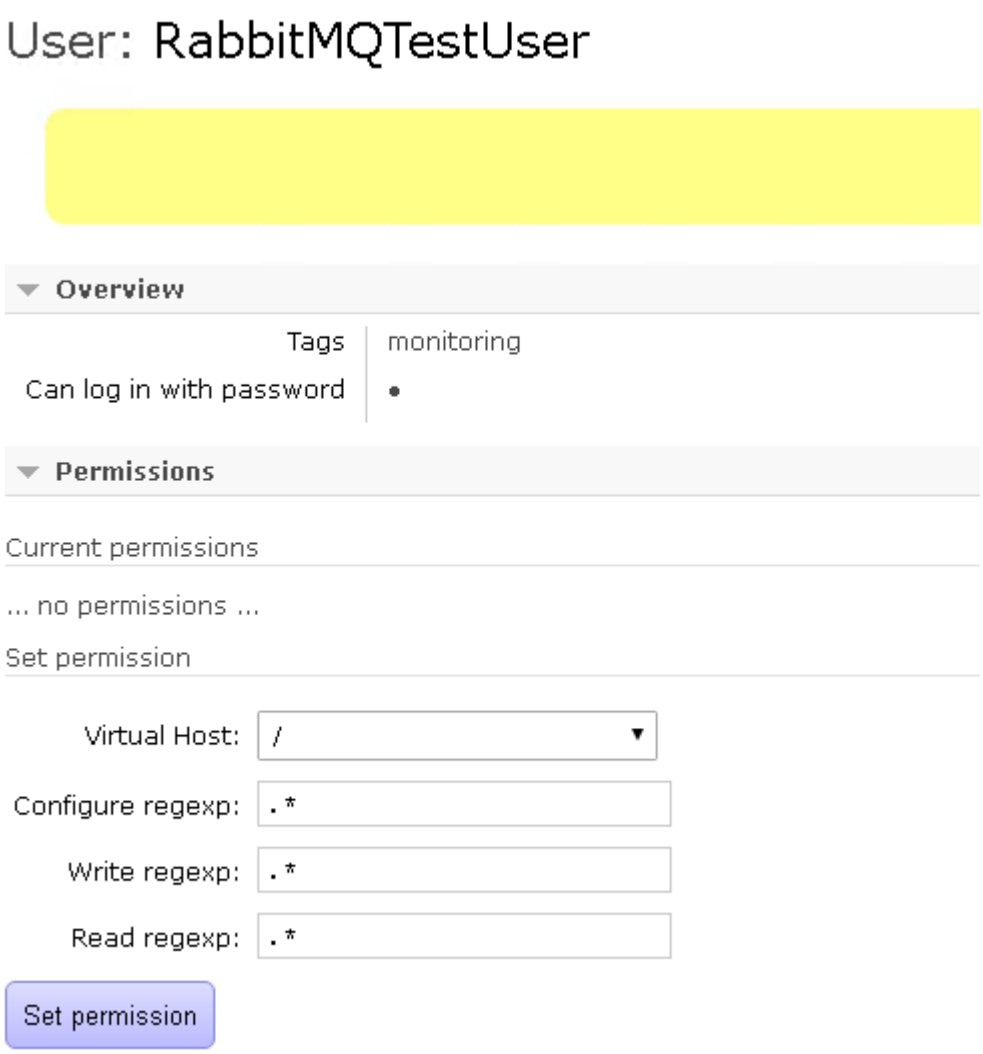


Figure 58 Setting the user's permissions

Note: If you have set up a RabbitMQ cluster, you only need to create this user on one of the nodes; it will replicate out to the others.

Populating Octopus Variables

Navigate to your Octopus installation and add two variables called **Confirmit.MessageBroker.User.Name** and **Confirmit.MessageBroker.User.Password**, and set the values to the user you created in the previous steps.

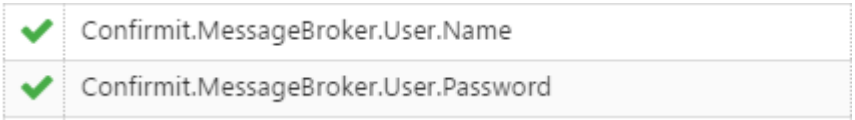


Figure 59 The two variables

Note: Remember to create a new snapshot of the Octopus variables for the current release to fetch the new variables.

After you have created the new snapshot of the Octopus variables, the Server Bootstrapper step in Octopus must be force redeployed for the configuration changes to take effect.

System Configuration

For Confirmit Horizons to know where the message broker servers are, the connection string must be added to the system configuration in Professional Authoring.

1. Log in to Professional Authoring as an admin user.
2. Go to the **Admin > System Configuration** menu.

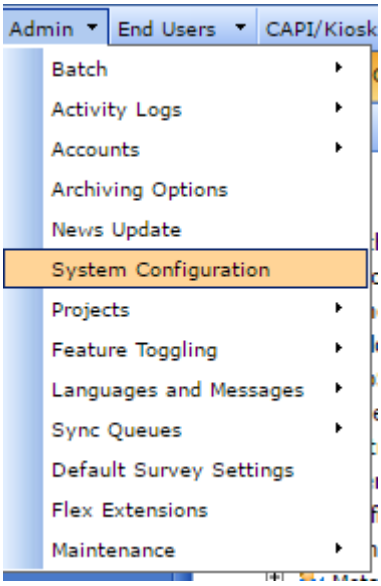


Figure 60 Accessing the System Configuration menu

3. Scroll down to the Message Broker section and add the hostnames of the hosts running RabbitMQ. If you have a cluster, separate the hostnames with a comma.

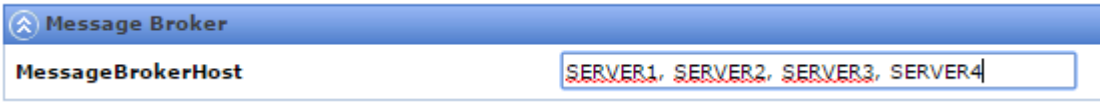


Figure 61 Adding the hostnames

Appendix C: Octopus Roles and Variables

This is a reference list of the various Confirmit Horizons Roles, and the available Octopus variables. The same information is available when running through the Confirmit Horizons Octopus setup process.

Confirmit Server Roles

Role name	Required/Optional	Description
Authoring	Required	Professional Authoring UI, SurveyEngine Test Modes, Survey Dashboard, Panel UI and other client applications.
Batch	Required	TaskSystem and Microservice Agents. Usually installed on one or more standalone machines.
BitStreamServer	Required	Usually installed on one or more standalone servers or machines with the Batch role.
CachingService	Required	Caching service. Usually installed on machines with the Authoring role.
CapiSynchronization	Deprecated	This role is not used anymore and can be removed.
ClickOnce	Optional	Only required for CAPI. Used to host CAPI download files. Usually installed on machines with the Authoring, Reportal or Deployment role.
Deployment	Required	Surveys, Short Url redirects and Panel UI.
Digital Feedback	Optional	New Digital Feedback service.
DiscoveryAnalytics	Optional	Must be installed on machines with the Reportal role. Requires Hub.
ExternalQuickTest	Optional	Support for external quick test. Should be installed on machines that have sticky sessions like the Authoring role.
HierarchyManagement	Optional	Must be installed on machines with the Authoring role. Requires Hub.
HubUI	Optional	Must be installed on machines with the Authoring role. Required for Hub.
IdentityService	Required	Identity Service used for authentication of users and applications. Can be installed on standalone servers.
MetadataApi	Required	Microservices with internal API's.
PowerMTA	Optional	Only used if customer has purchased the PowerMTA software and Confirmit Add-on.

Role name	Required/Optional	Description
Reportal	Required	Reportal and reporting services.
SearchService	Required	This role should be assigned to only ONE machine per environment. Install on standalone machine or together with SiteBootstrapper role.
SiteBootstrapper	Required	This role should be assigned to only ONE machine per environment. Used to run deployment steps that only execute on one machine. e.g. Confirmit system database(s) creation/update and updating of site config values specified as Octopus variables. Also used for Default Templates.
SmtpEventSink	Optional/Required	Required if not using a 3rd party email service like PowerMTA. Add this role to all machines that also have the Batch role.
Studio	Optional	New Studio UI. Must be installed on machines with the Authoring role.
SurveyDesigner	Optional	New Survey Designer UI. Must be installed on machines with the Authoring role.
WebService	Required	Used for public web services and API's. Some of these services and API's are also accessed from internal Confirmit applications and services.

Confirmit Octopus Variables

These variables are controlled via Octopus. Generally, Octopus will deploy these variables and overwrite conflicting settings on the site. You can test a setting by configuring it directly on the site, and then make it persistent by updating the Octopus variable.

Important
If you change a variable here, you will need to update the snapshotted variables for the release and force redeploy of the packages. If not, the updated variable will be persisted when the next release is loaded onto the Octopus server.

Name	Example Value	Explanation
Confirmit.AppInit.Enabled	TRUE	Sets whether or not the deployment script will try to start up (prewarm) the application pool
Confirmit.AppInit.SurveyEngine.InitPage	/warmup	This page is called by the AppInit function in IIS for prewarming the Survey Engine
Confirmit.Database.Server.ConfirmitUrl.DataPath	d:\confirmit_sql_data	SQL Data (mdf) drive for the Confirmit ShortURL database

Name	Example Value	Explanation
Confirmit.Database.Server.ConfirmitUrl.LogPath	e:\confirmit_sql_log	SQL Log (ldf) drive for the Confirmit ShortURL database
Confirmit.Database.Server.ConfirmitUrl.ServerName	confirm_system	SQL instance name hosting the Confirmit ShortURL database
Confirmit.Database.Server.DocumentStore.DataPath	d:\confirmit_sql_data	SQL Data (mdf) drive for the Confirmit Documentstore database
Confirmit.Database.Server.DocumentStore.LogPath	e:\confirmit_sql_log	SQL Log (ldf) drive for the Confirmit Documentstore database
Confirmit.Database.Server.DocumentStore.ServerName	confirm_system	SQL instance name hosting the Confirmit Documentstore database
Confirmit.Database.Server.NodeStore.DataPath	d:\confirmit_sql_data	SQL Data (mdf) drive for the Confirmit Nodestore database
Confirmit.Database.Server.NodeStore.LogPath	e:\confirmit_sql_log	SQL Log (ldf) drive for the Confirmit Nodestore database
Confirmit.Database.Server.NodeStore.ServerName	confirm_system	SQL instance name hosting the Confirmit Nodestore database
Confirmit.Database.Server.Reporting.DataPath	d:\confirmit_sql_data	SQL Data (mdf) drive for the Confirmit Hub database
Confirmit.Database.Server.Reporting.LogPath	e:\confirmit_sql_log	SQL Log (ldf) drive for the Confirmit Hub database
Confirmit.Database.Server.Reporting.ServerName	confirm_hub1	SQL instance name hosting the Confirmit Hub database
Confirmit.Database.Server.Search.DataPath	d:\confirmit_sql_data	SQL Data (mdf) drive for the Confirmit Search Service database
Confirmit.Database.Server.Search.LogPath	e:\confirmit_sql_log	SQL Log (ldf) drive for the Confirmit Search Service database
Confirmit.Database.Server.Search.ServerName	confirm_system	SQL instance name hosting the Confirmit Search Service database.
Confirmit.Database.Server.Survey.DataPath	d:\confirmit_sql_data	SQL Data (mdf) drive for the Confirmit Survey databases.
Confirmit.Database.Server.Survey.DataPathN	d:\confirmit_sql_data	SQL Data (mdf) drive for additional Confirmit Survey databases where 'N' is the instance ID (e.g. confirmit_surveys1, confirmit_surveys2 ...)
Confirmit.Database.Server.Survey.LogP	e:\confirmit_sql_log	SQL Log (ldf) drive for the Confirmit Survey

Name	Example Value	Explanation
ath		databases
Confirmit.Database.Server.Survey.LogPathN	e:\confirmit_sql_log	SQL Log (ldf) drive for additional Confirmit Survey databases where 'N' is the instance ID (e.g. confirmit_surveys1, confirmit_surveys2 ...)
Confirmit.Database.Server.Survey.ServerName	confirmit_surveys	SQL instance name hosting the Confirmit Survey databases
Confirmit.Database.Server.Survey.ServerNameN	confirmit_surveys2	SQL instance name hosting additional Confirmit Survey databases where 'N' is the instance ID (e.g. confirmit_surveys1, confirmit_surveys2 ...)
Confirmit.Database.Server.System.DataPath	d:\confirmit_sql_data	SQL Data (mdf) drive for the Confirmit System databases
Confirmit.Database.Server.System.FilestreamPath	e:\confirmit_sql_filestream	Path for storing SQL filestream files on the System instance
Confirmit.Database.Server.System.LogPath	e:\confirmit_sql_log	SQL Log (ldf) drive for the Confirmit System databases
Confirmit.Database.Server.System.ServerName	confirm_system	SQL instance name hosting the Confirmit Search Service database
Confirmit.Database.User.AspState.Name	ConfirmitASPState	SQL User used by the application. Generated automatically during deployment if it does not exist.
Confirmit.Database.User.AspState.Password	*****	Needs to be set as Sensitive Variable
Confirmit.Database.User.AuthorC.Name	ConfirmitAuthor	SQL User used by the application. Generated automatically during deployment if it does not exist.
Confirmit.Database.User.AuthorC.Password	*****	Needs to be set as Sensitive Variable
Confirmit.Database.User.ConfirmitUrl.Name	ConfirmitURL	SQL User used by the application. Generated automatically during deployment if it does not exist.
Confirmit.Database.User.ConfirmitUrl.Password	*****	Needs to be set as Sensitive Variable
Confirmit.Database.User.DeployC.Name	ConfirmitDeployer	SQL User used by the application. Generated automatically during deployment if it does not exist.

Name	Example Value	Explanation
Confirmit.Database.User.DeployC.Password	*****	Needs to be set as Sensitive Variable
Confirmit.Database.User.DocumentStore.Name	ConfirmitDeployer	SQL User used by the application. Generated automatically during deployment if it does not exist.
Confirmit.Database.User.DocumentStore.Password	*****	Needs to be set as Sensitive Variable
Confirmit.Database.User.Helper.Name	ConfirmitHelper	A SQL user used during deployment steps. Generated automatically during deployment if it does not exist.
Confirmit.Database.User.Helper.Password	*****	Needs to be set as Sensitive Variable
Confirmit.Database.User.NodeStore.Name	ConfirmitDeployer	SQL User used by the application. Generated automatically during deployment if it does not exist.
Confirmit.Database.User.NodeStore.Password	*****	Needs to be set as Sensitive Variable
Confirmit.Database.User.ReportC.Name	ConfirmitReport	SQL User used by the application. Generated automatically during deployment if it does not exist.
Confirmit.Database.User.ReportC.Password	*****	Needs to be set as Sensitive Variable
Confirmit.Database.User.Reporting.Name	ConfirmitDeploy	The SQL User used for the Confirmit Hub instance. Generated automatically during deployment if it does not exist. NOTE: These values will be used for all Hub instances unless there is an override. To override, specify variables Confirmit.Database.User.Reporting.Name1 and Confirmit.Database.User.Reporting.Password1 for the first additional instance, then Name2/Password2 for the second additional instance, etc.
Confirmit.Database.User.Reporting.Password	*****	Needs to be set as Sensitive Variable
Confirmit.Database.User.Search.Name	ConfirmitDeploy	SQL User used by the application. Generated automatically during deployment if it does not exist.

Name	Example Value	Explanation
Confirmit.Database.User.Search.Password	*****	Needs to be set as Sensitive Variable
Confirmit.Database.User.SurveyC.Name	ConfirmitSurvey	SQL User used by the application. Generated automatically during deployment if it does not exist.
Confirmit.Database.User.SurveyC.Password	*****	Needs to be set as Sensitive Variable
Confirmit.Database.User.Surveys.Name	SA	The SQL User used for accessing Confirmit Survey databases. NOTE: These values will be used for all Survey instances unless there is an override. To override, specify variables Confirmit.Database.User.Surveys.Name1 and Confirmit.Database.User.Surveys.Password1 for the first additional instance, then Name2/Password2 for the second additional instance, etc.
Confirmit.Database.User.Surveys.Password	*****	Needs to be set as Sensitive Variable
Confirmit.Database.User.SystemAdmin.Name	SA	The SQL User used for administrative tasks on databases. This user needs SA permissions, and needs to exist on the main system instance before installation.
Confirmit.Database.User.SystemAdmin.Password	*****	Needs to be set as Sensitive Variable
Confirmit.Deploy.WindowSize.Authoring	2	The amount of servers to deploy the Authoring role to at the same time. Use with caution and ensure that enough servers are available at all time to avoid downtime during deployment.
Confirmit.Deploy.WindowSize.Reportal	2	The amount of servers to deploy the Reportal role to at the same time. Use with caution and ensure that enough servers are available at all time to avoid downtime during deployment.
Confirmit.Deploy.WindowSize.SurveyEngine	1	The amount of servers to deploy the Deployment (data collection) role to at the same time. Use with caution and ensure that enough servers are available at all time to avoid downtime during deployment.

Name	Example Value	Explanation
Confirmit.IsAlive.Offload.Delay	20	The period in seconds the deployment script will wait for traffic to bleed off from a server before performing disruptive steps. Based on renaming an isalive.htm file hosted in \Inetpub\wwwroot
Confirmit.IsAlive.Preload.Delay	10	The period in seconds the deployment script will wait for the application to start up before putting a server back in the loadbalanced group. Based on renaming an isalive.htm file hosted in \Inetpub\wwwroot
Confirmit.MessageBroker.User.Name	RabbitMessageBroker	The username for the RabbitMQ message broker service
Confirmit.MessageBroker.User.Password	*****	Needs to be set as Sensitive Variable
Confirmit.NuGet.TentacleDownload	FALSE	Determines whether or not the tentacle on the application server downloads the nuget package from the nuget feed (TRUE) or the Octopus server (FALSE)
(If using PowerMTA) Confirmit.PowerMTA.Server.LocalPath.ArchiveLogFolder	P:\LogFiles\pmta\process	Sets the location of the logfiles generated by PowerMTA, which will be picked up by the Confirmit MTA Notification service
(If using PowerMTA) Confirmit.PowerMTA.Server.LocalPath.ErrorFileFolder	P:\LogFiles\pmta\error	Sets the output location of logfiles the Confirmit MTA Notification service is unable to process
(If using PowerMTA) Confirmit.PowerMTA.Server.LocalPath.ProcessedFileFolder	P:\LogFiles\pmta\archive	Sets the output location of logfiles the Confirmit MTA Notification service has successfully processed. These files can be stored elsewhere for record keeping
(If using PowerMTA) Confirmit.PowerMTA.Site.Setting.DeleteFilesOnProcessing	FALSE	Determines whether or not the Confirmit MTA Notification service should delete logfiles after processing them
(If using PowerMTA) Confirmit.PowerMTA.Site.Setting.ProcessedFilesLiveLength	60	Sets the time in days logfiles will be stored in the Confirmit.PowerMTA.Server.LocalPath.ProcessedFileFolder before being cleaned up
Confirmit.Reportal.Pools	0-3,4-7,8-b,c-f	Splits up Reports into multiple application pools, and creates IISRewrite rules. Reportal GUID's are based on hex, so this example will create 4 application pools. Single reports can also be split out by having the first part of the Reportal GUID as the first part of the configuration value

Name	Example Value	Explanation
		(<reportGUID>,0-3,4-7,...)
Confirmit.Server.Certificate.SSL		
Confirmit.Server.Certificate.SSL.Password	*****	Needs to be set as Sensitive Variable
Confirmit.Server.IIS.WebSite	Default Web Site	Determines the website to use
Confirmit.Server.LocalPath.ConfirmitData	e:\confirmit	Local path on the application servers for hosting data files. This folder is recommended to back up.
Confirmit.Server.LocalPath.ConfirmitProgram	d:\confirmit	Local path on the application servers for program files.
Confirmit.Server.NodeExtraCACertsFilePath	c:\\confirmit_prog\\extra_ca_certs.pem *Note! The path must have double backslashes.*	This variable can be used if you use privately signed/self signed certificates in your environment and need to add them to Node.js' list of approved certificate vendors When set, the well known "root" CAs (like VeriSign) will be extended with the extra certificates in the .pem file listed here. The file should consist of one or more trusted certificates in PEM format. The .pem file can be placed in this exact path locally on all servers, or a network share (as long as the app pool identity has access there). The Confirmit.Server.Bootstrapper package, executed on all servers, will update the web.config at the web site root level, for example c:\inetpub\wwwroot\web.config, and add a “NODE_EXTRA_CA_CERTS” value in the AppSetting section pointing to the specified .pem file. This web.config will be used by all services and apps on the server - both existing and future.
Confirmit.Server.User.ConfirmitDeployer.Name	domain\confirmitdeploy	Windows user used to deploy files across multiple servers in a Confirmit site. Can be a local user or a domain user. Must exist prior to deployment and have a profile on all machines in the site. Must have the permissions Logon as a service and Logon as a batch job on all machines.
Confirmit.Server.User.ConfirmitDeployer.Password		Needs to be set as Sensitive Variable
(BETA) Confirmit.SessionState.Redis.Enabled	FALSE	Determines whether or not to use Redis as cache for session state for Reportal and

Name	Example Value	Explanation
		Authoring
(BETA) Confirmit.SessionState.Redis.Host	Server01:6379,Server02:6379, Server03:6379	List of hostname:port of servers hosting the Redis cluster
Confirmit.Site.Application.Default.SQLServerDataPath	d:\confirmit_sql_data	SQL Data (mdf) drive for the default Confirmit Microservice databases.
Confirmit.Site.Application.Default.SQLServerLogPath	e:\confirmit_sql_log	SQL Log (ldf) drive for the default Confirmit Microservice databases.
Confirmit.Site.Application.Default.SQLServerName	confirm_system	Default SQL server instance name for databases from microservices deploying their own database. Can be overridden per microservice by specifying variables Confirmit.Site.Application.<name of microservice>.SQLServerName,Confirmit.Site.Application.<name of microservice>.SQLServerDataPath and Confirmit.Site.Application.<name of microservice>.SQLServerLogPath
Confirmit.Site.HostName.Authoring	author.site.com	Base URL for the Confirmit Authoring application
Confirmit.Site.HostName.Reportal	reportal.site.com	Base URL for the Confirmit Reportal application
Confirmit.Site.Id	CustomerName_Site	Site name displayed in the Confirmit application
Confirmit.Site.Identity.ClientKeyGeneratorSecret	Automatically generated	Secret used for generation of trusted subsystem client keys. Default value is a new Guid unique to this environment. This value should not be changed after initial installation
Confirmit.Site.IdentityDomain	idp.site.com	Base URL for the Confirmit Identity provider application
(BETA) Confirmit.Site.LogDestination.Address	logstash.site.com	The domain name of the logstash server for log collection
(BETA) Confirmit.Site.LogDestination.Port	9998	The port of the logstash server for log collection
Confirmit.Site.LogPath	D:\Logfiles	The location of the Confirmit Application log files. Please refer to the chapter on logging for more details. This folder will host logfiles which will need to be handled in accordance with your log file policies
Confirmit.Site.SSLEnabled	TRUE	Determines whether or not to create SSL

Name	Example Value	Explanation
		endpoints
Confirmit.Site.SSLAccelerator.Mode	FALSE	For defining if SSL offloading is done by a loadbalancer on the site.
Confirmit.Site.SSLAccelerator.Port	81	If SSL Accelerator Mode is TRUE, this defines the port that SSL traffic will flow from the loadbalancer to servers.
Confirmit.Site.UncPath.FtpRootPath	\\filecluster\FTPRoot	UNC path to the FTP server share
Confirmit.Site.UncPath.ImageUploadRootFolder	\\filecluster\ImageUpload	UNC path to the Image Upload folder
Confirmit.Site.UncPath.SharedData	\\filecluster\ConfirmitShared	UNC path to the common Shared Data folder
Confirmit.Site.Url.DictionaryRestApi	http://rest.site.com/api	URL to the Dictionary service, needs to point to servers with the MetadataAPI role
Confirmit.Site.Url.DownloadBaseUrl	http://author.site.com	Base URL for hosting console application files like CAPI
Confirmit.Site.Url.MetaDataRestApi	http://rest.site.com/confirmit/metadata	URL to the Metadata/ REST API service, needs to point to servers with the MetadataAPI role
Confirmit.Site.Url.ShortUrlApi	http://ws.site.com/confirmit/ConfirmitUrl/api/urls	URL to the ShortURL service, needs to point to servers with the Deployment and Webservice role
Confirmit.Site.Url.ShortUrlRedirect	http://short.url/r/api/urls	URL to the ShortURL service, needs to point to servers with the Deployment and Webservice role
(Add-on) Confirmit.Site.Url.TextAnalyticsWorkflowRestApi	http://ws.site.com/textanalytics/api	Only in use if using the TextAnalytics service.
Confirmit.Site.Url.WebServiceBaseUrl	http://ws.site.com	Base URL for Confirmit external API/ Webservices, needs to point to servers with the Webservice role
(Optional) Confirmit.SurveyEngine.WixPools	0,1,2,3,4,5,6,7,8,9	Leave empty if not in use. This setting is to reduce memory strain on individual application pools. This setting splits Surveys into multiple application pools, and creates corresponding IISRewrite rules. The redirects will be to /wix/0, /wix/1/ ... This example will create 10 application pools.
Octopus.Acquire.MaxParallelism	20	This setting determines how many packages can be downloaded from the Octopus server simultaneously.

Index

D

E

F

G

H

I

J

K

L

M

N

O

P

R

S

T

U

V

W