



CONFIRMIT SERVER MANUAL

Copyright © 2010 by Confirmit. All Rights Reserved.

This document is intended only for registered Confirmit clients. No part of the contents of this document may be reproduced or transmitted in any form or by any means without the written permission of Confirmit.

Confirmit makes no representations or warranties regarding the contents of this manual, and specifically disclaims any implied warranties of merchantability or fitness for any particular purpose. The information in this manual is subject to change without notice.

The companies, names and data used or described in the examples herein are fictitious.

The information herein describes the Confirmit Server setup and its features as of November 2010. New features may be introduced after this date. Go to www.confirmit.com or check “News” on the Customer Extranet for the latest updates.

Table of Contents

Table of Contents	3
What's New in this Issue?	8
1. About This Manual	1
2. Confirmit Architecture and Features	2
2.1. Basics	2
2.2. Features and Modules	2
2.2.1. Confirmit Authoring	2
2.2.2. Confirmit Deployment	3
2.2.3. Publisher.....	3
2.2.4. Confirmit Reportal.....	3
2.2.5. Confirmit Task System.....	3
2.2.6. Response Data Storage	4
2.2.7. Application Data Storage	4
2.2.8. Subsystems and Optional Components.....	4
2.2.8.1. Confirmit Web Services	4
2.2.8.2. Confirmit Translator	4
2.2.8.3. Confirmit BitStream Service.....	4
2.2.8.4. Confirmit Multimode.....	4
2.2.8.5. Confirmit Flex Extension	4
2.2.8.6. Add-ons	4
2.3. Application Dataflow	5
2.3.1. Confirmit Authoring Dataflow	5
2.3.2. Confirmit Survey Dataflow	6
2.3.3. Confirmit Reportal Dataflow.....	7
2.3.4. Confirmit Data Export Dataflow.....	7
2.3.5. Confirmit BitStream Dataflow.....	8
3. Building a Confirmit Site	9
3.1. Before You Start	9
3.2. Network and Hardware Configuration.....	9
3.2.1. Server Hardware.....	9
3.2.1.1. Web Servers (Frontend)	9
3.2.1.2. Task System Servers (Backend)	10
3.2.1.3. Database Servers (Backend).....	10
3.2.1.4. FastObjects Database Server.....	11
3.2.1.5. Multimode Server.....	11
3.2.2. Network Equipment	11
3.2.2.1. Switches	11
3.2.2.2. Load Balancers.....	11
3.2.2.3. Firewalls/IDS.....	11
3.3. Site Layout.....	12
3.3.1. Basis for Setting Up the Recommendations	12
3.3.2. Example Scenario.....	13
3.3.3. Entry-Level Multi-Server Setup.....	13
3.3.4. Mid-Level Multi-Server Setup.....	14
3.3.5. High-End Multi-Server Setup	15
3.4. Virtual Infrastructure	17
4. Preparing an Installation	18
4.1. Operating System.....	18

4.1.1. Supported Operating Systems.....	18
4.2. Windows Server Editions.....	18
4.2.1. Key Features	18
4.3. Operating System Additions and Patches	19
4.3.1. Service Packs and Updates.....	19
4.3.2. Additional Components.....	19
4.3.2.1. Microsoft .NET Framework 1.1	19
4.3.2.2. Web Services Enhancements 2.0 SP3 for .NET Framework 1.1	19
4.3.2.3. Enterprise Instrumentation Framework.....	19
4.3.2.4. Microsoft .NET Framework 2.0 SP1	19
4.3.2.5. Microsoft .NET framework 3.5 SP1.....	19
4.3.2.6. Web Services Enhancements 3.0 for .NET Framework 2.0.....	19
4.3.2.7. Microsoft Visual J# 2.0 for .NET Framework 2.0 Second Edition.....	20
4.3.2.8. Microsoft Report Viewer Distributable 2008 SP1	20
4.3.2.9. Microsoft SQL Server 2008 Native client	20
4.3.2.10. Microsoft SQL Server 2008 Shared Management Objects	20
4.3.2.11. Microsoft SQL Server 2008 CLR Types.....	20
4.3.2.12. Open XML SDK 2.0	20
4.3.2.13. Microsoft Visual C++ 2008 Redistributable Package	20
4.4. Additional Components per Server Role.....	20
4.4.1. General / Required for All Features (Except Dedicated Database Servers)	22
4.4.2. Authoring and Express Features	23
4.4.2.1. Dundas Web Chart/Gauge Components (Optional).....	23
4.4.2.2. Wintertree Software Spelling Server DE/XE (Optional)	23
4.4.3. Task System Features (Batch Server).....	24
4.4.3.1. SQL Server Client Tools and SQL Server 2008 Native Client	24
4.4.3.2. SQL Server Integration Services (SSIS).....	24
4.4.3.3. SQL Server Reporting Services (SSRS).....	24
4.4.3.4. Microsoft SQL Server 2008 Analysis Services 10.0 OLE DB Provider	24
4.4.3.5. Microsoft SQL Server 2008 Datamining Viewer Controls	24
4.4.3.6. SQL Server Components Version Support	24
4.4.3.7. Dundas Web Chart/Gauge Components	25
4.4.3.8. Microsoft Office 2007 (English Version).....	25
4.4.4. Dundas Web Chart/Gauge Components in Reportal.....	26
4.4.5. Community Panels - Panelist Reporting (OLAP)	26
4.4.6. Confirmit XML Web Services Development.....	26
4.4.7. FastObjects Server.....	26
4.4.8. Database Server - Version Support.....	26
4.4.8.1. Microsoft SQL Server 2008 Editions.....	27
4.4.8.2. Configuring the Model Database	27
4.4.8.3. Enable FILESTREAM on SQL	28
5. Installing Confirmit.....	29
5.1. Upgrading from an Existing Version	29
5.1.1. Confirmit Licensing	29
5.1.2. Changing the Location of PGP Folders	29
5.1.3. Keeping Existing Settings	30
5.1.4. Stop the Confirmit Task System Service	30
5.1.5. Uninstall Confirmit SDK	30
5.1.6. Back up the FastObjects Database	30
5.1.7. Disable FastObjects Incremental Backup	30
5.1.8. Back-up Confirmit System Databases	30
5.2. Pre-Installation.....	30

- 5.2.1. Creating Data Folders for SQL Databases 30
- 5.2.2. Setting up User Accounts for Confirmit System Services 31
 - 5.2.2.1. Account for the Confirmit Task System and Confirmit BitStream Services 31
 - 5.2.2.2. Account for the Confirmit Task System Helper Service 31
 - 5.2.2.3. Account for Reading File Library Files from External Location (Optional)..... 31
- 5.3. Installation 31
 - 5.3.1. Install / Upgrade Sequence 32
 - 5.3.2. Confirmit Installation 32
 - 5.3.3. The Confirmit Configuration Tool 35
- 5.4. Post-Installation Server Configuration 41
 - 5.4.1. Replacing browscap.ini 42
 - 5.4.2. Obtaining and Installing a License File 42
 - 5.4.2.1. Licensing Concepts 42
 - 5.4.2.2. Add-ons 42
 - 5.4.2.3. Finding the Site ID 43
 - 5.4.2.4. Installing the License File..... 44
 - 5.4.2.5. Confirmit SDK License..... 45
 - 5.4.3. File System Permissions - Confirmit Data Share 45
 - 5.4.4. Assigning Tasks to a Task System Server 45
 - 5.4.4.1. Task Isolation Level 46
 - 5.4.5. Deploying Language and Resource Files 46
 - 5.4.6. BitStream Server Operation Modes 46
 - 5.4.6.1. Single Server, No Sync..... 47
 - 5.4.6.2. Single Server With Sync 47
 - 5.4.6.3. Multi Server With Sync 47
 - 5.4.7. Setting Up the Confirmit File Library 47
- 5.5. Upgrading From a Previous Version..... 48
 - 5.5.1. Assigning New Task Types to Task System Servers..... 48
 - 5.5.2. Deploying Language and Resource Files 48
- 5.6. Confirmit Configuration 48
 - 5.6.1. Configuration System Concepts 48
 - 5.6.1.1. Site-Wide Settings 48
 - 5.6.1.2. Role-Wide Settings 49
 - 5.6.1.3. Server-Wide Settings 49
- 5.7. The Command Line Management Tools..... 49
 - 5.7.1. The ConfirmitAdministrator Tool 49
 - 5.7.2. The aspnet_setreg.exe Tool 49
 - 5.7.3. The ConfirmitAdmin Tool 50
- 5.8. Initiating Special Task Types 50
 - 5.8.1. The Project Info Collector Task..... 50
 - 5.8.2. The Archiving Task 50
 - 5.8.3. The Database Cleanup Task 51
- 6. Confirmit Maintenance 52**
 - 6.1. Backup and Recovery..... 52
 - 6.1.1. System Files / Partitions 52
 - 6.1.2. Data Files Generated by Confirmit..... 52
 - 6.1.3. SQL Server Databases 52
 - 6.1.4. The FastObjects Database 53
 - 6.1.4.1. The ptadmin.exe tool 53
 - 6.1.4.2. Backup..... 53
 - 6.1.4.3. Restore 55
 - 6.1.4.4. Confirmit Upgrades 55

6.2. Monitoring and Systems Maintenance	55
6.2.1. SQL Database Server Maintenance	55
6.2.1.1. Restoring Survey Databases	55
6.2.1.2. Automatic Detachment of Inactive Survey Databases	56
6.2.1.3. Deleting Survey Test Databases	57
6.2.1.4. Clearing Contents of Log Tables	57
6.2.1.5. Defragmenting Indexes.....	57
6.2.1.6. Shrinking Databases.....	57
6.2.2. Anti-Virus Recommendations	57
6.2.3. Time Synchronization	57
6.2.4. Event Logs.....	57
6.2.4.1. Secure Event Log Viewing.....	58
6.2.5. IIS Configuration	58
6.2.5.1. IIS Hardening.....	58
6.2.5.2. IIS Log Files.....	58
6.2.6. IIS Monitoring.....	58
6.2.6.1. Using a Basic Page to Monitor Web Server Availability	58
6.2.6.2. Using a Survey to Monitor Survey Engine Availability	59
7. Advanced System Configuration	60
7.1. IIS Advanced Configuration	60
7.1.1. Configuring IIS to Enforce Compatibility Mode for IE8 Browsers	60
7.1.2. Configuring the ASP Response Size Limit in IIS7.....	60
7.1.3. IIS Application Pool Performance Tuning	60
7.1.3.1. Configuration	60
Other Application Pool Recommendations	63
7.1.4. IIS Compression Tuning	64
7.1.5. Preventing Search Engine Indexing of Surveys.....	66
7.1.6. Using Performance Counters to Troubleshoot Survey Problems	66
7.1.7. Using Confirmit BitStream Performance Counters.....	67
7.2. Using Confirmit with a Load Balancer with SSL Acceleration	68
7.3. Database System Redundancy	69
7.3.1. FastObjects in Windows Cluster	69
7.3.1.1. First Cluster Server.....	69
7.3.1.2. Second Cluster Server.....	69
7.3.2. SQL Server in Windows Cluster	72
7.3.3. Multiple SQL Server Instances	72
7.3.3.1. SQL Server Collation	73
7.3.3.2. SQL Server Logins	73
7.3.3.3. Linked Servers.....	73
7.3.3.4. System Databases and Survey Databases.....	73
7.3.3.5. The [confirm_admin] Database.....	73
7.3.3.6. Creating Aliases for the New Database Server.....	74
7.3.3.7. Making the New Database Server Visible in Confirmit.....	74
7.4. Database Encryption	74
7.4.1. Configure Database Encryption.....	74
7.4.1.1. Preparing SQL Server for Database Encryption	74
7.4.1.2. Preparing Confirmit for Database Encryption.....	75
7.5. Confirmit and UAC (User Account Control)	76
7.5.1. When Installing Confirmit.....	76
7.5.1.1. On Windows 2008	76
7.5.1.2. On Windows 2008 R2.....	78
7.5.2. When Using Confirmit.....	80

- 8. Monitoring Confirmit 81**
- 8.1. The Confirmit Monitoring Utility 81
- 8.2. Generic Server Monitoring 81
- 8.3. External Location Monitoring 82
- 8.4. Confirmit-Specific Monitoring and Recommended Sensors 82
 - 8.4.1. Confirmit System Service Monitoring 82
 - 8.4.2. SQL Query for Monitoring Surveys Returning Errors 82
 - 8.4.3. SQL Query for Monitoring Errors from Authoring 82
 - 8.4.4. SQL Query for Monitoring Errors from Reportal 83
 - 8.4.5. SQL Query for Monitoring Errors from Web Services 83
 - 8.4.6. SQL Query for Monitoring Delays in Task Queue Execution 83
 - 8.4.7. SQL Query for Monitoring Failing Launch Survey Tasks 83
 - 8.4.8. Monitoring Memory Usage for the Confirmit Task System Service 83
 - 8.4.9. SQL Query for Monitoring the Number of Attached Survey Databases 84
- 8.5. Monitoring Example 84
- Index 85**

What's New in this Issue?

This is a reformatted edition of the Confirmit Server Manual. The information presented here applies to Confirmit version 15, Service Pack 1. Numerous small changes, and some larger, have been made to the text and illustrations during re-formatting, to tailor the information to the new formats, to correct minor errors and to update the information for the v15 SP release.

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

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.

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 the Microsoft Windows Server platform.

The first tier is based on Microsoft Active Server Pages (ASP) and the Microsoft .NET Framework. Both support 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 ASP and Microsoft .NET Frameworks are natively supported by Microsoft's web platform, Internet Information Services.

The ASP scripts or .NET applications access a middle tier of components that implement business logic and data access.

In the third tier, data storage is provided by two database systems:

- Versant FastObjects .NET, storing application objects such as project definitions, questionnaires, report templates, ACL's, etc.
- Microsoft SQL Server, storing response data and system databases.

Figure 1 illustrates the Confirmit architecture and layout:

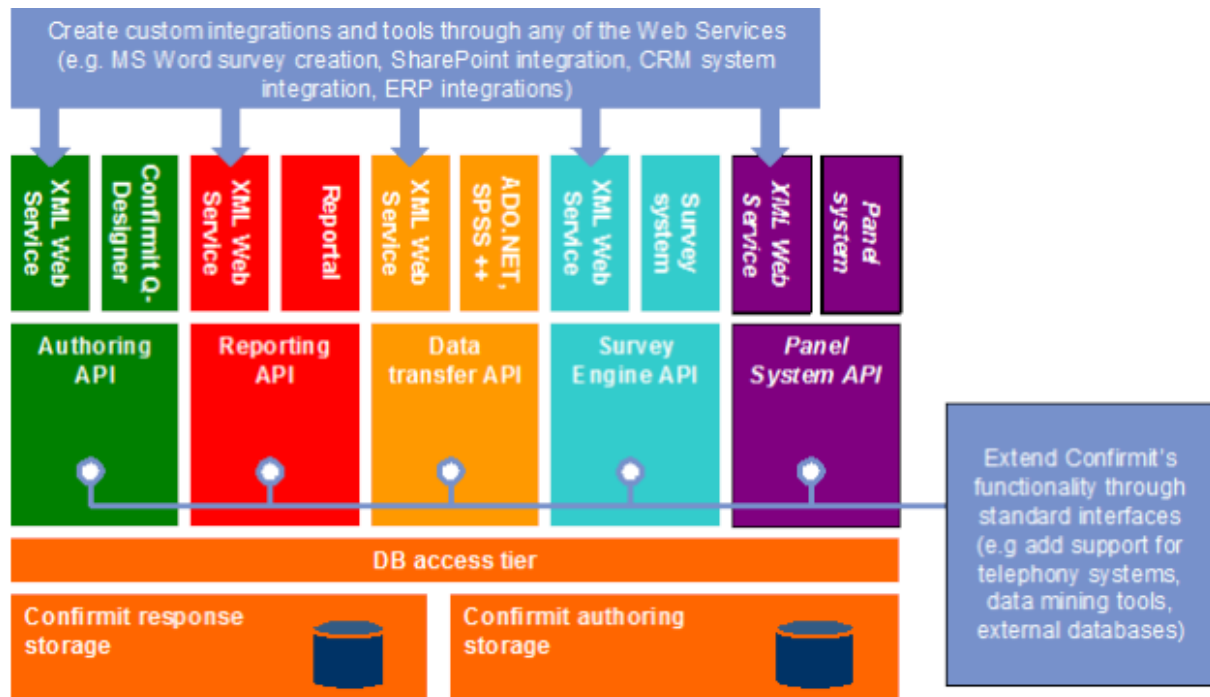


Figure 1 Confirmit architecture

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 Microsoft Internet Explorer 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. For many systems it's common to run the FastObjects database on the server that is also running this module, as the FastObjects server software and database are automatically installed when selecting the Authoring feature during Confirmit setup. It is possible to run a dedicated FastObjects server, or even multiple FastObjects servers configured with database replication or clustering by installing a 'dummy' Authoring server where only the FastObjects database system is used. You can find more information on the FastObjects database system later in this chapter. FTP services are optional for uploading files for importing into the system.

2.2.2. 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.

2.2.3. Publisher

The Publisher module provides static reports and the end user interface, typically for allowing external report viewing by non-Confirmit Authoring users. It is normally run in parallel with surveys on deployment servers, and on Authoring servers for project managers to view their generated reports.

2.2.4. 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. This feature first introduced in Confirmit 8.0 includes features previously unavailable via the now superseded (but still available) "online reporting" service. 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.

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 the Microsoft ASP State Service. The session state control can be configured in Reportal's web.config file.

2.2.5. 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), 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.

2.2.6. 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.

2.2.7. Application Data Storage

The FastObjects Universal Object Server is the central repository for storing application objects such as questionnaire definitions, project ACLs and user, group, role and company objects. This object oriented database system allows for fast indexing and efficient storage of such object types. The FastObjects database supports replication and clustering, although Confirmit does not officially support live fail-over of FastObjects databases. Replication can be used to ensure the database content is cloned onto a passive system which can be activated manually if the primary database server should fail.

Important

The FastObjects database is a critical component 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.

2.2.8. Subsystems and Optional Components

2.2.8.1. 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.

2.2.8.2. Confirmit Translator

The Confirmit Translator module can be acquired as an add-on to Confirmit. When enabled, it allows for outsourcing questionnaire translation 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.

2.2.8.3. 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.

2.2.8.4. 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 may require a separate installation of the Confirmit Multimode module, which is covered in a separate manual.

2.2.8.5. Confirmit Flex Extension

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 Confirmit Flex extension in the user guide, located on the Confirmit Extranet.

2.2.8.6. 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

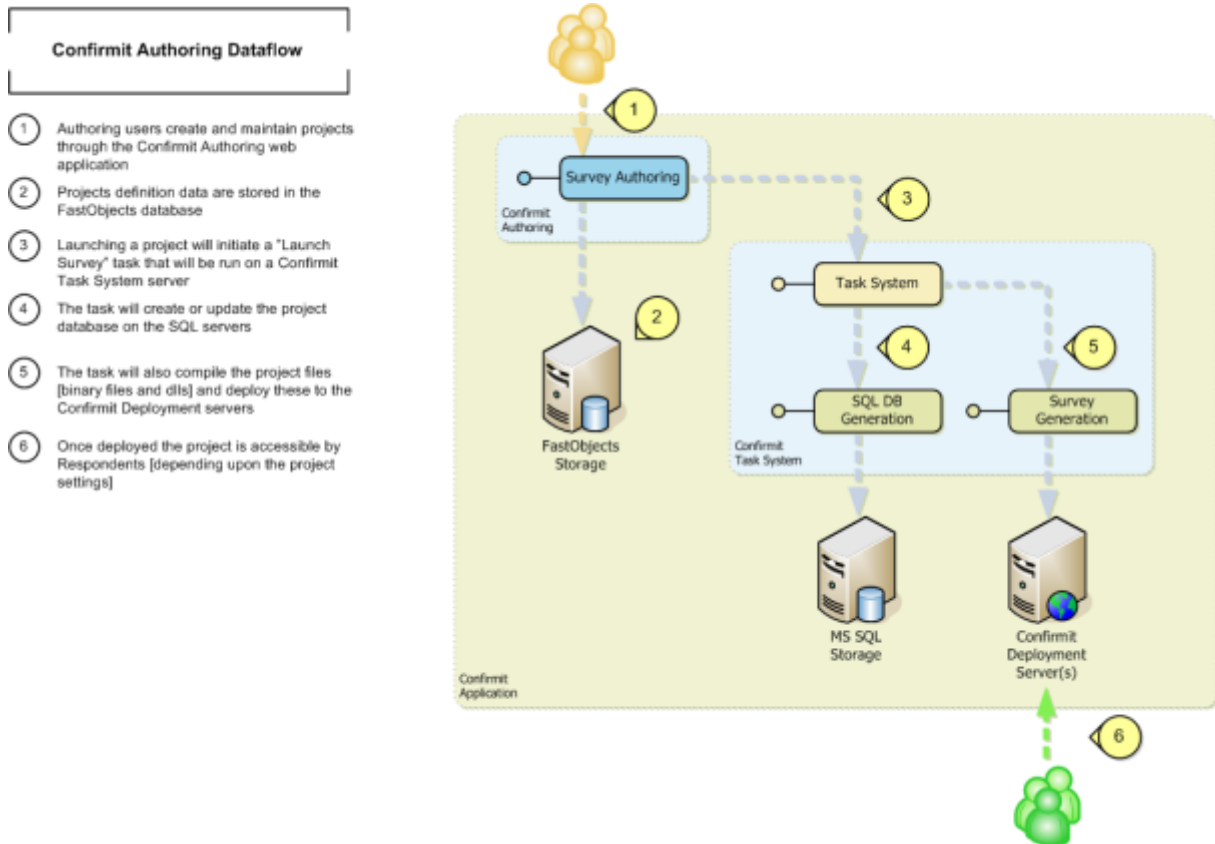


Figure 2 The Confirmit Authoring dataflow

2.3.2. Confirmit Survey Dataflow

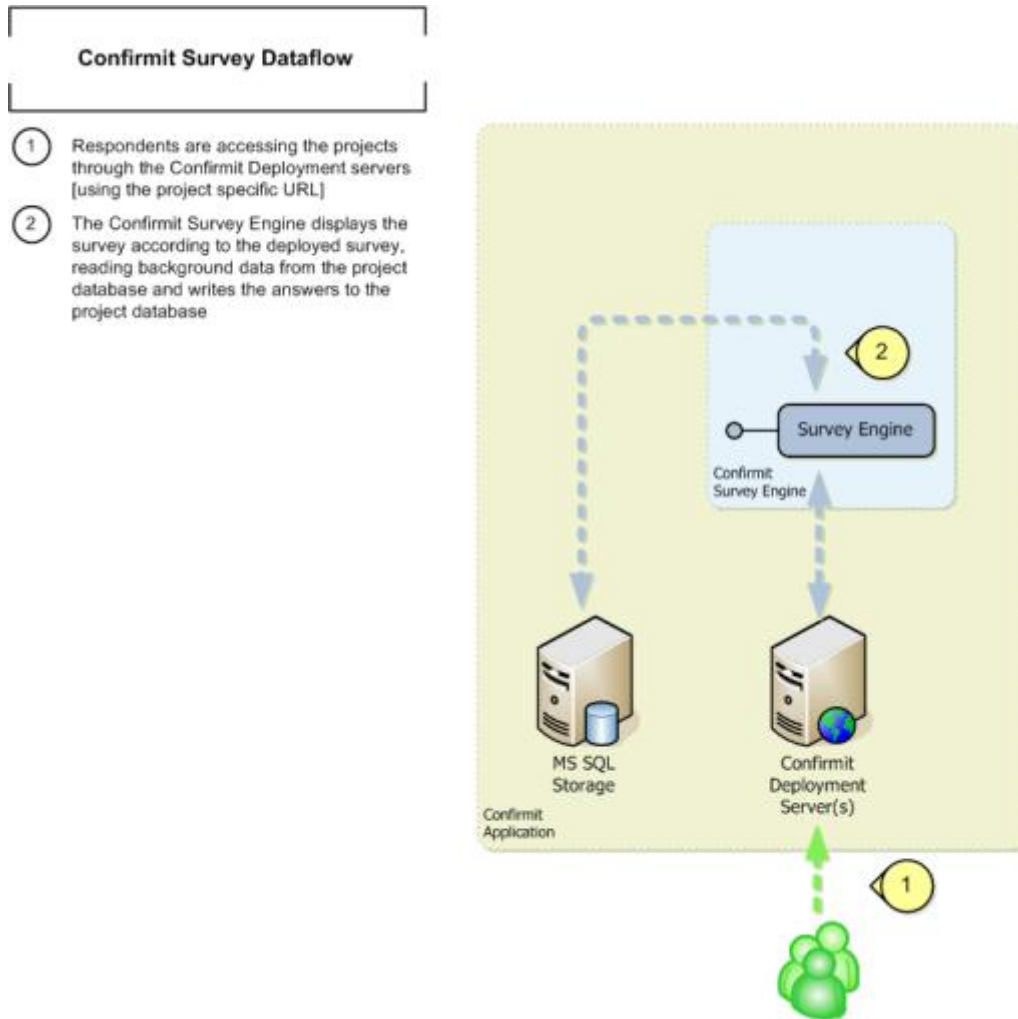


Figure 3 The Confirmit Survey dataflow

2.3.3. Confirmit Reportal Dataflow

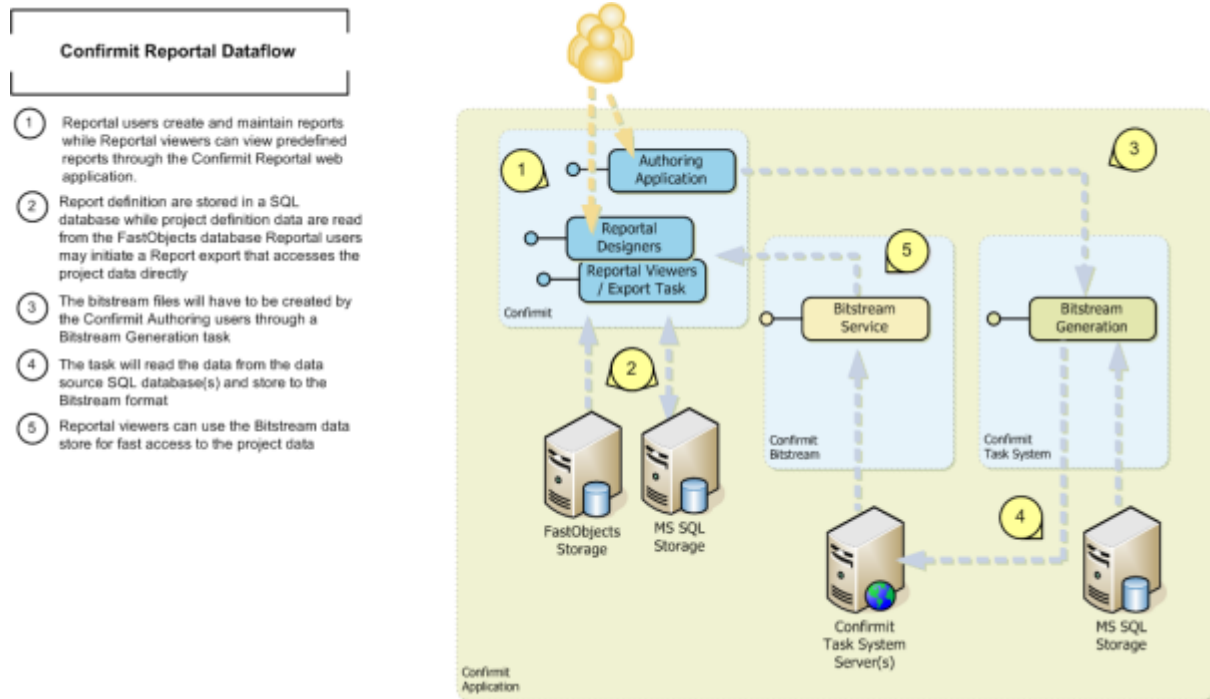


Figure 4 The Confirmit Reportal dataflow

2.3.4. Confirmit Data Export Dataflow

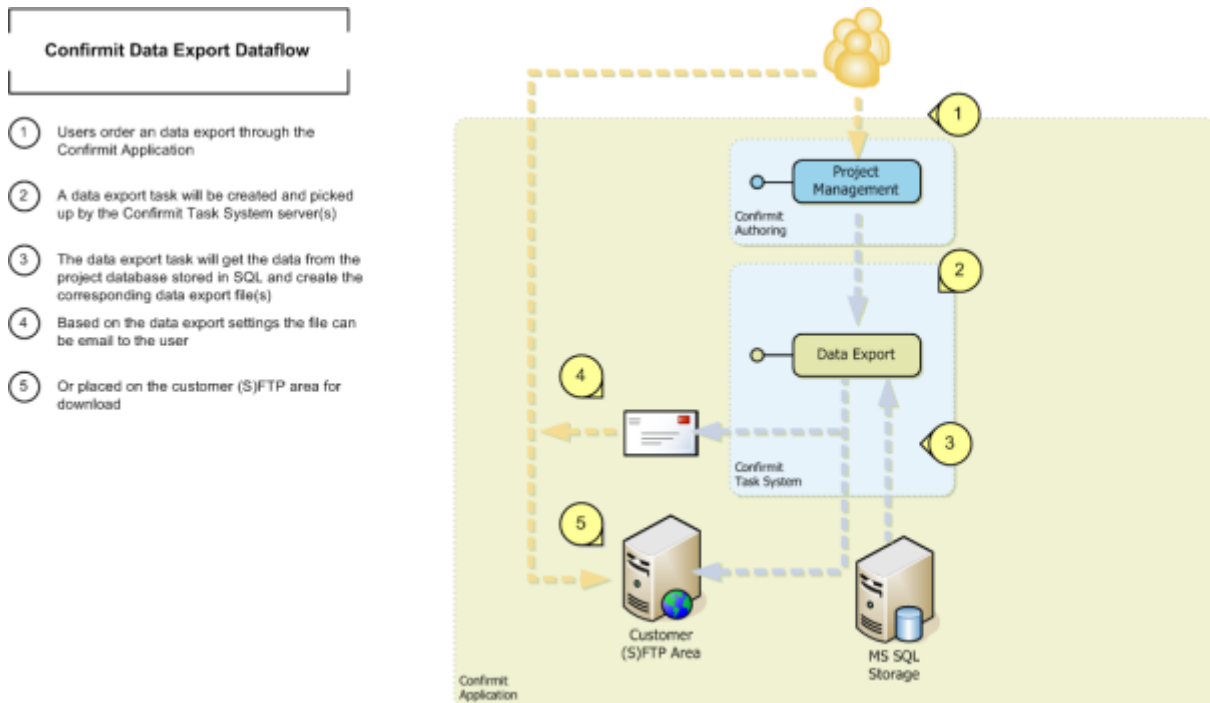


Figure 5 The Confirmit Data Export dataflow

2.3.5. Confirmit BitStream Dataflow

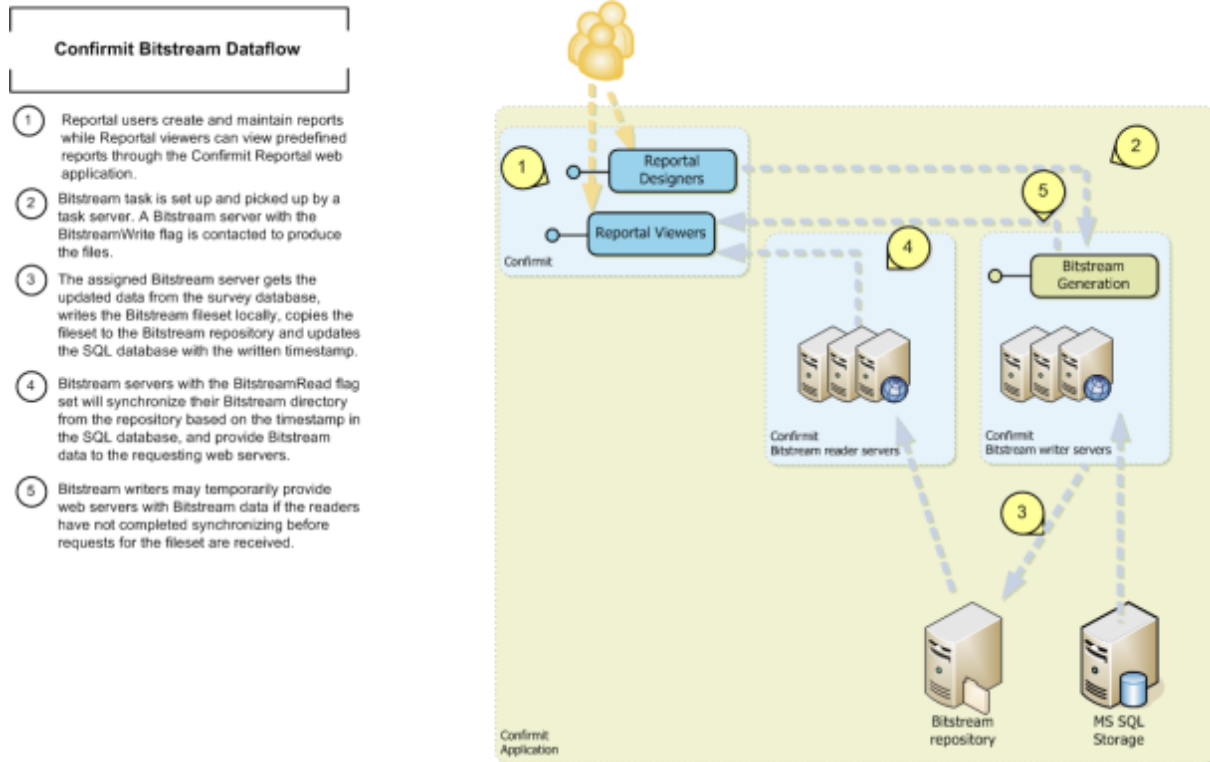


Figure 6 The Confirmit BitStream dataflow

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.

To decide which configuration is more suited for your needs, Confirmit Technical Services / On-Demand Operations team can assist you in 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 ((see the Supported Operating Systems section on page 18 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. Confirmit Technical Services can also assist in setting up a tailored recommendation on request.

3.2.1.1. Web Servers (Frontend)

The typical minimum hardware outfit for a web or combined server (deployment, stand-alone Authoring or Reportal) would be:

- 2Ghz dual-core CPU
- 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 the Load Balancers section on page 11 for more information).

A typical recommended hardware setup:

- Dual quad-core CPUs
- 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

3.2.1.2. Task System Servers (Backend)

Dedicated Task System servers would typically not require as much memory as a web server. 2GB should be sufficient for most cases, 4GB is recommended if the server will also be running SQL Server Integration Services. Other hardware recommendations would follow that of a web server above. There should also be enough disk space on batch servers to store BitStream files unless dedicated BitStream servers are used.

3.2.1.3. 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).
- 2GB RAM (recommended: increase by minimum 2GB 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 2GB per additional SQL Server instance).
- 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-1 or RAID-5 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.

3.2.1.4. FastObjects Database Server

In larger systems it might be preferable to run the FastObjects database on a dedicated server. The recommendations for a server would follow that of a SQL Server database server configured for a single DB instance with the database stored on a fast disk set. If using incremental backups for FastObjects, a separate disk may be used for storing database incremental update files (see the Incremental Backup section on page 54 for more information).

3.2.1.5. Multimode Server

Confirmit Multimode requires (a) separate, dedicated machine(s) for installation of the multimode component. Hardware requirements are the same as for Web Servers (Frontend) (see the Web Servers (Frontend) section on page 9 for more information). The multimode component can be load-balanced if necessary for performance distribution and redundancy.

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 or ASP state 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.

3.2.2.3. Firewalls/IDS

There are no specific requirements in terms of firewall hardware. However it is strongly recommended that firewalls 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	80,443 (customizable)	TCP
Users	Authoring servers	HTTP	80,443 (customizable)	TCP
All Confirmit servers	Remote mail servers	SMTP	25	TCP
<i>Users</i>	<i>Server holding shared data / FTP data</i>	<i>FTP / SFTP / SSH / FRTPS</i>	<i>21 / 22, 990 or custom</i>	<i>TCP</i>

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
All Confirmit servers	SQL Database server(s)	SQL connection	1433 (customizable)	TCP
All Confirmit servers	FastObjects Database server	FastObjects connection	6001, 6002	TCP
Authoring servers, Task System servers	All Confirmit servers (also database servers if Archiving is enabled)	File system access over SMB/CIFS	445	TCP
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

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 $3,000,000 \times (100/20) = 15,000,000$ 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: $112,500 \times (1+25/100) \times (1+30/100) \times (1+30/100) \times (1+30/100)$ 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 FastObjects database since this does not add noticeable strain to the server on which it runs.

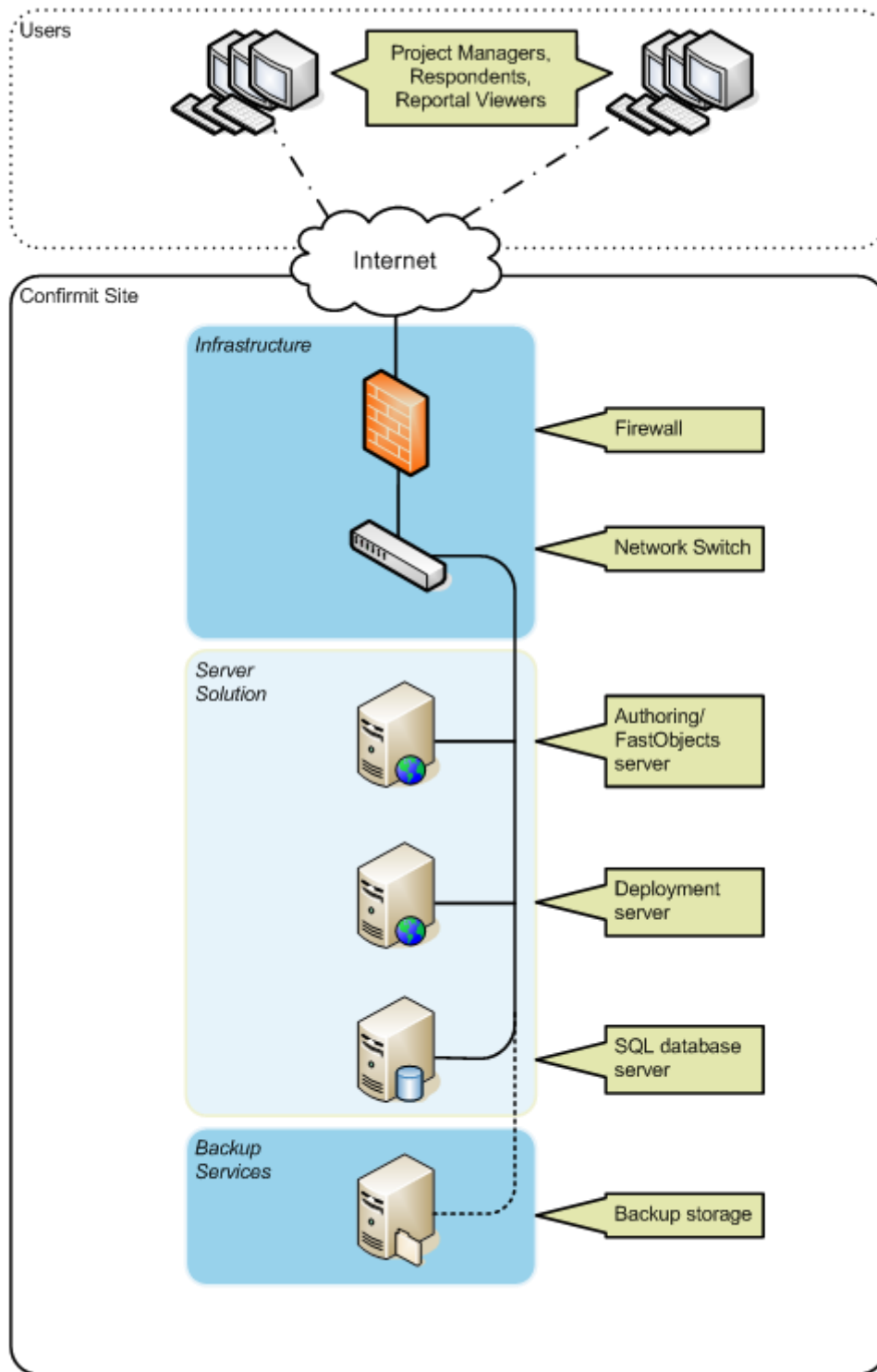


Figure 7 Features divided across three servers

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.

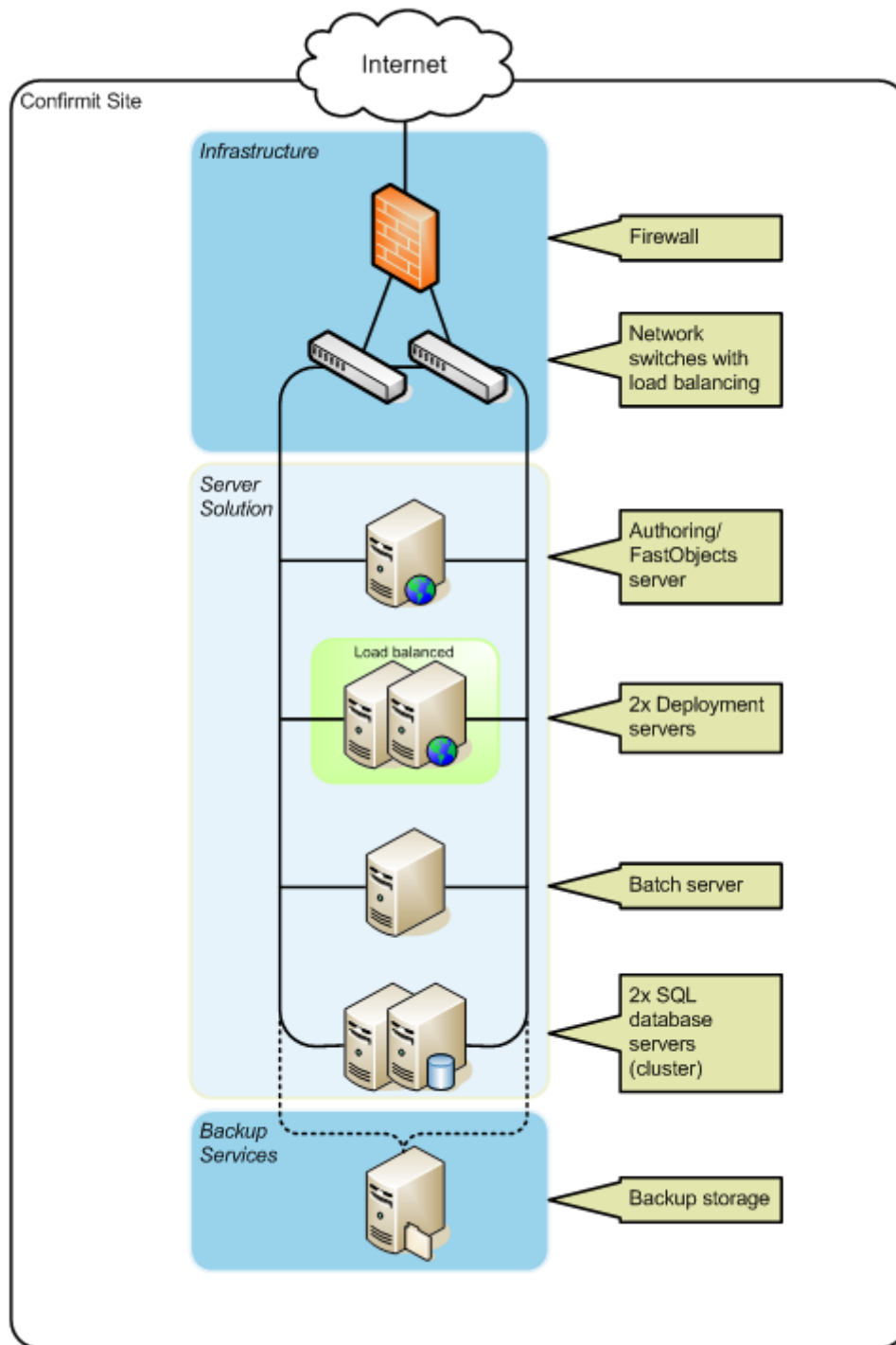


Figure 8 Multi-server setup with redundancy for data collection and response storage

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:

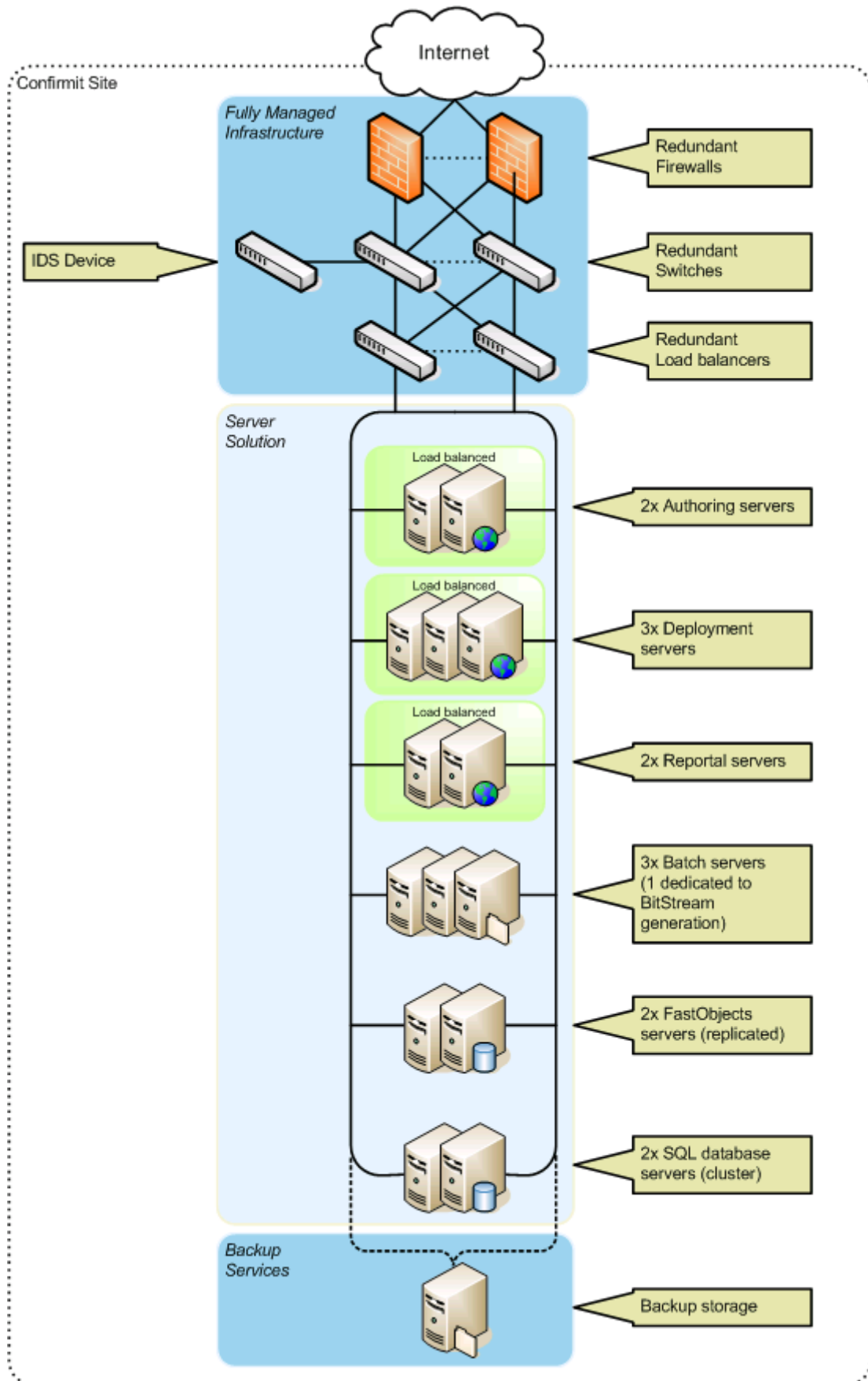


Figure 9 Full redundancy for all features

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

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 On-Demand 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 the following virtualization options:

- VMware Infrastructure 3 (ESX 3.5)
- VMware vSphere 4 (ESX 4.0 and 4.01)
- VMware ESXi (not recommended for production environments)

Disclaimer:

Confirmit does not support running SQL Server or FastObjects on virtual servers.

Disclaimer:

In the event you were to experience issues with the Confirmit Professional software that, to Confirmit's reasonable discretion, appear to be directly 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 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.

4.1. Operating System

Confirmit Horizons v15 Service Pack supports installation on the Windows Server 2008 and Windows Server 2008 R2 operating systems. Only 64-bit editions are supported.

Note: Windows 2003 x86 is supported but for specific features only (see below for further details).

4.1.1. Supported Operating Systems

The following table lists supported operating system versions for running various Confirmit modules in Confirmit Horizons v15 Service Pack:

Server Role	FastO bjects	SQL	Author	Deploy ment	Batch	Reportal	Multimode
Windows Server 2003 32 Bit	No	No	No	No	No *	No	No
Windows Server 2008 32 Bit	No	No	No	No	No	No	No
Windows Server 2003 64 Bit	No	No	No	No	No	No	No
Windows Server 2008 64 Bit	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Windows Server 2008 R2	Yes	Yes	Yes	Yes	Yes	Yes	Yes

* For version 15, batch servers running ActivePDF (pdf exports) may run Windows 2003 x86 in specific cases. Contact support for more information.

4.2. Windows Server Editions

The Windows Server operating systems comes in multiple editions. It's important to choose the right edition for the intended environment as the different versions have fairly different features, and licensing prices vary.

Windows Server 2008 R2 versions can be compared here:

<http://www.microsoft.com/windowsserver2008/en/us/r2-compare-specs.aspx>

4.2.1. Key Features

- **Windows Web Server 2008 / R2** – Supports 4 physical CPUs and 32GB RAM. SQL Server cannot be installed.
- **Windows Server 2008 Standard / R2** – Supports 4 physical CPUs and 32GB RAM, and supports installation of SQL Server 2008 Standard Edition on a stand-alone server.
- **Windows Server 2008 Enterprise / R2** – Supports 8 physical CPUs and up to 2TB RAM, and supports installation of SQL Server 2008 Enterprise Edition in stand-alone or cluster configuration.

4.3. Operating System Additions and Patches

4.3.1. Service Packs and Updates

Keeping servers updated with the latest Critical Updates and Important Updates from Windows update is recommended. Exceptions for any non-compatible updates will be disclosed in this manual or via separate newsletters. Confirmit recommends Windows Server 2008 Service Pack 2 or Windows Server 2008 R2.

4.3.2. Additional Components

The topics linked below describe the additional components that are available.

4.3.2.1. Microsoft .NET Framework 1.1

All new source code is currently being written and compiled for the .NET 3.5 Framework. However, some code still uses NET 1.1, which requires manual installation and an update to SP1 to be fully supported on Windows Server 2008.

Note: Installation of .NET Framework 1.1 gives a message that it is not supported on Windows Server 2008 and above. This can safely be ignored.

4.3.2.2. Web Services Enhancements 2.0 SP3 for .NET Framework 1.1

Web Services Enhancements 2.0 SP3 for .NET 1.1 is a required component. It can be downloaded from Microsoft using this link:

<http://www.microsoft.com/downloads/en/details.aspx?FamilyID=8070e1de-22e1-4c78-ab9f-07a7fc1b6aa>

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, this can be downloaded from:

<http://www.microsoft.com/downloads/details.aspx?displaylang=en&FamilyID=8070e1de-22e1-4c78-ab9f-07a7fc1b6aa>

4.3.2.3. Enterprise Instrumentation Framework

The Enterprise Instrumentation Framework is used for logging from ASP.NET pages to Confirmit monitor tables (Authoring/Express/Community Portal/Reportal/Web Services monitor.) It can be downloaded from:

<http://www.microsoft.com/downloads/details.aspx?FamilyID=80df04bc-267d-4919-8bb4-1f84b7eb1368&DisplayLang=en>

The Enterprise Instrumentation Framework requires the presence of .NET 1.1 before it can be installed.

4.3.2.4. Microsoft .NET Framework 2.0 SP1

Microsoft .NET Framework 2.0 SP1 is a required component. It can be downloaded from:

<http://www.microsoft.com/downloads/en/details.aspx?FamilyID=0856eacb-4362-4b0d-8edd-aab15c5e04f5&displaylang=en>

4.3.2.5. Microsoft .NET framework 3.5 SP1

Microsoft .NET Framework 3.5 SP1 is a required component. It can be enabled as a server feature in the Windows Server 2008 Server Manager.

4.3.2.6. Web Services Enhancements 3.0 for .NET Framework 2.0

Web Services Enhancements 3.0 for .NET 2.0 is a required component. It can be downloaded from Microsoft using this link:

<http://www.microsoft.com/downloads/details.aspx?FamilyID=018a09fd-3a74-43c5-8ec1-8d789091255d&DisplayLang=en>

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, this can be downloaded from:

<http://www.microsoft.com/downloads/details.aspx?FamilyID=9e59c3fb-e7bc-4d91-908e-411a8d329f3d&DisplayLang=en>

4.3.2.7. Microsoft Visual J# 2.0 for .NET Framework 2.0 Second Edition

Visual J# 2.0 SE (Second Edition) is used by Confirmit for communicating with the FastObjects database from the .NET Framework. It can be downloaded from Microsoft:

<http://www.microsoft.com/downloads/details.aspx?FamilyID=42c46554-5313-4348-bf81-9bb133518945&displaylang=en>

4.3.2.8. Microsoft Report Viewer Distributable 2008 SP1

The Microsoft Report Viewer Distributable is used by Community Panels Panelist Reporting feature and is required by the core components of Confirmit Horizons. It can be downloaded from:

<http://www.microsoft.com/downloads/details.aspx?familyid=CC96C246-61E5-4D9E-BB5F-416D75A1B9EF&displaylang=en>

4.3.2.9. Microsoft SQL Server 2008 Native client

The Microsoft SQL 2008 Native client is used by all servers with Confirmit installed, to communicate with the SQL server. It is a part of the SQL server 2008 Feature Pack and can be downloaded from:

<http://www.microsoft.com/downloads/en/details.aspx?FamilyId=C6C3E9EF-BA29-4A43-8D69-A2BED18FE73C&displaylang=en>

4.3.2.10. Microsoft SQL Server 2008 Shared Management Objects

The Shared Management Objects component is used by all servers with Confirmit installed. It is a part of the SQL server 2008 Feature Pack and can be downloaded from:

<http://www.microsoft.com/downloads/en/details.aspx?FamilyId=C6C3E9EF-BA29-4A43-8D69-A2BED18FE73C&displaylang=en>

4.3.2.11. Microsoft SQL Server 2008 CLR Types

The SQL Server CLR Types component is used by all servers with Confirmit installed. It is a part of the SQL server 2008 Feature Pack and can be downloaded from:

<http://www.microsoft.com/downloads/en/details.aspx?FamilyId=C6C3E9EF-BA29-4A43-8D69-A2BED18FE73C&displaylang=en>

4.3.2.12. Open XML SDK 2.0

The Open XML SDK 2.0 Add-on is required on Authoring, Reportal and Batch servers running Office 2007. This component can be downloaded from:

<http://www.microsoft.com/downloads/en/details.aspx?FamilyID=c6e744e5-36e9-45f5-8d8c-331df206e0d0>

Only the smallest file (~3 MB) is needed.

4.3.2.13. Microsoft Visual C++ 2008 Redistributable Package

The Visual C++ Redist package is required on Confirmit servers running the Batch role. It can be installed from here:

<http://www.microsoft.com/downloads/en/details.aspx?familyid=9B2DA534-3E03-4391-8A4D-074B9F2BC1BF&displaylang=en>

4.4. Additional Components per Server Role

Some of the main Confirmit features require third party software. Software and licenses are to be purchased, installed and maintained by the Client, as 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. If a combined server is used, review all features. A compacted list per feature is included below:

Prerequisite	Role	Comment
Microsoft Internet Information Services	All	
Microsoft Simple Mail Transfer Protocol	All	
Microsoft .NET Framework 1.1	All	Installation of .NET Framework 1.1 gives a message that it is unsupported on Windows Server 2008 and above. This can safely be ignored.
Microsoft .NET Framework 2.0 SP1	All	
Microsoft .NET Framework 3.5 SP1	All	Needs to be installed via Microsoft Server Manager
Microsoft J# Runtime 2.0 SE (Second Edition)	All	
WSE2.0 SP3	Author	
WSE3.0	Author	
Browscap.ini	Author	Recommended source: http://browsers.garykeith.com/
Microsoft SQL Server 2008, Integration Services (SSIS)	Batch	
Microsoft SQL Server 2008, Reporting Services (SSRS)	Batch (recommended)	Only required if new OLAP reporting is to be used
Microsoft SQL Server 2008, Analysis Services (SSAS)	Batch (recommended)	Only required if OLAP reporting is to be used
Microsoft SQL Server 2008 Analysis Services 10.0 OLE DB Provider	Batch	Only required if OLAP reporting is to be used
Microsoft SQL Server 2008 Datamining Viewer Controls	Batch	Only required if OLAP reporting is to be used
Microsoft Report Viewer Redistributable 2008 SP1	All	
Microsoft Core XML Services (MSXML) 6.0	All	

Microsoft SQL Server 2008 Native Client	All	Not needed if "Client tools" is installed from SQL setup (recommended). All servers need a way to connect to SQL server using Native Client. Warning: If this is installed, it must be uninstalled before any SQL Server features can be installed on the server (SSIS/SSRS)
Microsoft SQL Server System CLR Types	All	Not needed if "Client tools" is installed from SQL setup. Prerequisite for Microsoft SQL Server 2008 Native Client
Microsoft SQL Server 2008 Shared Management Objects	All	Not needed if "Client tools" is installed from SQL setup. Prerequisite for Microsoft SQL Server 2008 Native Client
Microsoft SQL Server 2008 Management Tools - Complete, x86 version	Batch servers on x64 running SSIS	
Microsoft Visual C++ 2008 Redistributable Package (x86)	Batch	
ActivePDF WebGrabber 2009 *	Batch servers running PDF export	
Microsoft Office 2007 *	Batch server running MS Office Exports or Reportal Report export tasks	
2007 Microsoft Office System Update: Redistributable Primary Interop Assemblies	Batch servers with Office 2007	
Open XML SDK 2.0 for Microsoft Office	Author, Reportal and Batch servers with Office 2007	Updated from v15
Dundas Enterprise *	Author(*), Batch, Reportal. Required on Authoring to display graphs in Rapid Results, but can be omitted to save license costs (no graphs will be displayed).	Add to Global Assembly Cache.
Dundas WebGauge *	Author(*), Batch, Reportal. Required on Authoring to display graphs in Rapid Results, but can be omitted to save license costs (no graphs will be displayed).	Add to Global Assembly Cache.
* Optional		

4.4.1. General / Required for All Features (Except Dedicated Database Servers)

In the Windows Server Manager, ensure the following options are installed and enabled (in addition to any default settings):

- IIS 7 with following role services

Component	Folder
Static Content	Common HTTP Features
Default Document	Common HTTPFeatures
Directory Browsing	Common HTTP Features
HTTP Redirection	Common HTTP Features
ASP.Net	Application Development
ASP	Application Development
ISAPI Extension	Application Development
ISAPI Filters	Application Development
Windows Authentication	Security
Dynamic Content Compression	Performance
IIS Metabase	Management Tools
IIS 6 WMI	Management Tools
IIS Management Scripts and Tools	Management Tools
IIS 6 Scripting Tools	Management Tools

- Additional Windows 2008 Server Features
 - SMTP Service

Note: In order for SMTP logging to work (if this feature is selected), the IIS role feature 'ODBC Logging' must be enabled.

- User Access Control

User Access Control (UAC) must be disabled in order for Confirmit to install properly (see the Confirmit and UAC (User Account Control) section on page 76 for more information).

4.4.2. Authoring and Express Features

4.4.2.1. Dundas Web Chart/Gauge Components (Optional)

Dundas Web Chart Component Enterprise is required on Authoring servers to display charts in Rapid Results (without this component, only tables will be displayed in Rapid Results.) The component should be added to the .NET Global Assembly Cache on the server. The Dundas Web Chart component is a 3rd party add-on that must be purchased separately from Confirmit. To install:

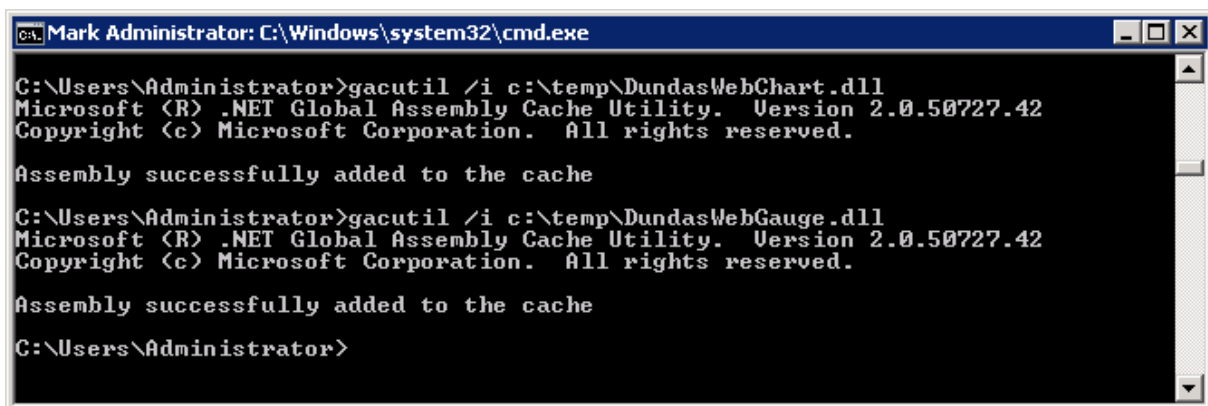


Figure 10 Installing the Dundas component

4.4.2.2. Wintertree Software Spelling Server DE/XE (Optional)

The Wintertree Software Spelling Server can be purchased as an add-on from Confirmit. When installed, it will enable spell checking in the questionnaire designer. Access to the component is added to the Confirmit license file if the add-on has been purchased from Confirmit.

4.4.3. Task System Features (Batch Server)

4.4.3.1. SQL Server Client Tools and SQL Server 2008 Native Client

The files **bcp.exe** and **bcp.rll** from SQL Server Client tools are required in the <confirmprog>\bin folder on batch servers for running sampling and sample upload tasks. The versions of these files on batch servers can be copied from the database server. In order to run the bcp tool on the task system server, the SQL Server 2008 Native Client must be installed. The SQL Server 2008 Native Client is also required for SSIS/SSRS installation on batch servers.

4.4.3.2. SQL Server Integration Services (SSIS)

SQL Server Integration Services 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 on a x64 batch server you will also need Microsoft SQL Server 2008 Management Tools - Complete, x86 version.

4.4.3.3. SQL Server Reporting Services (SSRS)

SQL Server Reporting Service is required only if the Community Panel Panelist Reporting is used (OLAP.). Installation instructions follow in separate documentation.

4.4.3.4. Microsoft SQL Server 2008 Analysis Services 10.0 OLE DB Provider

This requirement only applies if the Community Panel Panelist Reporting feature is being used. The SSAS OLE DB provider is downloadable from:

<http://www.microsoft.com/downloads/details.aspx?FamilyID=c6c3e9ef-ba29-4a43-8d69-a2bed18fe73c&displaylang=en>

4.4.3.5. Microsoft SQL Server 2008 Datamining Viewer Controls

This requirement only applies if the Community Panel Panelist Reporting feature is being used. The datamining viewer controls are downloadable from:

<http://www.microsoft.com/downloads/details.aspx?FamilyID=c6c3e9ef-ba29-4a43-8d69-a2bed18fe73c&displaylang=en>

4.4.3.6. SQL Server Components Version Support

The following SQL Server components are supported in Confirmit Horizons v15:

Component	x86 Supported?	x64 Supported?
SQL Server 2005 SSIS	No	No
SQL Server 2005 SSAS	No	No
SQL Server 2005 SSRS	No	No
SQL Server 2008 SSIS	No	Yes
SQL Server 2008 SSAS	No	Yes
SQL Server 2008 SSRS	No	Yes

4.4.3.7. Dundas Web Chart/Gauge Components

If the Reportal add-on is used, Dundas Web Chart/Gauge Components are required on batch servers to produce exports from Reportal. The component should be added to the Global Assembly Cache on the server. The Dundas components are distributed by Confirmit as part of the Reportal activation package and are licensed on a per machine basis. To install:

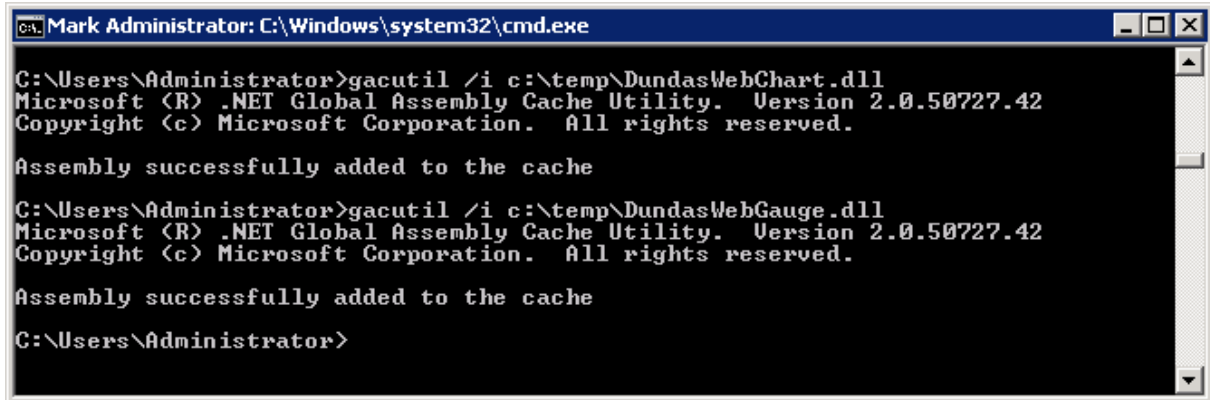


Figure 11 Installing the Dundas components

4.4.3.8. Microsoft Office 2007 (English Version)

Microsoft Office 2007 is required on batch servers for producing exports in Microsoft file formats such as .xls, .ppt and .doc. The minimum required Office components are:

- Excel
 - o .NET Programmability Support
 - o Add-ins
 - o Internet Assistant VBA
- PowerPoint
 - o .NET Programmability Support
- Word
 - o .NET Programmability Support
- Office Shared Features
 - o Converters and Filters
 - o Graphics Filters
 - o Digital Certificate for VBA projects
 - o Fonts
 - o International Support
 - o Microsoft Office Themes
 - o Proofing Tools
 - o English Proofing Tools
 - o Visual Basic for Applications
- Office Tools
 - o Microsoft Graph
 - o Windows SharePoint Services Support

4.4.4. Dundas Web Chart/Gauge Components in Reportal

The Dundas Web Chart Component Enterprise Edition is required on Reportal servers to produce graphs and gauge components in reports. The component should be installed in the Global Assembly Cache on the Reportal server. This component is distributed by Confirmit as part of the Reportal activation package and is licensed on a per machine basis. To install:

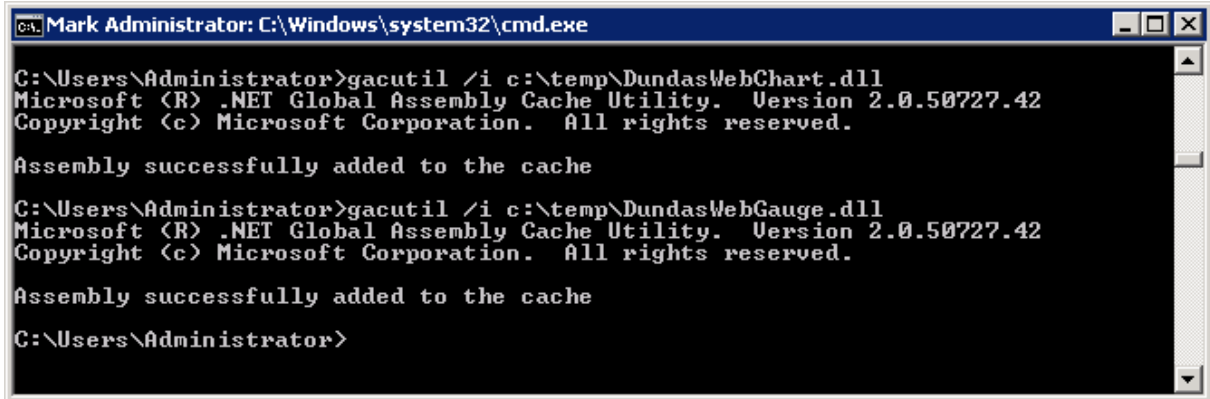


Figure 12 Installing the Dundas component

Alternatively, open **C:\Windows\Assembly** in Windows Explorer and simply drag-and-drop the files into this folder.

4.4.5. Community Panels - Panelist Reporting (OLAP)

In order to use the new panelist reporting features in community panels, SQL Server Analysis Services (SSAS) and SQL Server Reporting Services (SSRS) must be installed. We recommend installing SSAS on a batch server, or on the main database server. Note that it can be installed separately, though this may incur additional SQL Server licensing. SSRS can be installed alongside SSIS on batch servers without additional license costs.

4.4.6. Confirmit XML Web Services Development

To connect to Confirmit Web Services, SOAP (Simple Object Access Protocol) is required. There are several SOAP toolkits on the market; the one you require will depend on which environment you are developing in. If you are developing in a Windows-based environment, the Microsoft SOAP toolkit version 2.0, or the .NET Framework v1.1 can be used.

4.4.7. FastObjects Server

No additional features other than those required for the Confirmit setup are required.

4.4.8. Database Server - Version Support

Microsoft SQL Server version support:

SQL Server Version	Supported?
SQL Server 2000	No
SQL Server 2005 x86	No
SQL Server 2005 x64	No
SQL Server 2008 x86	No

SQL Server 2008 x64	Yes (Minimum SP1)
SQL Server 2008 x64 R2	No

In multi-database server environments, all instances/servers must run the same version of Microsoft SQL Server.

4.4.8.1. Microsoft SQL Server 2008 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 2008 Standard Edition will be sufficient for running Confirmit in a production environment. SQL Server 2008 Standard Edition also supports installation in a fail-over cluster, albeit only with two servers in the cluster. For enterprise installations, SQL Server 2008 Enterprise Edition can be considered for high availability features such as 16-node cluster support, 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 2008 is right for your environment, the following link might be of help:

<http://www.microsoft.com/sqlserver/2008/en/us/editions-compare.aspx>.

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 a SQL login with 'sysadm' permissions during installation but the account is not needed after the initial installation is complete. Since Confirmit v15 does not use any login account with sysadm permissions, a login may have to be temporarily enabled for performing future upgrades.

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.

4.4.8.1.1. Required Update

SQL Server 2008 Service Pack 1 is required for installing Confirmit Horizons v15 Service Pack 1.

4.4.8.1.2. 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.

4.4.8.2. 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 10% of the database size. 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 should 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.

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.

4.4.8.3. Enable FILESTREAM on SQL

FILESTREAM is required on the main (first) SQL Server instance for installing Confirmit Horizons v15. This is 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:

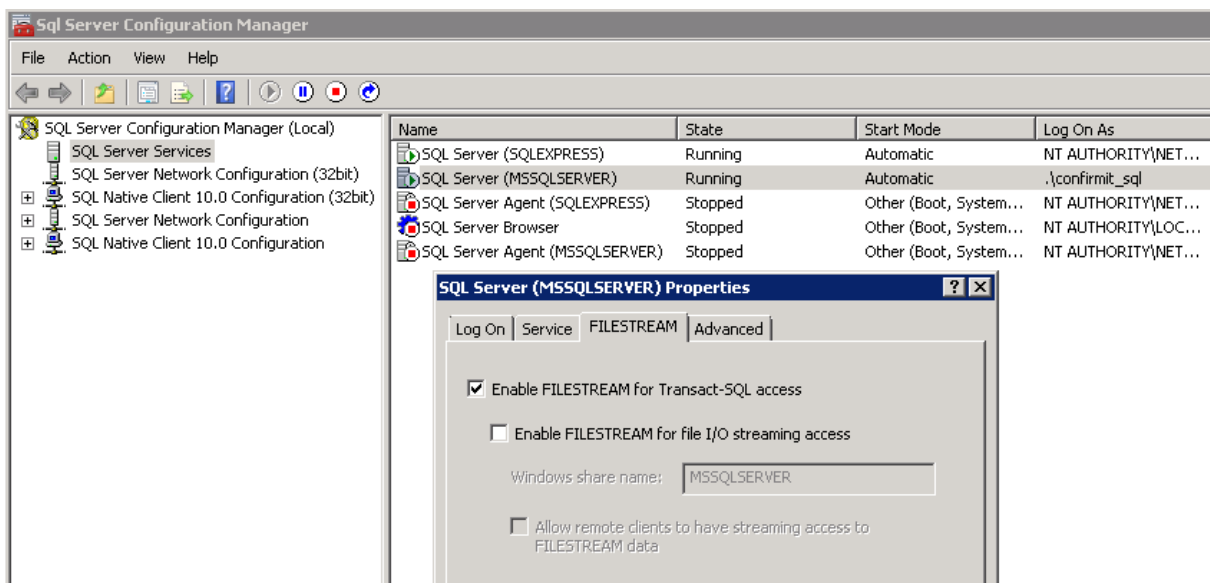


Figure 13 Enabling FILESTREAM

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

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

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

5. Installing Confirmit

After the servers have been set up, all connectivity has been tested and Confirmit system requirements have been installed, the next step is to install Confirmit itself. This version of this manual provides instructions for installing **Confirmit Horizons v15 Service Pack**.

5.1. Upgrading from an Existing Version

For Confirmit v15 Service Pack we have created a whole new setup program. If you are performing an upgrade from a previous version, you may notice that a lot of windows still are being opened during the setup process; this is due to the old setup uninstalling the old version before installing the new 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.

Important

After you have installed the Confirmit Horizons v15.0 Service Pack, you must then immediately apply patch 1 or later. You can download the latest patch from the Confirmit Extranet.

The only supported upgrade path to Confirmit Horizons v15.0 Service Pack is from Confirmit Horizons v15.0 on SQL Server 2008 with the latest Confirmit patch installed. If you are running Confirmit version 12.0, 12.5 or 14.0 on SQL Server 2005, you will have to upgrade Confirmit to version 15.0, and upgrade the database server to SQL Server 2008 before proceeding to upgrade to version 15.0 SP. You can request upgrade assistance by contacting support@confirmit.com or your account manager.

Important

If an upgrade fails, requiring a full uninstall/reinstall of Confirmit on the server, the encrypted connection strings for the Confirmit Deployment and Confirmit Legacy Deployment accounts may have obsolete/invalid values in the configuration tables in the confirm database. Bug reports may ensue as a result of this. The solution for this is to manually run confirmitadministrator.exe in the "CD" and "LD" modes to re-encrypt the two connection strings after reinstallation. Usage of the confirmitadministrator.exe tool is documented elsewhere in this manual.

5.1.1. Confirmit Licensing

Confirmit is protected software and requires a license file to work. Confirmit is not installed with a standard license. If a license cannot be obtained immediately after installation, a trial license can be downloaded from the Confirmit Extranet alongside the Confirmit installation files. Confirmit will function for 14 days on a trial license, and a trial can only be used once per site (i.e. once a trial is started it cannot be suspended).

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 the Obtaining and Installing a License File section on page 42 for more information).

If the trial license is not sufficient until a license file can be created and sent to you, please contact support@confirmit.com at least **two full business days in advance** of the upgrade so a temporary license file can be sent to you for installation on the day of the upgrade. This file will include all paid add-ons, but will be limited to a 14-day period without a Site ID, so a full license file must still be obtained within those 14 days to ensure continued functionality.

If you are upgrading from a version that already uses a license, the existing license should continue working with Confirmit Horizons v15 SP. 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.

5.1.2. Changing the Location of PGP Folders

If upgrading from a previous version of Confirmit via Confirmit v15.0 to v15.0 SP, be aware that if PGP has been used then PGP key rings will have to be migrated from the **<confirmdata>PGP** folder to the **<Confirmit Shared Data>PGP** folder to remain accessible. If this is done, PGP keys will no longer have to be distributed across multiple servers; servers will read the files from the shared data folder when requested.

5.1.3. Keeping Existing Settings

If you upgrade an existing version of Confirmit, existing .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 notes 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.

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.

5.1.4. Stop the Confirmit Task System Service

Due to schema and dictionary updates for the FastObjects database during upgrade, the setup will require exclusive access to the database for a few moments. If the Confirmit Task System service is left running on any servers, it may lock the database during a read and cause the upgrade task to fail. To avoid having to redo the upgrade manually after the setup has completed, stop the Confirmit Task System service prior to upgrading. Also, ensure no tasks are still in the executing state when the setup runs.

Note: In addition, the Windows Installer-based setup program can sometimes experience problems uninstalling and reinstalling Confirmit system services during upgrades if the Services MMC is running during install, so ensure any instances of the MMC for all logged on users are closed before proceeding with the upgrade.

5.1.5. Uninstall Confirmit SDK

Customers using the Confirmit Web Services should uninstall their existing version before upgrading Confirmit. A new version of the SDK supplementing the newest Confirmit version can be obtained by contacting support@confirmit.com.

5.1.6. Back up the FastObjects Database

Important

Always perform a full backup of the FastObjects database before starting an upgrade of Confirmit (see the The FastObjects Database section on page 53 for more information).

5.1.7. Disable FastObjects Incremental Backup

If incremental FastObjects backup is running, this must be disabled before starting an upgrade. Failing to perform this step will render the database unusable after an upgrade as the database upgrade steps cannot be run while the feature is enabled.

5.1.8. Back-up Confirmit System Databases

In addition to the FastObjects database, Confirmit system databases should also be backed up before an update in case a rollback is required (see the SQL Server Databases section on page 52 for more information).

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 dialogue choice is presented for installing 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. The user account under which it runs must be a member of the local Administrators group on the server. Before installation this user must also be granted two permissions in the Local Security Policy:

- 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 the Confirmit and UAC (User Account Control) section on page 76 for more information).

5.2.2.2. Account for the Confirmit Task System Helper Service

Similarly, the Confirmit Task System Helper service is also installed when installing the server with the 'Batch' role. The service itself runs as the Local System user as it only requires local server access, but when running exports it impersonates an existing user account on the system. The account under which it runs must be a member of the local Administrators group on the server, and it should not be the same account that runs the Confirmit Task System service as exports running under the different services may be interfering with each other if running under the same user credentials. 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 from Online Reporting may fail to run.

5.2.2.3. Account for Reading File Library Files from External Location (Optional)

The Confirmit File Library allows users to upload files for using in surveys, such as images, videos, HTM files, CSS style sheets, etc. (File types are configurable in the system administration area in Confirmit.) Files uploaded from the GUI are deployed along with generated thumbnails to a central repository under the **Shared Data** folder, and without thumbnails to registered web servers (servers with any of the Authoring, Survey or Reportal roles) in the site.

If more than one Authoring server is used, or if the Shared Data folder resides on a different server than the Authoring server, the *lisa* virtual directory on Authoring servers must be configured to point to the **Shared Data** folder, and to log on as a Windows user account with permissions to read files over an UNC path in order to access the **Shared Data** folder. In this case, the user must exist both on the Authoring server where the virtual directory is configured and on the server where the **Shared Data** folder resides. The user account created for the Task System service may also be used. The server holding the **Shared Data** folder does not necessarily need to have Confirmit installed.

Note: As virtual directories in IIS are recreated during Confirmit upgrades, any customized settings will be lost and will have to be replaced after a successful upgrade.

5.3. Installation

If all requirements are installed, backups are complete, Windows user accounts have been created and the Confirmit setup has been downloaded from the Extranet, you should be ready to install Confirmit. Once started, the installer will start looking for existing installations, and if any are found, will proceed to upgrade the existing version. One or more of the steps below might be required for the update to continue.

Important

The user performing the Confirmit installation must be member of the local administrators group on the server, either via user or group membership. The setup uses system calls that require full access to the server components, and will fail unless the user has administrator permissions. No information about the installing user is stored during the installation, so the permissions can be removed once the installation is complete (but will need to be added again during future upgrades).

5.3.1. Install / Upgrade Sequence

As a Confirmit upgrade makes changes to database schemas as well as updating the code running locally on each server, it is recommended to upgrade servers in a specific order to prevent new code from creating files that are incompatible with the version still running on servers that have not yet been upgraded. The recommended installation / upgrade sequence is listed below:

- Dedicated FastObjects server(s)
- Authoring servers
- Deployment servers
- Task system servers
- Reportal servers

5.3.2. Confirmit Installation

The first page of the installation is the Welcome page.

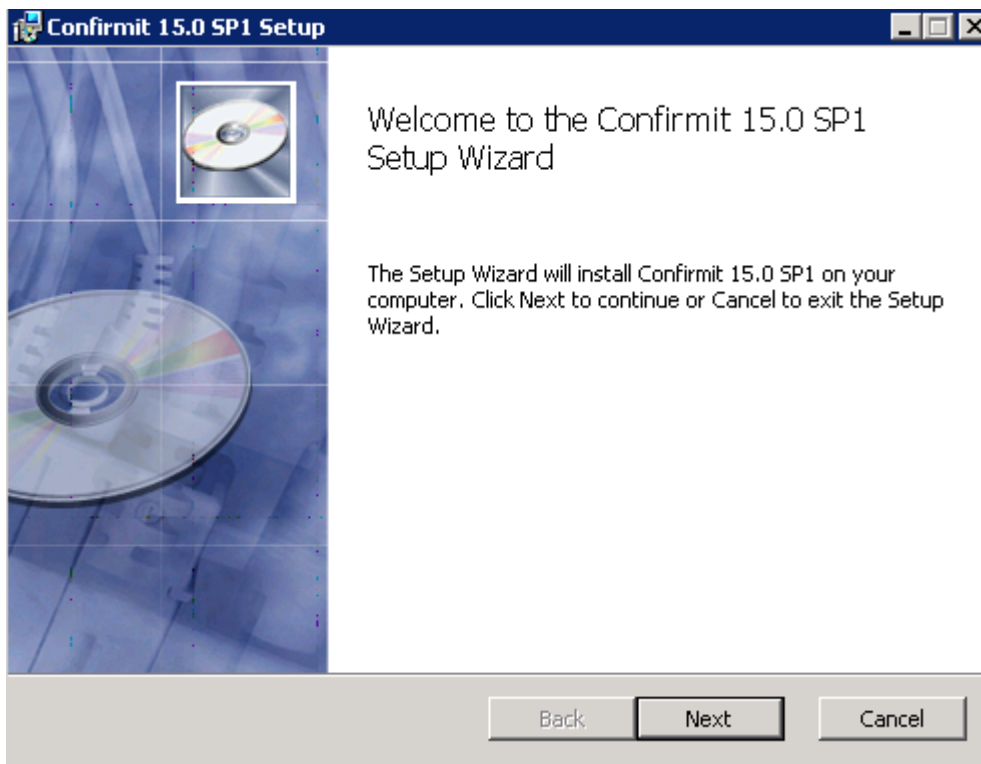


Figure 14 The Welcome page

1. Click the **Next** button to proceed to the next step.
The Destination Folder page opens.

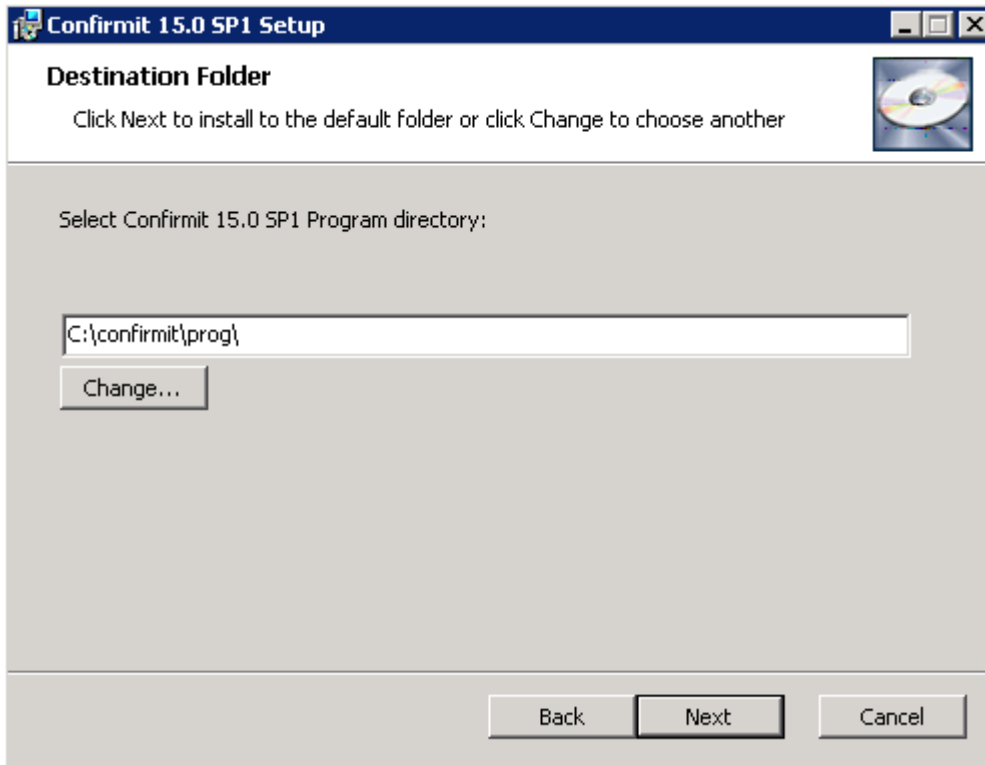


Figure 15 The Destination Folder page - 1

2. Choose a target location for the Confirmit application files (approximately 150MB is required).
3. Click **Next** to proceed.

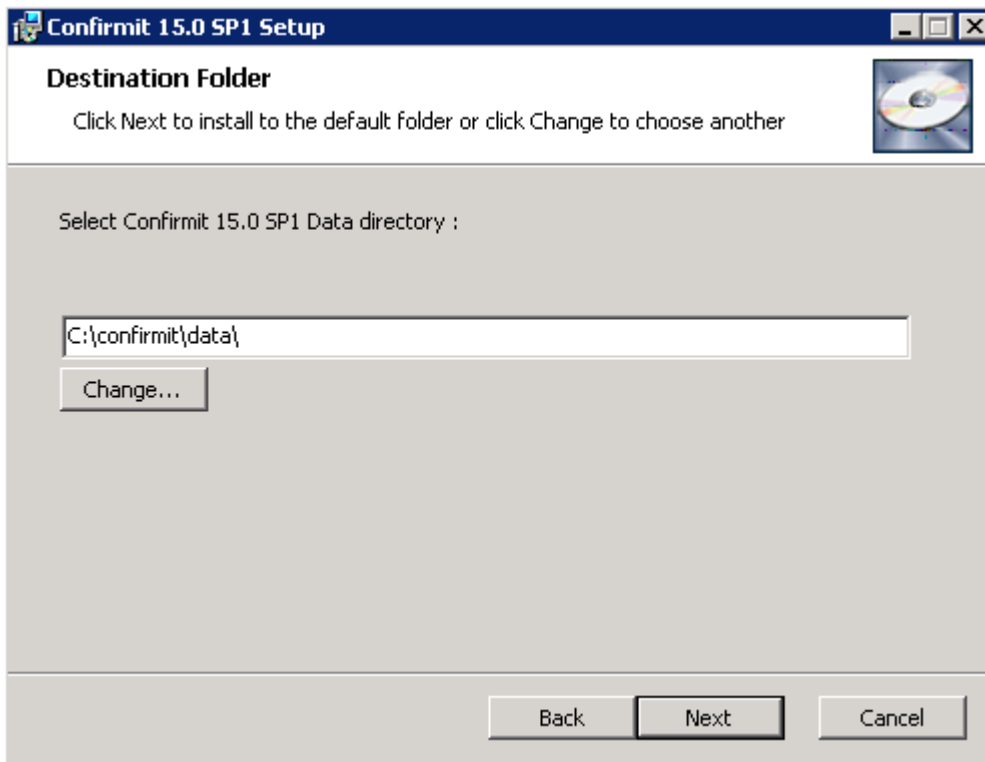


Figure 16 The Destination Folder page - 2

4. Choose a target location for the Confirmit data files (space requirements are subject to application usage; see previous chapters for server sizing recommendations).
5. Click **Next** to proceed.

The Ready to Install page opens.

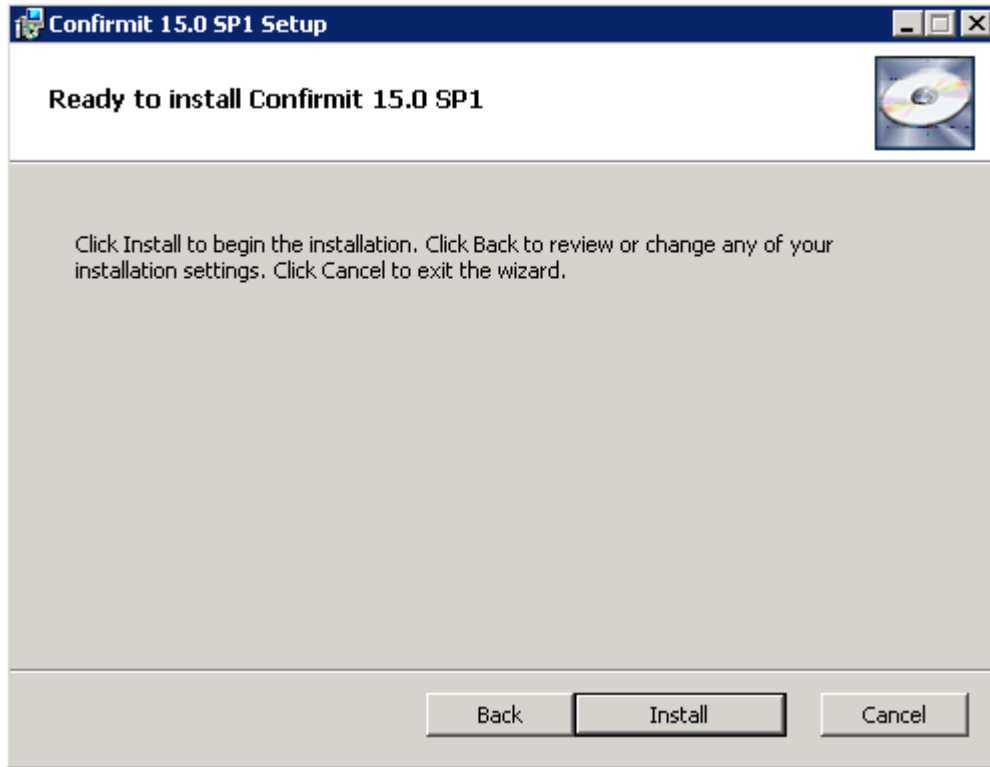


Figure 17 The Ready page

6. Click **Install** to proceed.

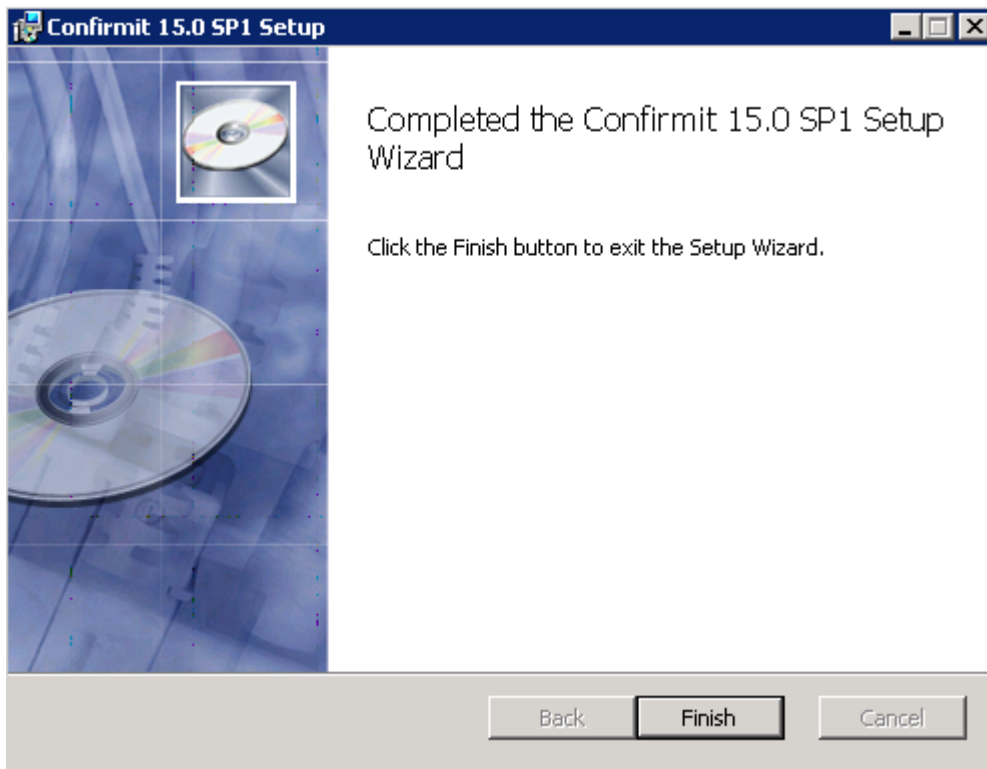


Figure 18 The Completed page

7. Click **Finish** to close the setup.

5.3.3. The Confirmit Configuration Tool

After a successful installation the Confirmit Configuration Tool will be started automatically, and this must be run on every server to apply the upgrade. The configuration tool includes help texts for each parameter to make the installation easier. The tool will also create a shortcut in the start menu and can be run again later to change the roles of the server in question if necessary.

The first page of the Confirmit configuration tool provides a selection of features to be installed.

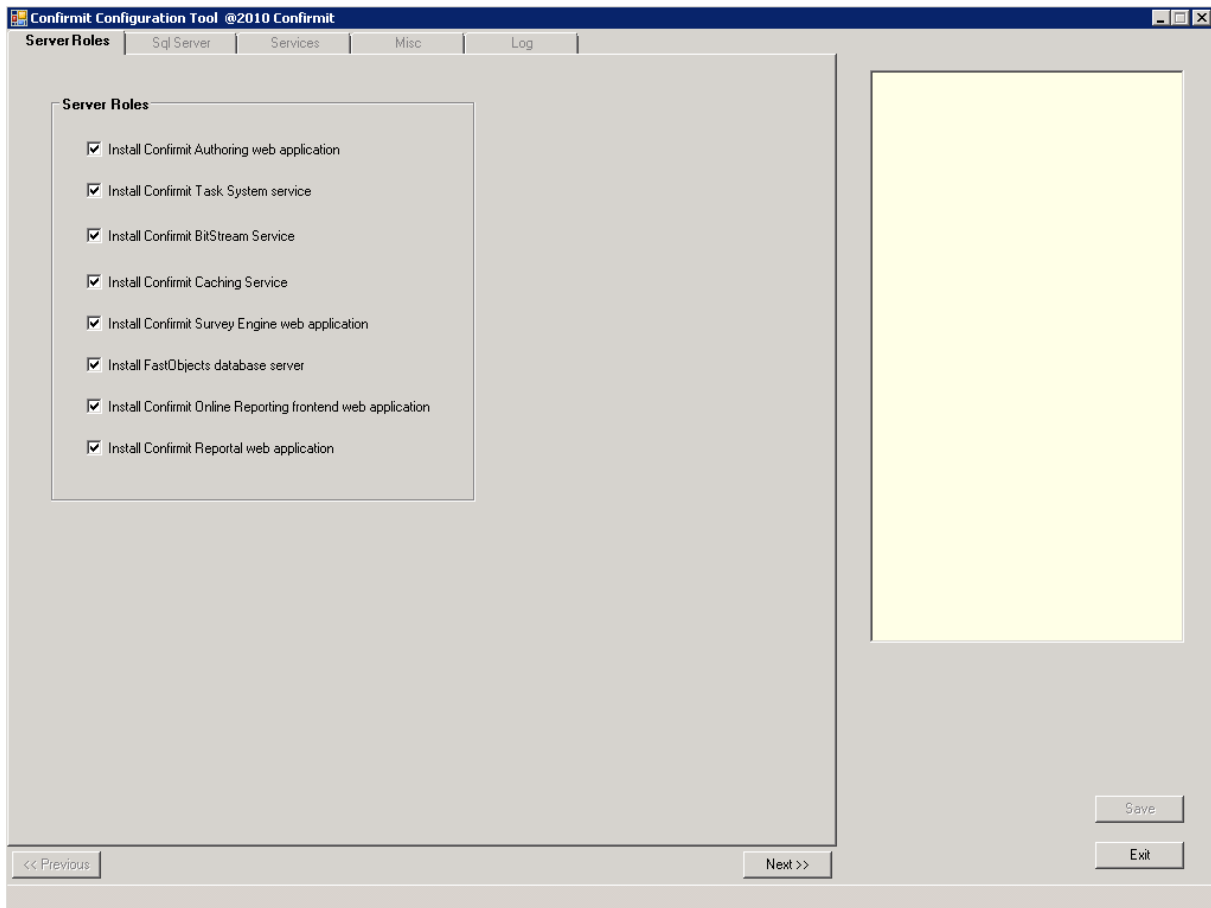


Figure 19 The Confirmit Configuration Tool

1. Select the relevant features for your server and click the **Next >>** button to proceed.

The next page configures details for the connection to the primary SQL Server such as server name, database file locations and login details for SQL Server logins used by Confirmit. If this is an upgrade, the values will be pre-populated from the existing configuration. Note however that the password for the **aspstate** login must be re-entered, also during upgrades, as Confirmit is not able to decrypt this value to use it in the configuration tool.

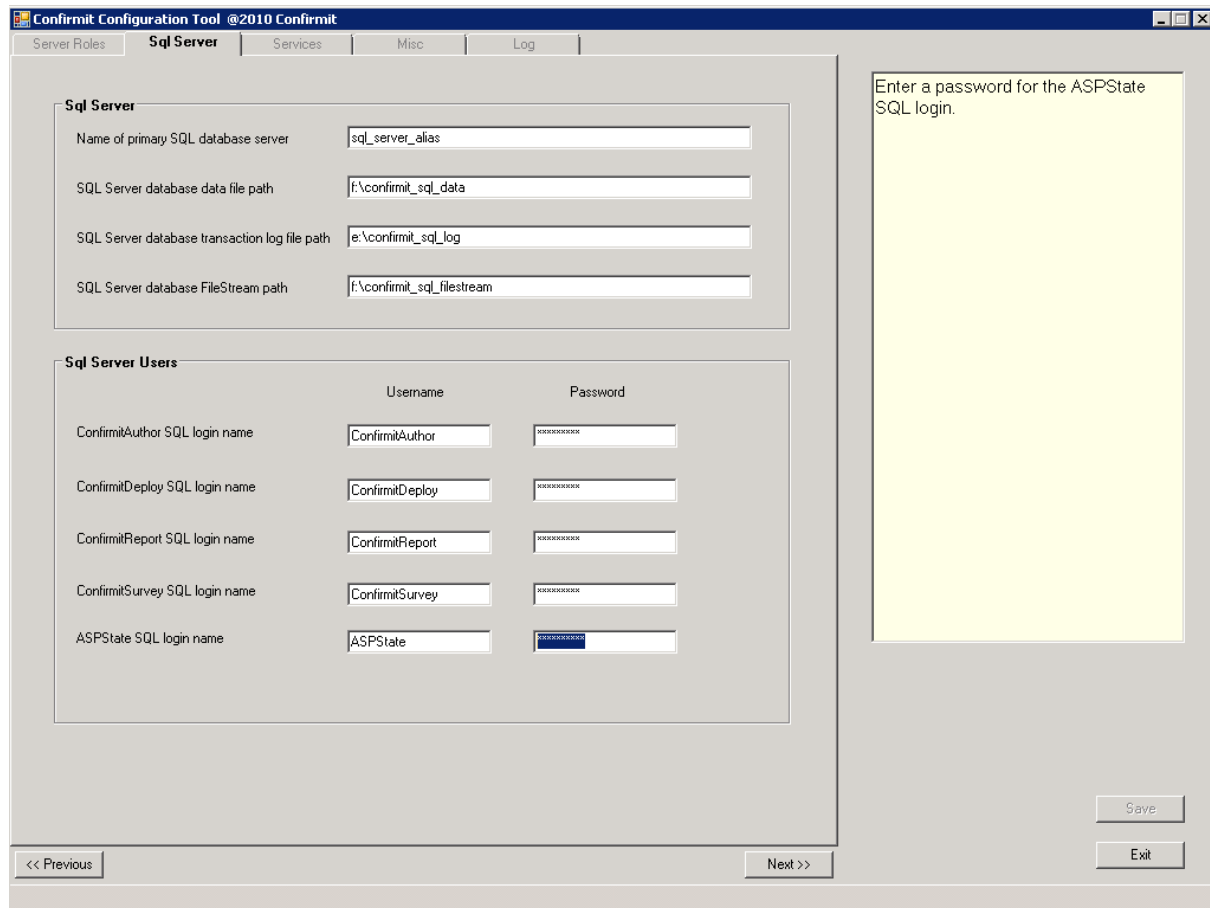


Figure 20 The SQL Server page

Note: The value used for the SQL Server Name or IP Address field is hard-coded into the generated binary survey packages, and if the database server name or IP address changes after installation, the links may be broken and surveys may not be able to connect to SQL Server. For this reason, using the SQL Client Configuration Utility to set up a server alias is strongly recommended as this will provide flexibility for future usage; if the server name/IP changes, simply update the alias and Confirmit will continue to work without any additional changes required. This feature is also helpful in scenarios where deployment servers and batch servers are on different sub-nets and use different network interfaces connecting to the database server.

As Confirmit applications run in 32-bit mode, you will need to add the aliases both in 64-and 32-bit client configuration when installing on a 64-bit server. The two different tools can be started from:

- o C:\Windows\System32\cliconfg.exe for the 64-bit version
- o C:\Windows\SysWow64\cliconfg.exe for the 32-bit version

For the SQL Server database FileStream path, use the location configured during pre-install (see the Creating Data Folders for SQL Databases section on page 30 for more information).

2. Make the settings as appropriate and click **Next >>**.

The next section of the setup covers locations of common Confirmit Services that are used for all servers. When installing on a multi-server installation, the same values must be used for all servers.

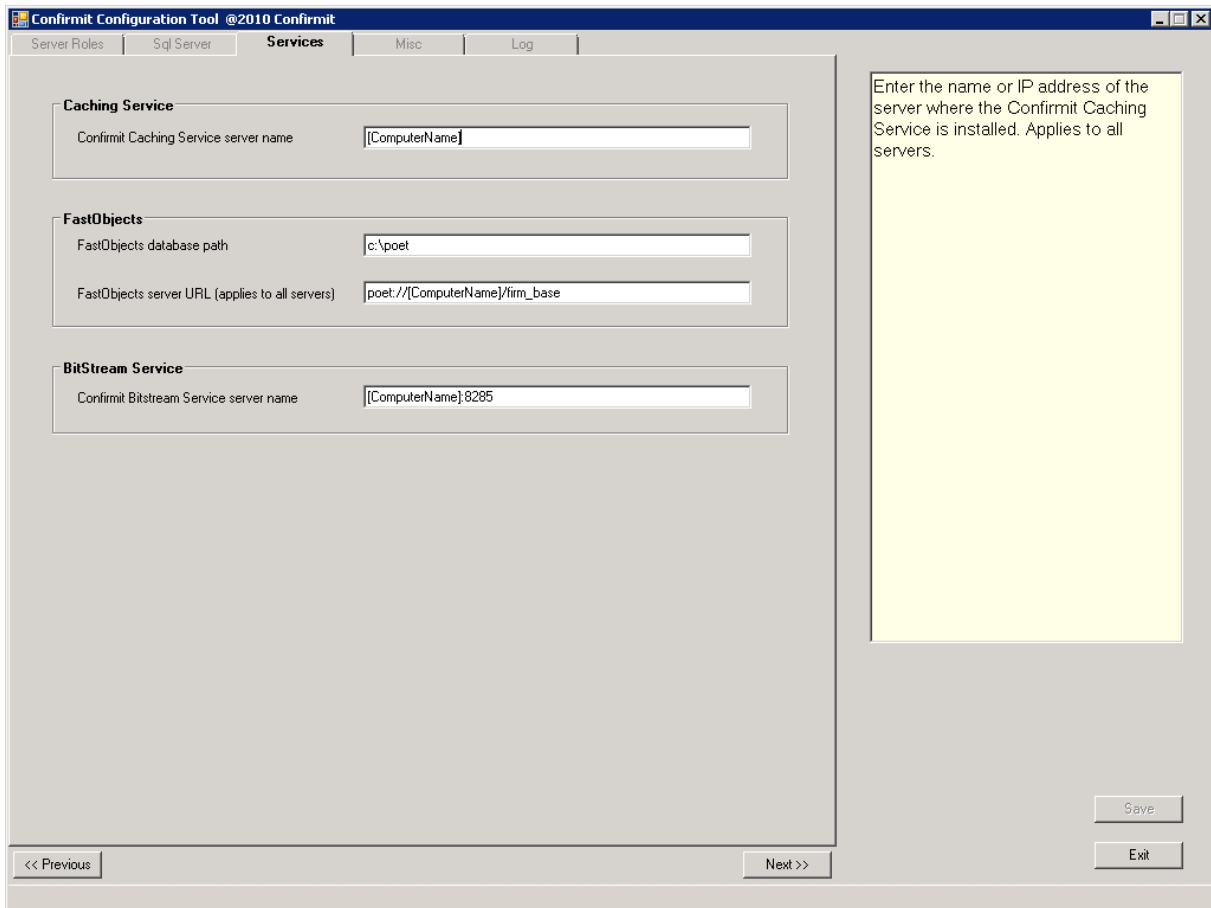


Figure 21 The Services page

3. Make the settings as appropriate and click **Next >>**.

The next section of the setup covers various items that are either unique to the specific server, settings that are optional or conditional based on the feature set, or whether the installation is new or an upgrade.

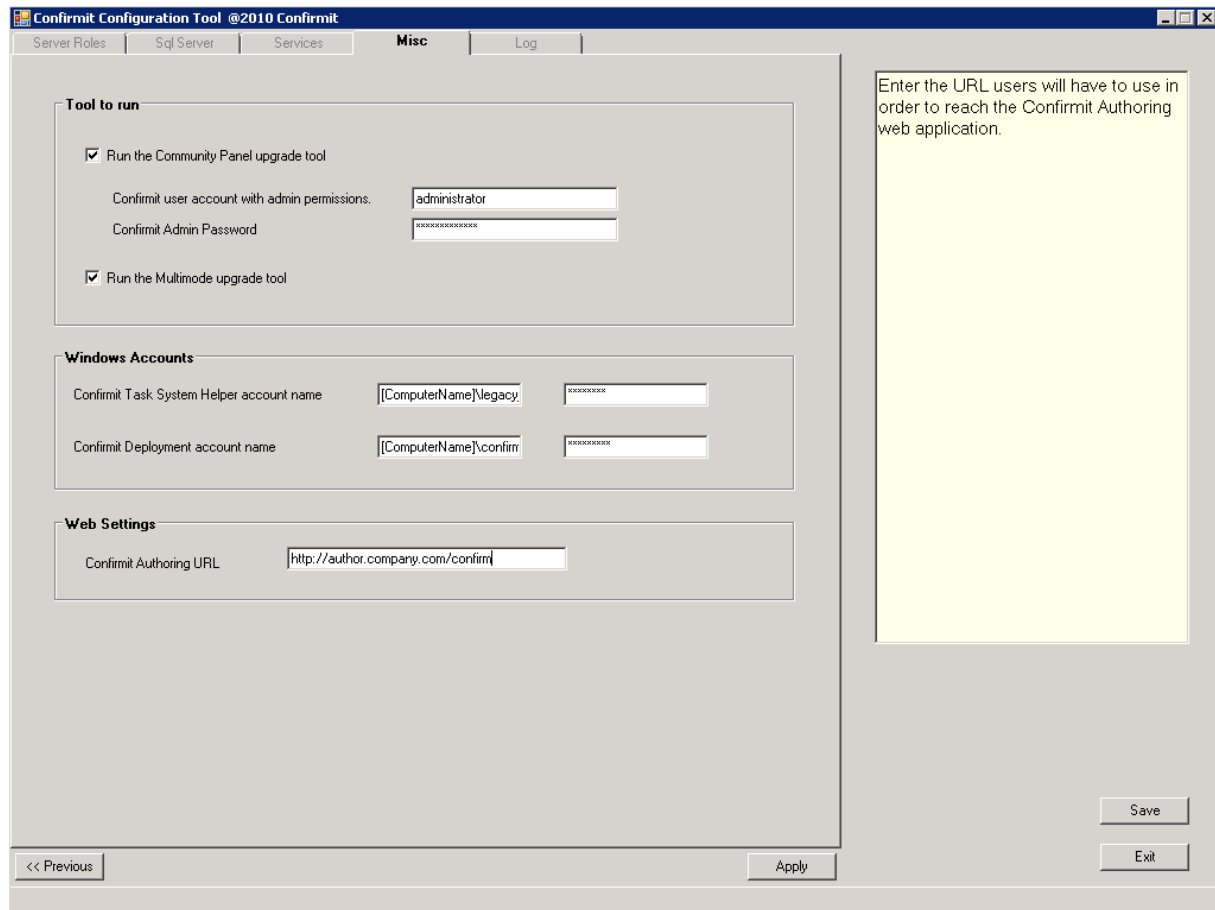


Figure 22 The Miscellaneous page

- The Community Panel upgrade tool only needs to run if the installation is an upgrade from an existing version, and only if the Community Panel feature is licensed. Additionally, for multi-server installations, the tool only has to run once for an upgrade and can be skipped for additional servers.
- The Multimode upgrade tool can be skipped for new installs or upgrades from version 14 or higher.
- The Confirmit Task System Helper account name only needs to be set if the “Install Confirmit Task System service” option was selected on the first screen of the setup.
- The Confirmit Deployment account name must be set for all servers as the configuration tool will apply the values to the Confirmit Task System Service (when this feature is selected for install) as well as to the Confirm.NetworkFileSystemObject COM+ application, which is used for deploying files from a web server to shared file areas.

When all options are configured, the configuration may be saved to an XML file in the **<Confirmit data>\config** folder using the **Save** button. If the tool is started again at a later time, it will load the known configuration items from this file automatically as long as the file exists (all password strings will be stored encrypted in the XML file).

4. Click the **Apply** button to open the final configuration screen, where you can review the selected options before committing the settings.

Server Configuration Summary

Current configuration

Label	Value
Run the Community Panel upgrade tool	False
Run the Multimode upgrade tool	False
ConfirmitAuthor SQL login password	1r8znGFx0dKMNRQM3zAwQ==
ConfirmitDeploy SQL login password	k7x/sK+oB6Knezcw/oLyzg==
ConfirmitReport SQL login password	8wUu5KyVbfjDZiqsn6ISIQ==
ConfirmitSurvey SQL login password	192XUIEi3ubgrZnsRKIZUQ==
ConfirmitAuthor SQL login name	ConfirmitAuthor
ConfirmitDeploy SQL login name	ConfirmitDeploy
ConfirmitReport SQL login name	ConfirmitReport
ConfirmitSurvey SQL login name	ConfirmitSurvey
ASPState SQL login password	nawFVPH9IGDfTaQ063GBA==
ASPState SQL login name	ASPState
Confirmit Task System Helper account password	kQkMA8q3wDstRZMvIc07cg==
Confirmit Task System Helper account name	CO-OSL-TST160\legacydeployer
Confirmit Deployment account password	k7x/sK+oB6L9uAHMriDglA==
Confirmit Deployment account name	CO-OSL-TST160\confirmitdeployer
Confirmit user account with admin permissions.	administrator
Password for Confirmit user account.	Rrd/OwJhZ7bnl4jOKfR5UQ==
Install Confirmit Authoring web application	True
Install Confirmit Task System service	True

Sysadm user credentials

Username:

Password:

Confirmit Helper user credentials

Username:

Password:

Confirmit will be configured with the settings listed above. Please review that the settings are correct.

If you want to make any changes before applying this configuration, you may close this window to edit your settings.

If the settings listed above are correct, enter the credentials of a SQL Server login with SYSADM permissions in the fields below and click "Apply" to apply the configuration. The SQL Server login details will only be used by this configuration tool and will not be saved in the configuration.

Figure 23 Review the selected settings

5. On this screen, supply SQL Server login details for a user with SySADM permissions. Confirmit Helper SQL login credentials also need to be specified here.

Note that the values for these settings are not stored after the configuration tool exits and will have to be specified each time the tool is run.
6. Finally, click the **Apply** button to apply the settings. If you need to change any settings you may also close the window and go back to the configuration screens to make any necessary changes.

After you click **Apply**, the configuration tool will configure the server with the selected parameters. While the tool runs it will show a status update for all the tasks it performs.

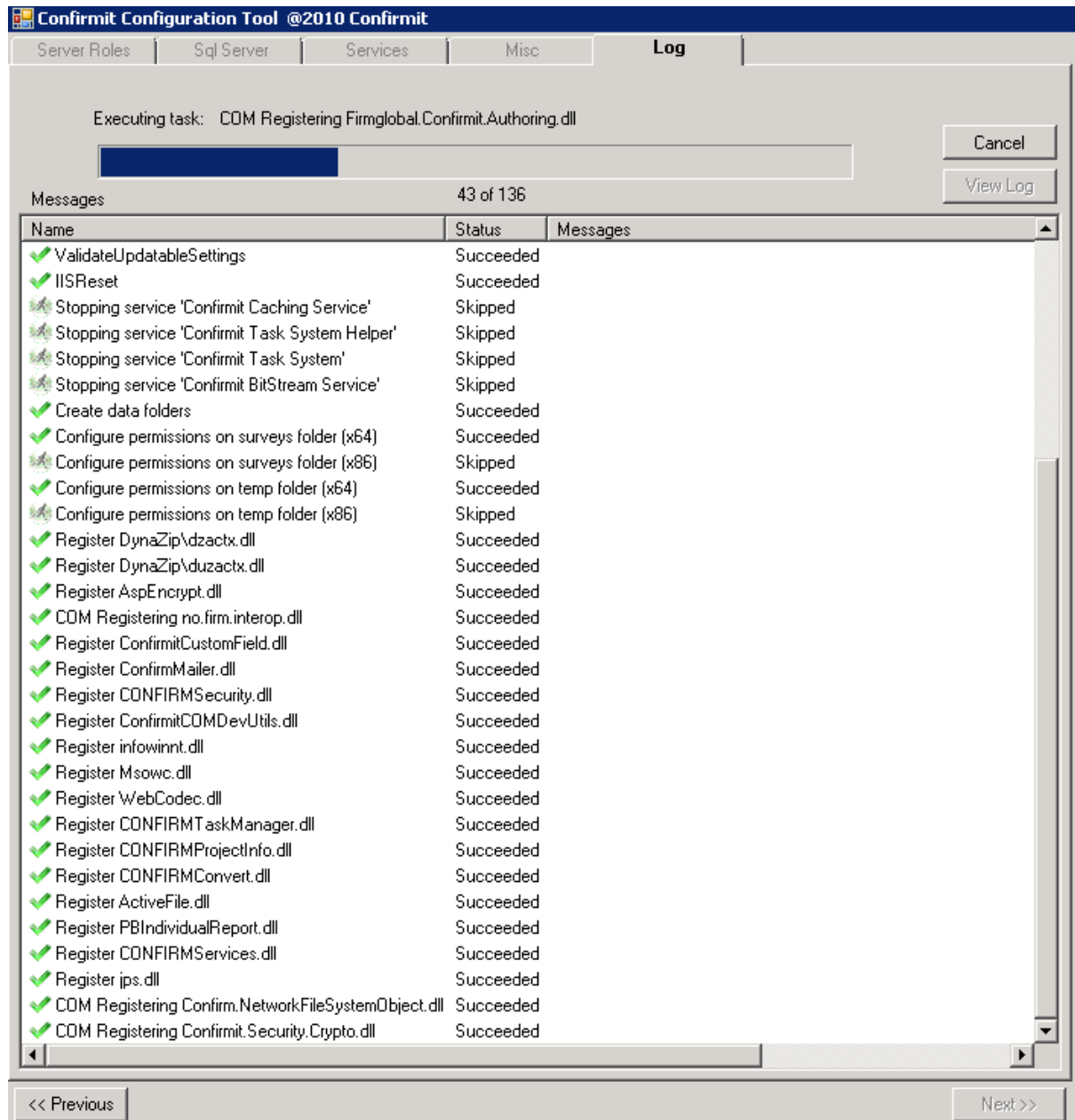


Figure 24 The Status page

If any tasks should fail during the installation, you can view the log for that step by selecting the step and clicking the **View Log** button. Depending on the severity of the error (most warnings can be ignored while errors should be investigated) the configuration tool may have to be run again after the cause of the error is resolved.

The full installation log will be located in the <confirmdata>\log directory after installation.

Note: If errors occur during the installation and you need help from support@confirmit.com to identify the problem, please attach the full installation log to the support request for reference as it may contain information that could be very helpful for troubleshooting the issue.

5.4. Post-Installation Server Configuration

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

5.4.1. Replacing browscap.ini

Important

The browscap.ini file included with Windows Server (also R2) does not have support for Internet Explorer 7 or 8, so in order to use this browser with Confirmit, the file must first be updated on the server. Microsoft decided not to keep the browscap.ini file up to date, so this has fallen on the hands of 3rd parties.

Gary Keith's website regularly maintains and publishes browscaps files with updated browser information, and can be used as a source for downloading. Please note that Confirmit cannot be held responsible for the content on third party sites.

5.4.2. Obtaining and Installing a License File

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

5.4.2.1. 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. If you cannot obtain a license immediately after installation, you can download a trial license from the Extranet from the same location as the installation file. Confirmit will function for 14 days on a trial license, and a trial license can only be used once per site. Once the trial license is loaded into the system, the licensing component starts counting down from the first time the application is used after installation. Once started, the countdown process cannot be paused, even by installing a valid license file in the meantime. Please note that the trial license does not include any add-ons, and functionality will therefore be somewhat limited. Normal Authoring and survey functionality is however maintained.

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.

Important

If the trial is allowed to expire and no valid license file is installed, all Confirmit components except for live surveys will stop working. Installing a valid license later will restore full functionality. Reinstalling Confirmit will not renew the trial license.

5.4.2.2. 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
Online Coding	No	No
Spell Checker	Yes	Yes
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
Database designer	No	No
Reportal report export	Yes (included with Reportal)	No

Reportal public access	Yes (included with Reportal)	No
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
PDF Respondent Report	Yes	Yes
Questionnaire Reviewer	Yes	No
ConcurrentSampling	Yes (included with Community Panels)	No
Code Completion for Scripting	No	Yes (included)
Community Snapshot Panel	Yes (included with Community Panels)	No
Data Processing	Yes	Yes
Sample Only	Yes	Yes
CATI	Yes	Yes
Legacy Reporting	No	No
Database Encryption	Yes	No

5.4.2.3. Finding the Site ID

Once the installation has completed, start the Confirmit Licensing Manager program from **<confirmprog>\licensing\LicenseManagerClient** - the executable filename is **Firmglobal.Confirmit.Licensing.LicenseManager.exe**.

1. Start the Confirmit License Activation application on any of the servers in the site.



Figure 25 The License Overview page

2. Click the **Activation Info** button to launch a popup with information relevant to the installation.

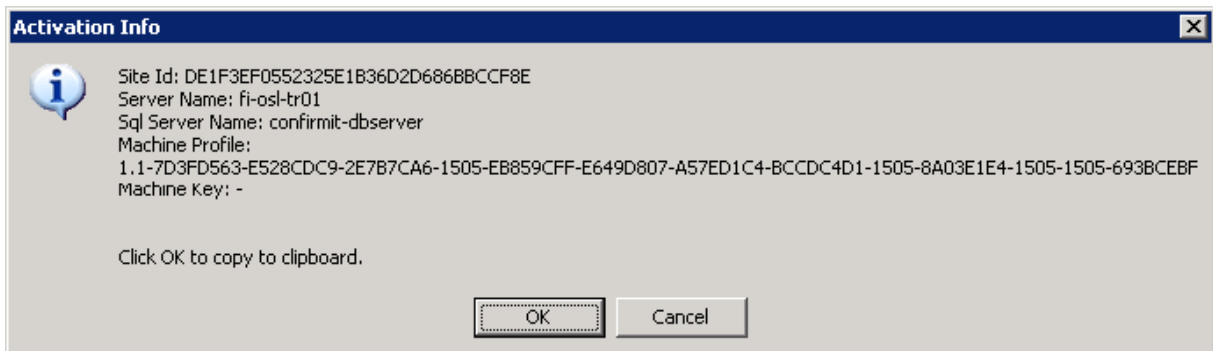


Figure 26 The Activation Info page

3. Click **OK** to copy the information to the clipboard.

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. 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.

5.4.2.4. Installing the License File

When the license file has been obtained from Support:

1. Start the Confirmit License Manager again.
2. Click the **Install license** button, browse for the license file and click **OK**.

This must be done for all servers running Confirmit application code (except for dedicated database servers which do not require a Confirmit license). Using this alternative will also validate the file for use on the relevant system before installing it to the license directory.

After the license is installed, it will be listed in the License Overview list in the Confirmit License Manager.

3. Once the license file has been installed, restart IIS on web servers and the Confirmit Task System service on task system servers to force a license reload on each server.

All issued license files have a specified expiry date. 90 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.

When replacing an existing license file, first delete the old file and then use the procedure described above to install the new one, or use the Confirmit License Manager to replace the existing license. In the unlikely event that the new license file does not work, replace it with the old one again, which should still be working provided the expiry date has not been overshoot. When installing the new license file, it is recommended to first install the license file on one server and verify that the installed license file is working correctly before deploying it across all servers in the site.

5.4.2.5. Confirmit SDK License

Customers using the Confirmit Web Services must install an additional license for the SDK. The procedure for installing the license file is the same, but the license is only required on servers where the Confirmit SDK is installed.

Note: The Confirmit License Manager tool will fail to validate an SDK license file unless the SDK is installed on the server.

5.4.3. 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).

Each server has a unique configuration setting in the Confirmit system configuration (named 'Root Path'), which determines the root target folder for files that Confirmit deploys to it. When installing Confirmit, the setting is by default set to the value of the local data path on the server that was given during installation, plus the addition of '\web'. In order to deploy servers to remote servers in a multiserver environment, this setting needs to be changed to a UNC path which other servers can use to reach the Confirmit data path. We recommend creating a file share on the Confirmit data path for this purpose. In order for Confirmit to function properly, the Confirmit deployment account should be set up with full access to this share. On systems where the user is not a member of the Administrators group (not required on dedicated Authoring, Reportal and Deployment servers), the root drive share (i.e. d\$, e\$) cannot be used as this can only be accessed by members of the Local Administrators group on each server.

After creating the share, 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**.

5.4.4. 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.

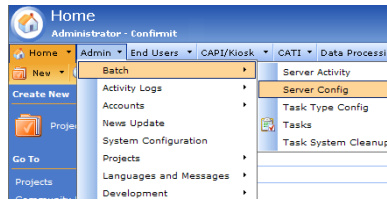


Figure 27 Going to the Admin > Batch > Server Config menu

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

5.4.4.1. 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 three isolation levels; Appdomain, 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, but if you experience task queuing on your system, especially for one or more task types, the number of concurrent tasks for that task type and the percentage wise weight a single instance of the task has on the overall tasks for each server.

5.4.5. Deploying Language and Resource Files

Note: Languages and resources (such as survey error messages) are read from local files on each server rather than from the resource SQL database to reduce overhead. The language and resource files must be deployed initially to all Confirmit servers after installation in order for each server to find the files. This can be done by initiating the task from the Admin > Languages and Messages menu in Confirmit.

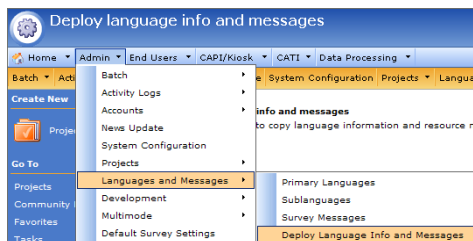


Figure 28 Going to the Admin > Languages and Messages menu

The Languages and Resources Deployer runs as a task, and must be assigned to a task system server in order to run.

If the task fails to deploy to all servers, you can copy the files manually from the task system server where the task ran, over to the other servers within the site. The files are located in the `<confirmdata>\surveys` folder on the server where the task ran:

- Firmglobal.Confirmit.SurveyEngine.ScriptCode.Languages.dll
- languageinfo.bin
- resourcetexts.bin

Note: If languages or messages are modified later, the task must be run again in order for servers to pick up the changes.

5.4.6. BitStream Server Operation Modes

The Confirmit BitStream service can be installed when installing a server with the 'Batch' role selected. The BitStream service provides fastest access to reporting data than SQL Server and provides a proprietary file format and query language developed by Confirmit. Confirmit BitStream Service can operate in three different modes:

5.4.6.1. Single Server, No Sync

The 'Single server, no sync' option is the default option when the Confirmit BitStream Service is installed on a single machine. The server will be responsible for handling BitStream generation and queries for all projects within the site.

5.4.6.2. Single Server With Sync

The 'Single server with sync' option provides an option to synchronize the Confirmit BitStream file repository against a secondary network path (UNC). It is enabled by setting a valid UNC path as the BitStreamSyncRepositoryPath in the general system configuration settings in Confirmit, under the 'Miscellaneous' section. When enabled, the server will generate a local copy of the BitStream files when a file create/update operation is requested, then copy the newly created file set to the repository.

5.4.6.3. Multi Server With Sync

The 'Multi server with sync' option allows running Confirmit BitStream as a 'load balanced' feature. It is similar in setup to the 'Single server with sync' option with the exception that multiple servers will be using the same repository for keeping their local files copies synchronized.

Enable multi server with sync in an existing environment.

- Upgrade Confirmit. Existing BitStream settings will be retained.
- Create BitStream repository in desired path and create a share for this location that can be accessed from all servers running Confirmit BitStream Service.
- Temporarily disable the BitStream Generator task type in Confirmit to prevent tasks from starting
- Set the repository path (UNC) as the BitStreamSyncRepositoryPath.
- Restart the Confirmit BitStream Service. For every BitStream task, the service will generate BitStream files locally and copy the resulting file set to the BitStreamSyncRepositoryPath, from where additional BitStream servers will synchronize. If you want to monitor this process, tracing can be enabled in the <confirmprog>\services\Firmglobal.Confirmit.BitStreamService.exe.config file, producing a detailed log of the operation.
- Re-enable the BitStream Generator task in Confirmit. BitStream sets that are written from now on will be copied to the BitStreamWritePath repository.

Note: When a new BitStream server is added as a Load Balanced server for a project, it starts synchronizing the BitStream files from the repository. Once the synchronization is complete, the BitStream service registers as ready, and the Load Balancing will be in operation. This may take some time based on the BitStream fileset size.

Note: Projects are assigned to specific servers according to a weight model that can be configured (for example, for every 3 projects, server1 is assigned 1 project while server2 is assigned 2 projects). This mode is provided for quickly adding new servers to an existing environment. It is also possible to move surveys that are potentially causing performance issues to a new server.

Note: If you have old projects with BitStream filesets that you wish to load-balance, where you no longer run scheduled BitStream generation tasks, you will need to run a BitStream task manually to synchronize the fileset to the repository after having set this up.

5.4.7. Setting Up the Confirmit File Library

Note: this only applies to systems with more than one Authoring server, or where the Shared Data location is on a different server than the Authoring server, and only if the Confirmit File Library add-on is activated in the license file.

In order for multiple Authoring servers to be able to read file library data from a single location, some settings have to be made.

- On the server containing the **Shared Data** directory:
 - o Create folder **contentupload** in "**<shared data path>**" if it does not already exist.
 - o Create a local user and provide read access to **<shared data path>\contentupload**.
- On each Authoring server:

- o Open the virtual directory <default web site>isa in the IIS Manager and set path "Network directory" to \\<shared data path>\contentupload.
- o Click **Connect as...** and insert the Confirmit Task System user.

Note: If <shared data> resides on an Authoring server, it is not necessary to change the virtual directory for this particular server.

5.5. Upgrading From a Previous Version

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

5.5.1. 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.

5.5.2. Deploying Language and Resource Files

After each upgrade, language and resource files must be deployed to include new error messages and resources.

5.6. 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 the Multiple SQL Server Instances section on page 72 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:

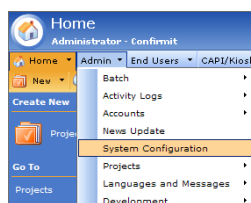


Figure 29 The Admin > System Configuration menu

5.6.1. 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 hover over the setting name.

5.6.1.1. Site-Wide Settings

Site wide settings apply to all servers within the site. They can only 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.

5.6.1.2. 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!)

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

5.6.1.3. 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.

5.7. The Command Line Management Tools

5.7.1. 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>]
```

5.7.2. 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.

5.7.3. 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. 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.

```
ConfirmitAdmin.exe /U:(username) /P:(password) -command <arguments>
```

5.8. Initiating Special Task Types

Some task types affect the whole system and must be initiated by a user with the SYSTEM_ADMINISTRATE permission. There are three tasks where this applies. Note that the links to the pages where the tasks are initiated are not available from the GUI, but must be entered manually in the address field of the browser after you have logged on to Confirmit.

Affected tasks are:

5.8.1. The Project Info Collector Task

The Project Info Collector gathers statistical data from all survey databases on the system, in order to provide background data for the Project Info Export. The collector should run periodically to maintain updated data for the exporter.

To start the task and set a recurring schedule, open the page

http://servername/confirm/project_info_collector_init.asp

5.8.2. The Archiving Task

Archiving detaches project databases and definitions that are no longer in use from the system and stores them on the server. Start the task and set a recurring schedule by opening the page

<http://servername/confirm/authoring/Admin/Archiving/InitArchiving.aspx>

Note: In order for archiving to work properly, the user account that runs the archiving task on the batch server must also exist on the server hosting the SQL Server databases, and needs to have a user profile on the database server in order for the task to be able to impersonate the account on the database server.

5.8.3. The Database Cleanup Task

When a survey is deleted from the GUI, the survey database is not deleted right away. Instead, it is rather marked for deletion in a system database. The database cleanup task drops databases marked for deletion. Start the task and set a recurring schedule by opening the page

<http://servername/confirm/DatabaseCleanupInit.asp>

(NB: The Page URL is case sensitive.)

Note: Starting with Confirmit v15, surveys that are deleted from the user interface are not permanently deleted until they are cleaned by the database cleanup task.

6. 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.

6.1. Backup and Recovery

6.1.1. 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.

6.1.2. 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 built-in NT Backup tool has shown useful to backing up the **<confirmdata>** folders. It works quite fast for large amounts of files and directories, creates a backup file in a general TAR 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 NTBackup 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 NTBackup 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.

6.1.3. 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:

- **[Authentication]** – contains information about Confirmit logins.
- **[Confirm]** – contains task system information, Reportal reports, system configuration, etc.
- **[Confirmlog]** – contains 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.
- **[Custom_tables_design]** – contains hierarchy data in design mode.
- **[Custom_tables_runtime]** – contains hierarchy data in runtime mode.
- **[DataCentral]** – contains temporary data for Data Central processing.
- **[Individualreports]** – contains individual report information (hit lists, single views, etc.).

- **[PanelManagement]** – contains community panel data.
- **[Project_management]** – contains information about projects and companies.
- **[Resources]** – contains languages and system messages.
- **[Templates]** – contains panel templates.
- **[WITemplates]** – contains web interview templates.
- **[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.

6.1.4. The FastObjects Database

Versant's FastObjects database stores all questionnaire and report definitions created in the online reporting (not Reportal reports), as well as ACLs and user information. The database contains an object model that supports cross-referencing that allows object linking on the database level, which works faster than cross-referencing in application code.

The database should be backed up regularly.

6.1.4.1. The *ptadmin.exe* tool

The *ptadmin.exe* tool can be used for many different operations, such as starting backup tasks, setting up replication, checking the state of the database, etc. The commands can be scripted into Windows batch files or scripts for scheduled runs.

The tool is located in the `<confirmprog>\3rd party\poet\bin` folder. This path is inserted in the system `%path%` environment variable during installation, so it should be possible to run the command from any directory on the server.

Usage of the tool normally requires specifying which server and database to run the command against. This is specified by the `-server` and `-base` logical database arguments. When running the command locally on the server where the FastObjects database is installed, the `-server` argument can take the parameter `localhost` and `-base` should always be `firm_base` for the Confirmit FastObjects database logical name. When running a command, `ptadmin -server localhost -base firm_base` should always be included with the desired command.

6.1.4.2. Backup

6.1.4.2.1. Concepts

The FastObjects database server will lock its database files for exclusive writing while the service is running, so the files cannot be backed up by normal file system copy tools. The FastObjects system includes support for setting both online and incremental backup modes though, which will allow the administrator to create a backup plan that won't interfere with production availability. It is also possible to back up the database offline if this is desired, but Authoring, Reportal and Task System modules will then be unavailable for the duration of the backup. Incremental backup can be used regardless of the strategy for full backups.

6.1.4.2.2. Full Offline Backup

The full offline backup refers to a full file copy of the database folder `<poetdir>` (including both the database `(\firm_base*.*)` and dictionary `(\firm_dict*.*)` as the database cannot function without a matching dictionary. Offline backup requires stopping the "FastObjects Server 10.0" system service where this is running before backing up the `<poetdir>` folder, and then restarting the service once the backup is completed to restore connectivity. Authoring, Reportal, Translator and tasks will be unavailable during a full offline backup.

6.1.4.2.3. Full Online Backup

Full online backup is similar to the full offline backup, but differs in that it does not require the system service to be stopped, and full connectivity will be maintained for application modules while the backup is running. This requires setting the database in an online backup mode, where any incoming changes that would normally be stored in the database, will be temporarily stored in server memory as long as the database is in online backup mode. Once the backup is completed and the database mode is set back to normal, the database system will automatically flush changes down to database files on disk.

Note: If the backup is scripted and the database is not set back to online mode by the script after completion, the server will continue to store incoming transactions in system memory. If available memory runs out or the server should fail while the server is in backup mode, any changes made since the database was placed in backup mode might be lost. When scripting backups, the scripts should log the status of its operations so an administrator can check the status later without having to query the database directly. Consult the appendix for script examples.

Note: If database recovery is enabled (as it should be at all times), all incoming changes (also during backup mode) will also be flushed down to a recovery file, and if the server process should end abruptly, the database will automatically perform a full recovery the next time it is started. This might take a while to complete, depending on the database and recovery log file size, but it keeps the database consistent in the event of an unforeseen server failure.

To set the database in online backup mode, run this command on the server hosting the FastObjects database:

```
ptadmin -server=localhost -base=firm_base -backup=begin
```

The **<poetdir>** folder can then be backed up like if the database was offline. Remember to backup both the database and dictionary.

To end the backup mode, run this command:

```
ptadmin -server=localhost -base=firm_base -backup=end
```

6.1.4.2.4. Incremental Backup

In addition to full backups, which could be run weekly, bi-weekly or even daily with small databases, incremental backup can be enabled to provide a smaller, more frequent backup that only contains changes since the last full backup. Incremental backup can be started at any time, but should preferably be started right after a full backup for consistency. Once initiated, the database server starts writing all changes to an increment file in addition to the database itself. With the relatively small amounts of data changes committed to the FastObjects database, the overhead should be negligible on a server in line with the recommendations (see the Task System Servers (Backend) section on page 10 for more information).

Incremental backup can be started with the following command (on the server hosting the FastObjects database, while the service is running):

```
ptadmin -server=localhost -base=firm_base -ibackup=begin -
location=<folder>
```

<folder> determines where the increment file should be created, i.e. d:\increments. The command would then be:

```
ptadmin -server=localhost -base=firm_base -ibackup=begin -
location=d:\increments
```

When issued, the database will create a numbered increment file in the given folder, starting with 'f0000000.ptd' if the folder contains no previous increment sets.

The increment file is locked for writing by the database, and, like the database itself, cannot be backed up externally. In order to release the file for backup, the database can be instructed to start writing to a new, consecutively numbered increment file. This is achieved by running the following command:

```
ptadmin -server=localhost -base=firm_base -ibackup=switch
```

The "switch" command can be issued as often as required. The database will keep track of which files belong together as long as they are all created within the same incremental backup session. In order to be able to restore the database up until a specific point in time, the database can restore increment files from a specific folder when issuing a restore command, including all files from the incremental backup session or files only up to a specific point in time. This information can be useful when considering a frequency for switching increment files (daily, several times a day, hourly, etc.)

To prevent huge data sets required for restoring a recent operation (i.e. with a monthly full backup and 25 days' worth of increment files, it would take a while to restore all transactions from the increment files) it would be recommended to start a new incremental backup session whenever a full backup is run.

To end incremental backup mode the following command can be issued:

```
ptadmin -server=localhost -base=firm_base -ibackup=end
```

The server will release the current increment file for external backup, and a new incremental session can be started following a new full backup (online or offline).

6.1.4.3. Restore

Restoring the FastObjects database from a full offline or online backup quite simply is a matter of replacing the current database files on the server with a copy from backup while the service is not running. Make sure incremental backup is not running when attempting to restore a database. Please note that partial restores are not directly supported by the database system due to its object structure.

Note: If you restore a full database backup, all questionnaire definitions, report definitions, users and ACLs will be reset to that of the time of the backup. If changes have been made to surveys, and database/WI files have been recompiled and re-launched in the meantime, new questions added between the point of backup and the point of restore might no longer be accessible from the GUI.

6.1.4.3.1. Restore with Incremental Backups

If you have been running incremental backups, you can restore the database back up until the last available increment file before the event that caused the need for a restore. To achieve this, first restore the latest full backup, and place all required increment files from that particular incremental backup session in a folder on the server, then run the following command to restore the database up to the increment point:

```
ptadmin -server=localhost -base=firm_base -ibackup=restore -  
location=<folder>
```

Similar to the initiation of incremental backup, the <loc> parameter again points to the path of the increment files, i.e. d:\increments, which would lead to the following command example:

```
ptadmin -server=localhost -base=firm_base -ibackup=restore -  
location=d:\increments
```

The procedure might take some time to complete, depending on the database size and amount/size of increments included in the restore process.

6.1.4.3.2. Partial Restore

We mentioned earlier that partial restores are not directly available, but it is possible to restore portions of a database, particularly surveys deleted in error (which will most likely be the main reason for a restore), although it requires some extra work.

To achieve this, you can either stop the FastObjects system service and move the current production database to a temporary location (remember to disable incremental backup first,) placing the backup in its original location (and restoring increments if required,) then making any required exports from the backup database before replacing the temporary backup database with the production database again. Once the operation has been completed, the project can be imported back into the production database with the original or new project ID using the ConfirmitAdmin tool (see the The ConfirmitAdmin Tool section on page 50 for more information).

6.1.4.4. Confirmit Upgrades

Before upgrading Confirmit, incremental backup must be disabled to prevent problems updating the database schema, which can fail if the database has any agents attached to it and in the worst case crash the whole setup, potentially requiring a reinstall or manually running upgrade scripts. Incremental backup can be re-enabled when the upgrade is complete.

6.2. Monitoring and Systems Maintenance

6.2.1. 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.

6.2.1.1. 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').
```

6.2.1.2. 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 hasn't 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) for 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.)

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.

6.2.1.3. 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. A working example script to add a procedure for automating this job is included in the appendix section.

6.2.1.4. 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 included in the appendix section.

6.2.1.5. 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. Scripts to do this are included in the appendix section.

6.2.1.6. Shrinking Databases

If you experience that the size of a survey database starts to grow excessively and your free disk size becomes a problem, you should check if shrinking the database could free up some space.

6.2.1.6.1. SQL Server LOB Compaction

SQL Server 2008 offers functionality to compact pages that are sparsely populated. Run the following statement to deallocate unused pages:

```
ALTER INDEX <indexname> ON <database>.dbo.<tablename>  
REORGANIZE  
WITH (lob_compaction=ON)
```

Then free the space to the OS by running:

```
use [survey_pXXXXXXXX]  
go  
DBCC SHRINKFILE (1, 1)
```

6.2.2. 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.

6.2.3. 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.

6.2.4. 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.

6.2.4.1. 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
```

6.2.5. IIS Configuration

6.2.5.1. IIS Hardening

IIS7 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.

6.2.5.1.1. 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>

6.2.5.2. 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.

6.2.5.2.1. IIS Log File Maintenance

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.

6.2.6. 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.

6.2.6.1. 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>
```

6.2.6.2. 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.

7. Advanced System Configuration

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

7.1. IIS Advanced Configuration

7.1.1. Configuring IIS to Enforce Compatibility Mode for IE8 Browsers

With the introduction of Internet Explorer 8, the browser is partially incompatible in its 'native' mode with current versions of Confirmit due to changes in the rendering engine. In order to support web sites that have been developed specifically to support settings used in Internet Explorer 7, the browser includes a compatibility mode that can be enabled on the client side, or enforced on the server side in IIS for any connecting client.

The compatibility mode can be set in IIS by adding a custom HTTP Response header that the client browser will read and interpret. The header name should be 'X-UA-Compatible' and the value should be 'IE=EmulateIE7'. The setting can be applied to individual web sites or to the top 'Sites' level.

7.1.2. Configuring the ASP Response Size Limit in IIS7

If users experience an error while saving an export template in Authoring it is quite possible that the ASP page buffer is exceeded. The users will then get a 403 error saying: Page size larger than the allowed size. The solution is to increase the Maximum Request Entity Body Limit in the ASP settings for the web site. The default value is 200 000 bytes.

In our hosted environments, to resolve this problem we have changed the value to 2097152 bytes (2MB). Configure as necessary.

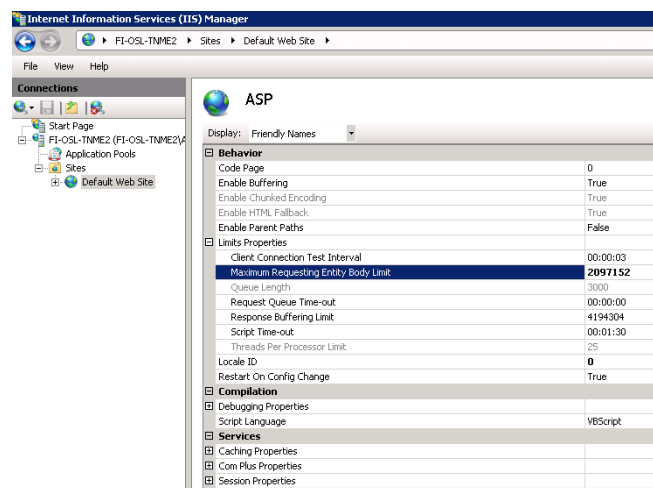


Figure 32 Configuring the ASP Response size limit

7.1.3. 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 two On-Demand sites, both sites have 10GB 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.

7.1.3.1. Configuration

How to set up the ConfirmitWIXAppPool recycle settings:

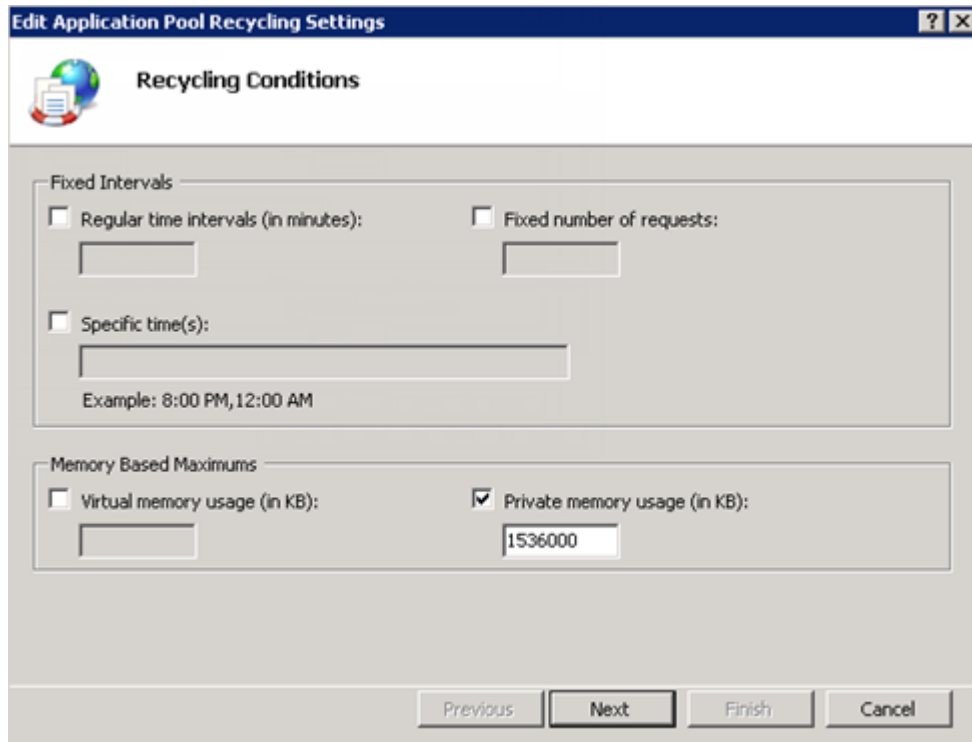


Figure 33 The Recycling Conditions page

1. Disable recycle every xx minutes.
2. Set to recycle at ~1536 MB used memory.

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.

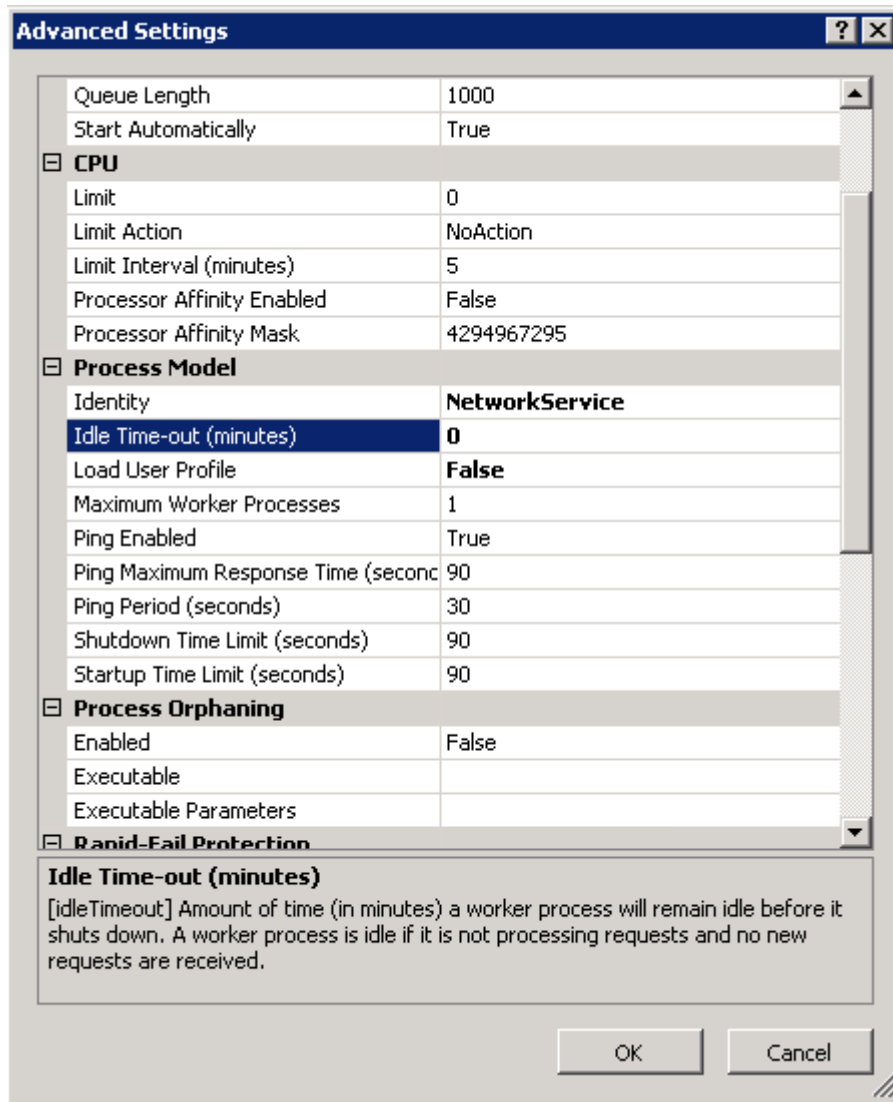


Figure 34 The Advanced Settings page - 1

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 handling surveys). Setting the value to 0 will disable idle time-outs.

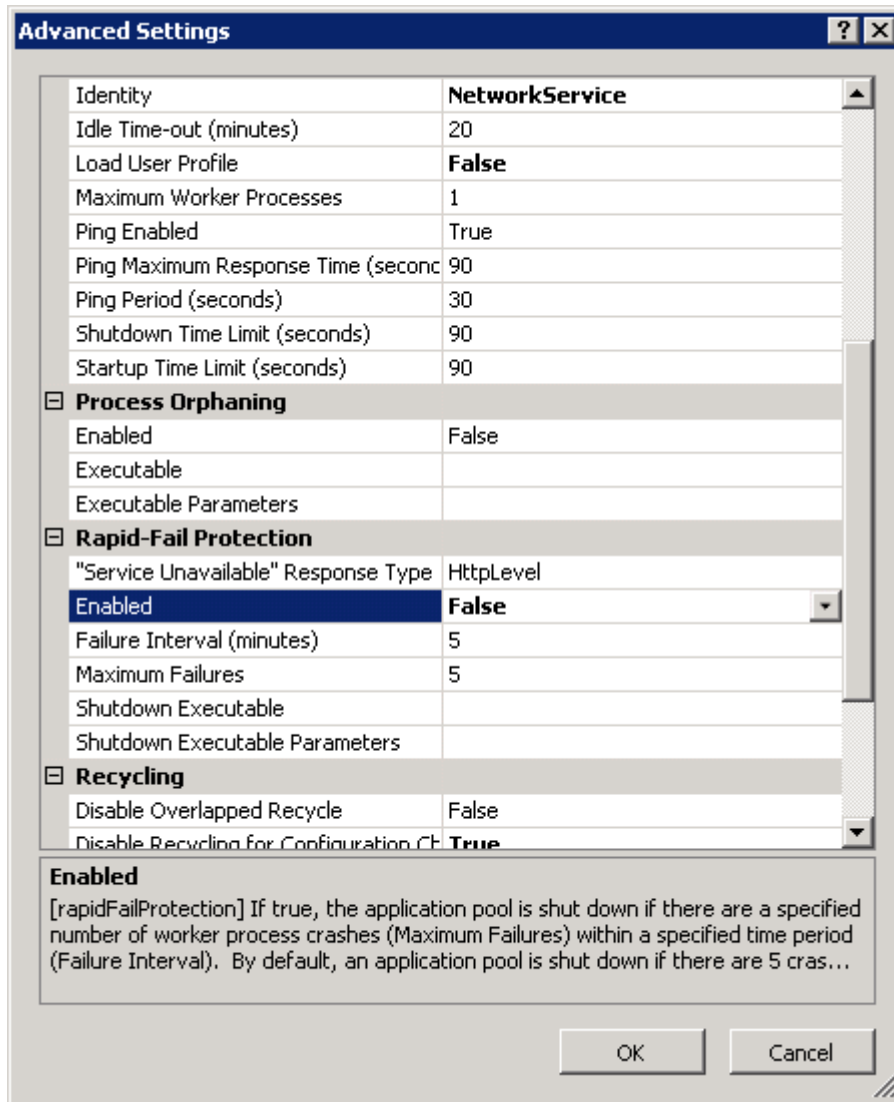


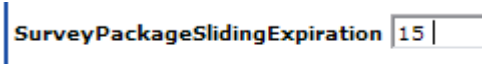
Figure 35 The Advanced Settings page - 2

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).

Other Application Pool Recommendations

Authoring Servers:

- ConfirmitAuthorASPAppPool
 - o Idle Time-out (minutes): 90 (original 20)
 - o Recycling
 - Private Memory Limit (KB): 921600
 - Regular Time Interval (minutes): 0
- o ConfirmitAuthorAppPool
 - o Idle Time-out (minutes): 90 (original 20)
 - o Recycling
 - Private Memory Limit (KB): 819200
 - Regular Time Interval (minutes): 0

Reportal Servers:

- ConfirmitReportalAppPool
 - o Idle Time-out (minutes): 20 (not changed)
 - o Recycling
 - Private Memory Limit (KB): 819200
 - Regular Time Interval (minutes): 0

7.1.4. 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.

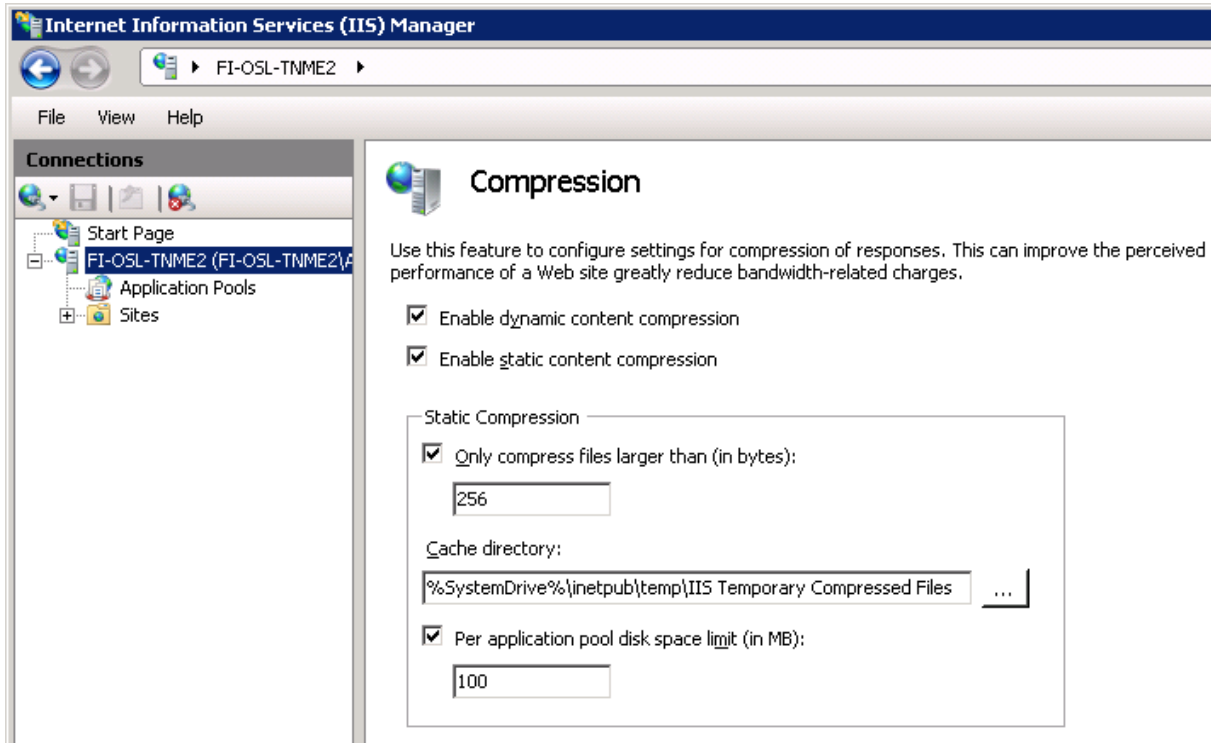


Figure 36 Compression settings for servers

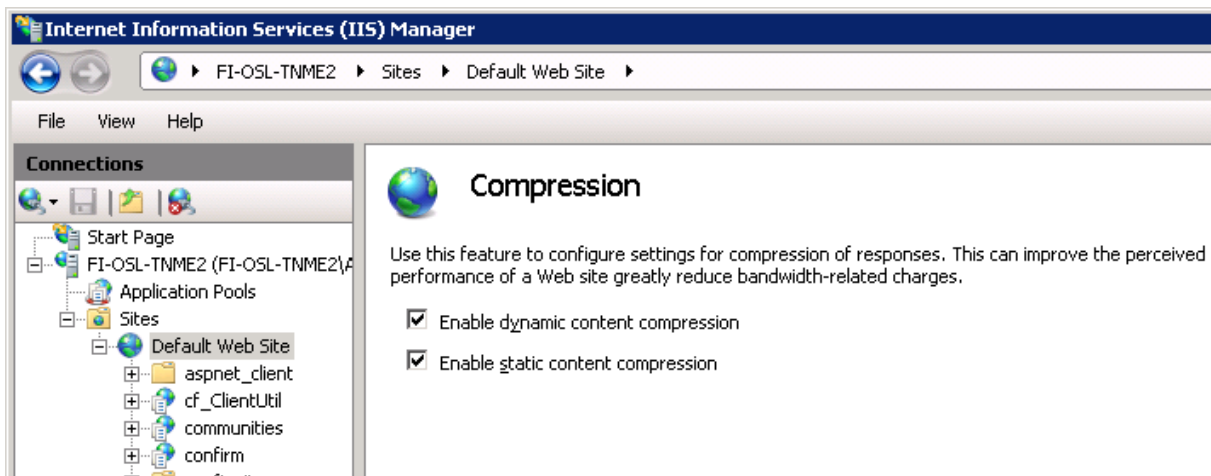


Figure 37 Compression settings for web site

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>.

Confirmit strongly recommends that you take 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="D:\inetpub\temp\IIS Temporary Compressed
Files" maxDiskSpaceUsage="256" minFileSizeForComp="4096">
  <scheme name="gzip" dll="%Windir%\system32\inetsrv\gzip.dll"
doDynamicCompression="true" doStaticCompression="true"
dynamicCompressionLevel="7" staticCompressionLevel="8" />
  <staticTypes>
    <add mimeType="text/*" enabled="true" />
    <add mimeType="message/*" enabled="true" />
    <add mimeType="application/javascript" 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>
```

7.1.5. 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: /
```

7.1.6. 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	Average CPU time in milliseconds used to execute statements

	milliseconds	
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

7.1.7. 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:

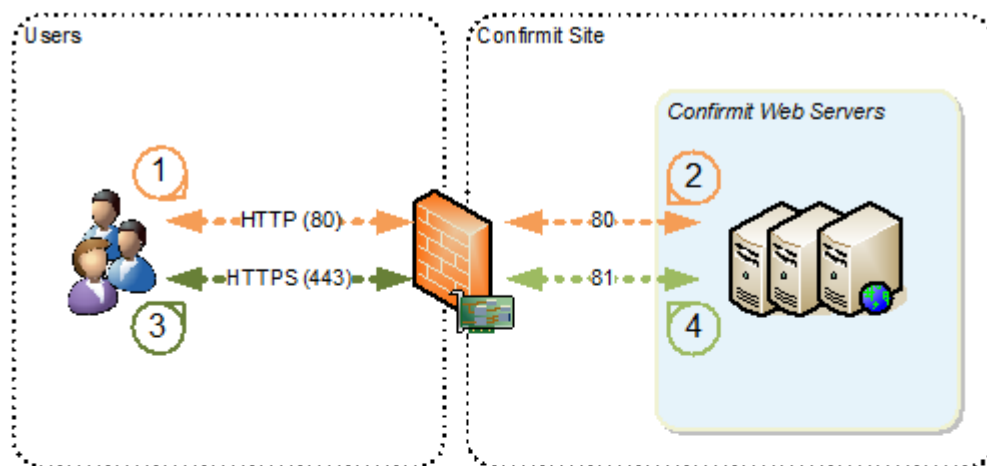
Category	Countername	Description
Monitor	Write - # Currently Runinng	Current number of BitStream writing operations.
Monitor	Write - # Total Started	Total number of BitStream writing operations since service start.
Monitor	Write - % Success	Current percentage of successful BitStream write operations.
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.

7.2. 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, but we would recommend against using port 443 - we use 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.**



- ① Clients connects to Confirmit on HTTP (port 80)
- ② Firewall allows port 80 trough to Confirmit which redirects client to HTTPS
- ③ Client reconnects to Confirmit on HTTPS (port 443)
- ④ SSL Accelerator translates 443 to 81 and passes traffic to Confirmit. As the internal port matches the SSLAcceleratorPort Confirmit knows this is encrypted traffic

Figure 38 Using Confirmit with a load balancer

7.3. Database System Redundancy

7.3.1. FastObjects in Windows Cluster

The cluster must be set up with a cluster name, and at least one disk; preferably two separate physical LUN's if possible (one for the database and one for increments), before installing Confirmit.

Recommended disk configurations are RAID-10 with as many spindles as possible.

Install on FastObjects servers before any other Confirmit Servers.

7.3.1.1. First Cluster Server

Make sure the resources are online on the First Cluster Node.

Run the Confirmit installation as you would on a normal Authoring server. Use the cluster disk resource as the target drive for FastObjects.

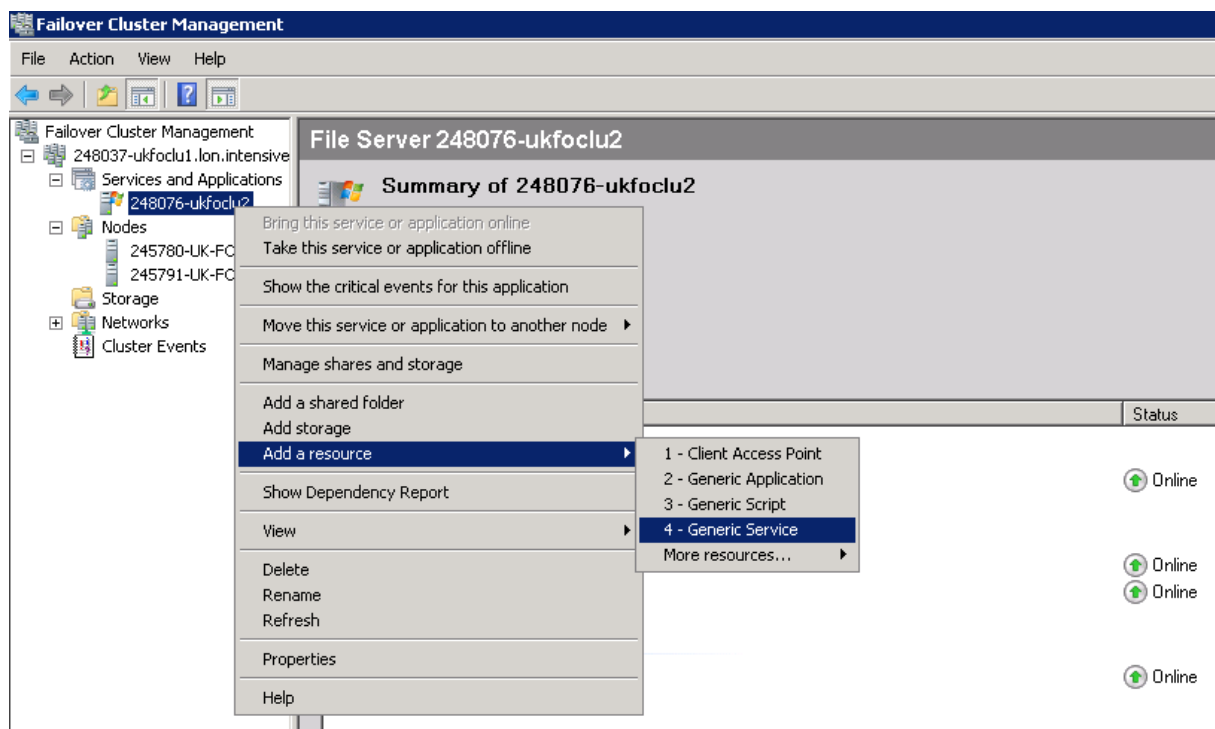
Note: Use non-clustered drives for Confirmit program and data folders!

7.3.1.2. Second Cluster Server

1. Fail over resources to the Second Cluster Node.
2. Run the Confirmit installation as you would on a normal Authoring server. Use the cluster disk resource as the target drive for FastObjects.

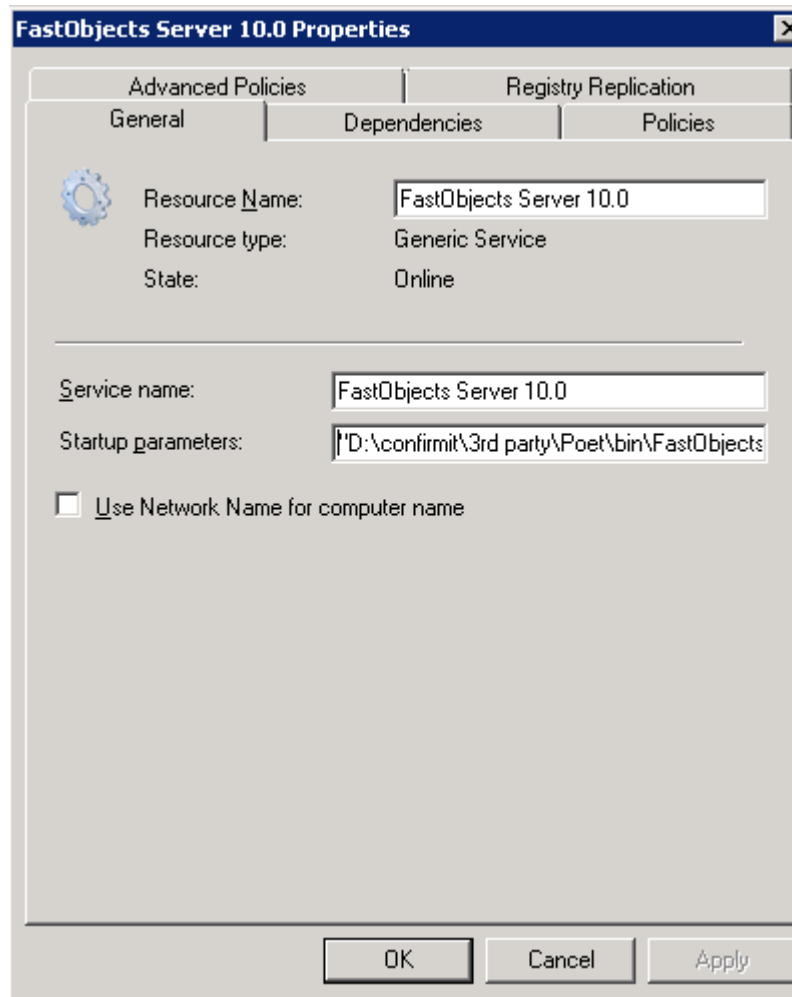
Note: Use non-clustered drives for Confirmit program and data folders, and use same install path for Confirmit for all cluster servers!

3. Repeat above steps for any additional Cluster Nodes.
4. Add the FastObjects service as a Generic Service resource in the cluster.



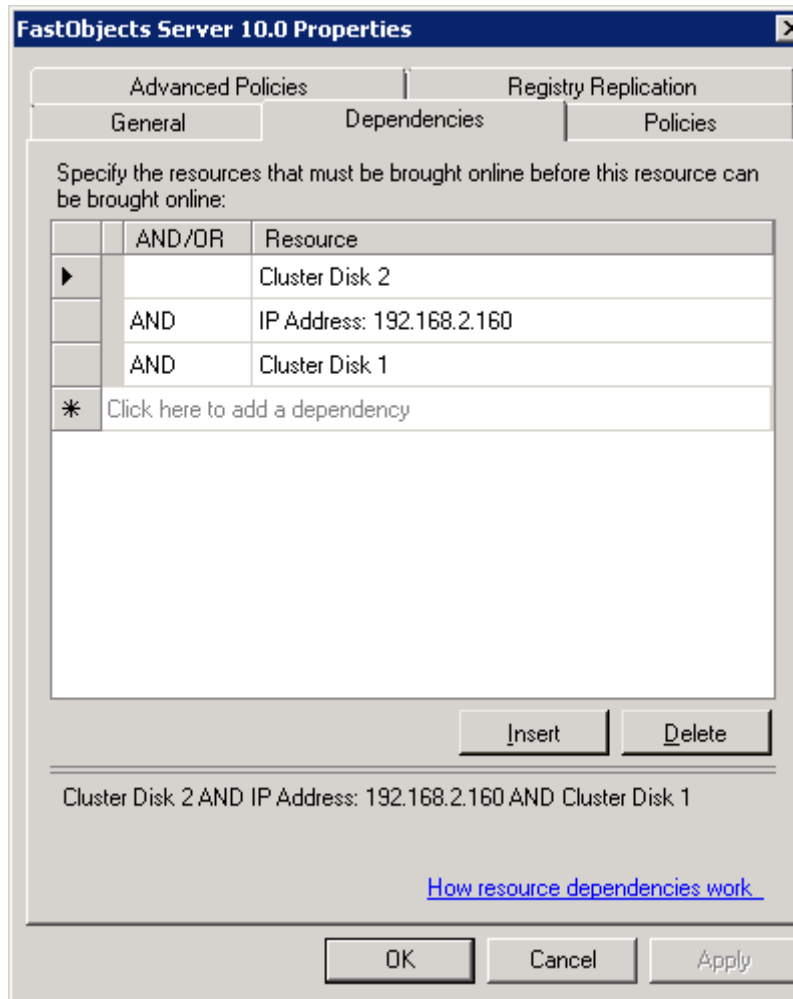
5. Modify the General parameters of the service to reflect the local paths (may be messed up by the install, so verify).

I.e. if Confirmit was installed to D:\confirmit, this should be: "D:\confirmit\3rd party\Poet\bin\FastObjectsServer.exe" -config "D:\confirmit\3rd party\poet\bin\ptserver.cfg".

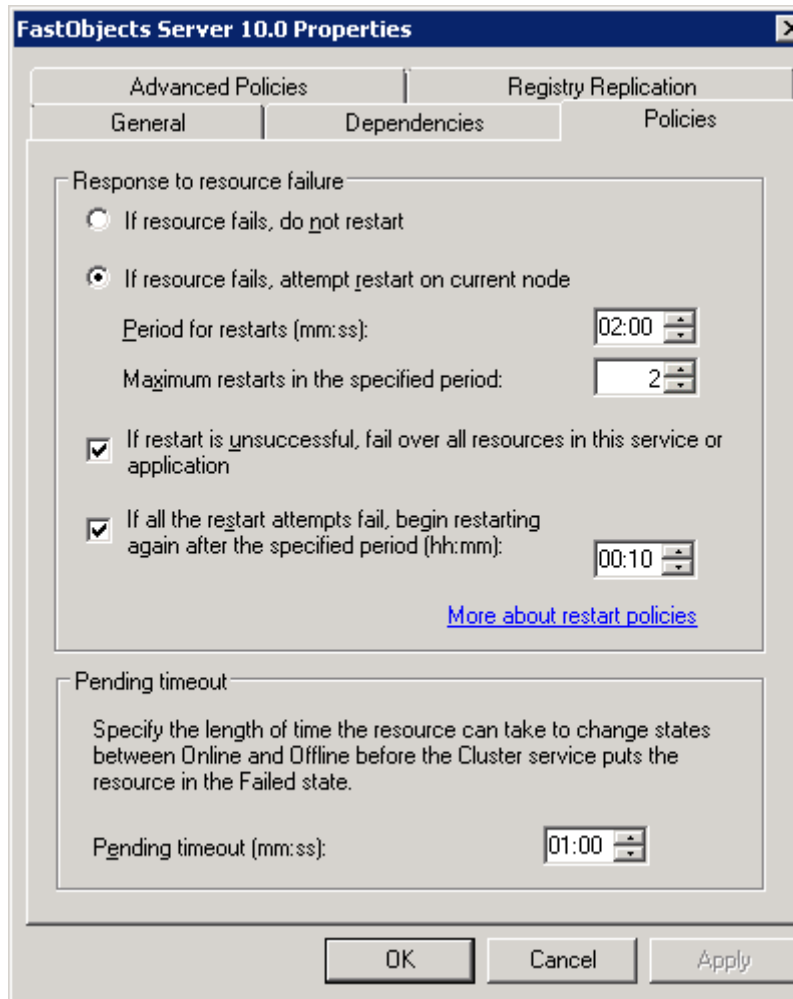


6. Set up dependencies for the service.

Recommended: at least one cluster disk, and the cluster name/IP. See the figure below for an example configuration.



- Set up policies for the service to control behavior.
See the figure below for an example configuration.



8. If using incremental backup, the ibackup needs to run towards a clustered drive and must follow any fail-over along with the service and primary disk.

7.3.2. 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.

7.3.3. 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.

7.3.3.1. 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.

7.3.3.2. 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.

7.3.3.3. 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.

7.3.3.4. System Databases and Survey 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 servers can only host Confirmit survey databases.

7.3.3.5. The [confirm_admin] Database

This database controls the status of all survey databases attached to a specific instance of SQL Server and must be present on all SQL Server instances. When setting up additional SQL Server instances for Confirmit, this database must be created manually and set up with required tables. After installing Confirmit, scripts for this can be found in the <confirmprog>\setup\sql\database\confirm_admin.

To set up the database, run the following steps:

1. Create the database and files, either in the SQL Server Enterprise Manager or from the SQL Server Query Analyzer using the following syntax (NB: should be done with the 'ConfirmitDeploy' login(see chapter 5.4.2.5):

```
create database [confirm_admin]
on (name='confirm_admin_dat',filename='<sqldatadir>\confirm_admin.mdf')
log
on (name='confirm_admin_log',filename='<sqllogdir>\confirm_admin.ldf')
```

2. If the step above was performed with a different user than 'ConfirmitDeploy', also run:

```
use [confirm_admin]
go
sp_changedbowner 'ConfirmitDeploy'
go
```

3. Locate and run the following scripts in the SQL Server Management Studio from the <confirmprog>\setup\sql\database\confirm_admin\ on one of the Confirmit application servers:
 - o Schema\create.sql

Note: This script is created dynamically on the server during setup and uses text substitution during runtime on the primary server, so it will require some editing before running. Specifically, all SQL Server login names will be generic and have two preceding and trailing underscores, i.e. '__confirmitauthor__'. Open the file in a text editor and run a search-replace throughout the file for the three SQL Server logins specified in the first part of the file.

- o Schema\content.sql
- o Schemaupdate\verN\stepN.sql
- o Schemaupdate\verN\stepN+1.sql
- o ...
- o Schemaupdate\verN+1\stepN.sql
- o Schemaupdate\verN+1\stepN+1.sql
- o ...

4. Run all scripts in the **schemaupdate** folders to bring the database up to the current version.

7.3.3.6. 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.

7.3.3.7. Making the New Database Server Visible in Confirmit

Log on to the primary database server as a user with system administrate permission using SQL Server Query Analyzer and run the following query (substituting 'ConfirmitDB2' with the chosen alias or IP address for the new server:

```
use [confirm]
go
insert into cfgServer (ServerName)
values ('ConfirmitDB2')
```

This statement will add the server to the system configuration table and concludes the SQL Server part of the setup. The final step is setting up the server properties in the Confirmit system configuration; assigning the SQLServer role, setting up target directories for databases, assigning a Server ID (incrementally higher than the existing IDs for existing servers) and setting a database weight. Consult the post-installation chapter for details.

7.4. 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 the Multiple SQL Server Instances section on page 72 for more information)) you can select which instance(s) you want Database Encryption activated on.

Note: Database Encryption requires SQL Server Enterprise Edition.

7.4.1. 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:

7.4.1.1. 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):

```
USEmaster;
GO
CREATEMASTERKEYENCRYPTIONBYPASSWORD='EncryptionPassword';
go
CREATECERTIFICATE 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.

7.4.1.2. 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.

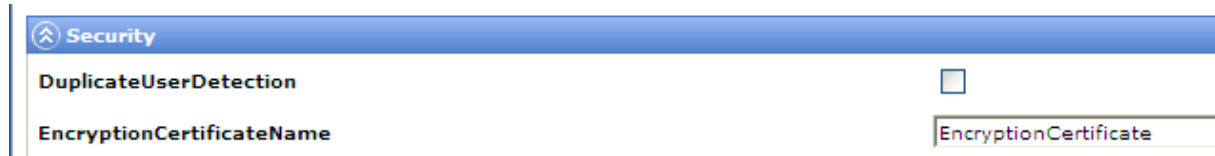


Figure 39 Entering the certificate name

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.

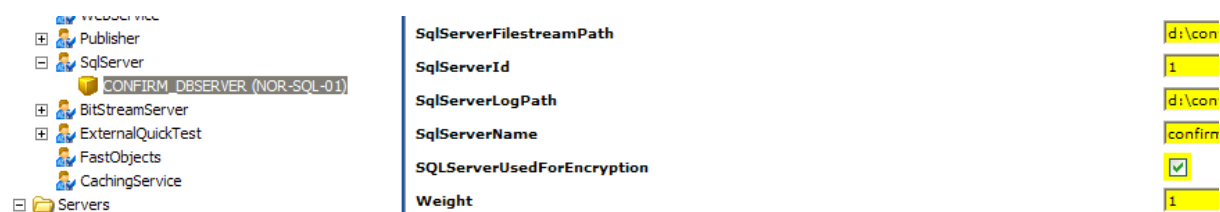


Figure 40 Checking the SQLServerUsedForEncryption checkbox

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 41 Adding the Database Encryption add-on

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

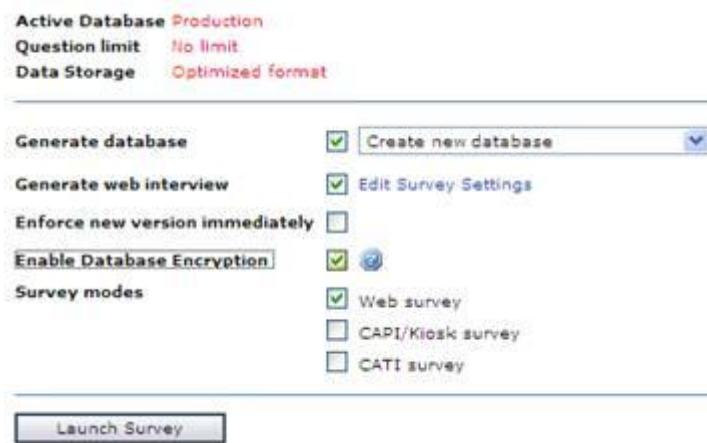


Figure 42 Selecting Enable Database Encryption

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.

7.5. Confirmit and UAC (User Account Control)

7.5.1. 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.

7.5.1.1. On Windows 2008

To disable UAC:

```
C:\Windows\System32\cmd.exe /k %windir%\System32\reg.exe ADD
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System /v
EnableLUA /t REG_DWORD /d 0 /f
```

Or run msconfig.exe, select **Disable UAC** and click **Launch**.

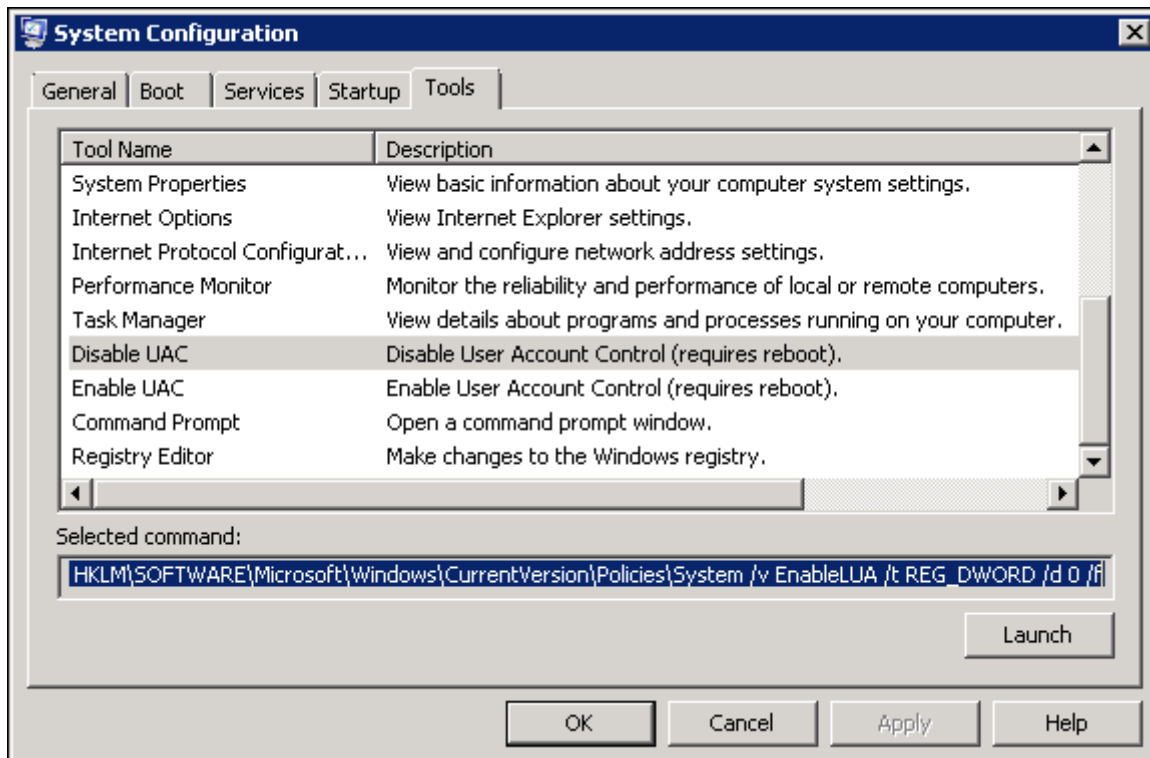


Figure 43 Disabling UAC

To enable UAC:

```
C:\Windows\System32\cmd.exe /k %windir%\System32\reg.exe ADD
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System /v
EnableLUA /t REG_DWORD /d 1 /f
```

Or run msconfig.exe, select **Enable UAC** and click **Launch**.

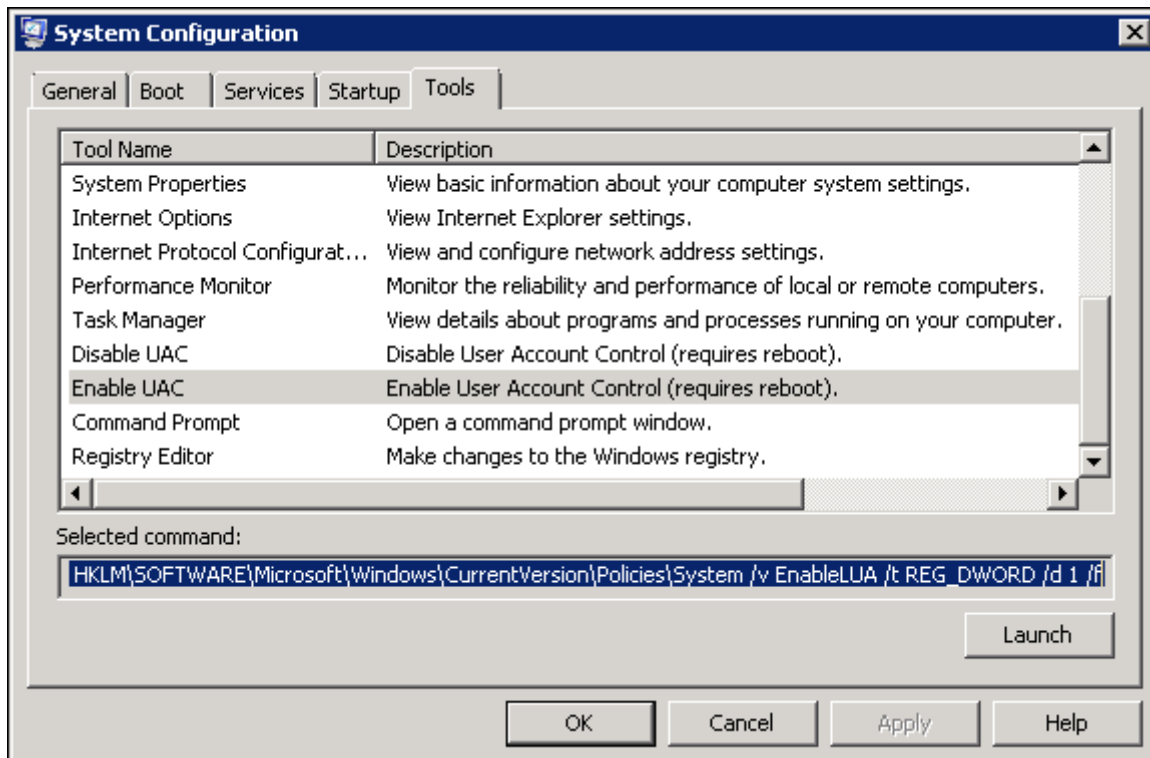


Figure 44 Enabling UAC

7.5.1.2. On Windows 2008 R2

To disable UAC:

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

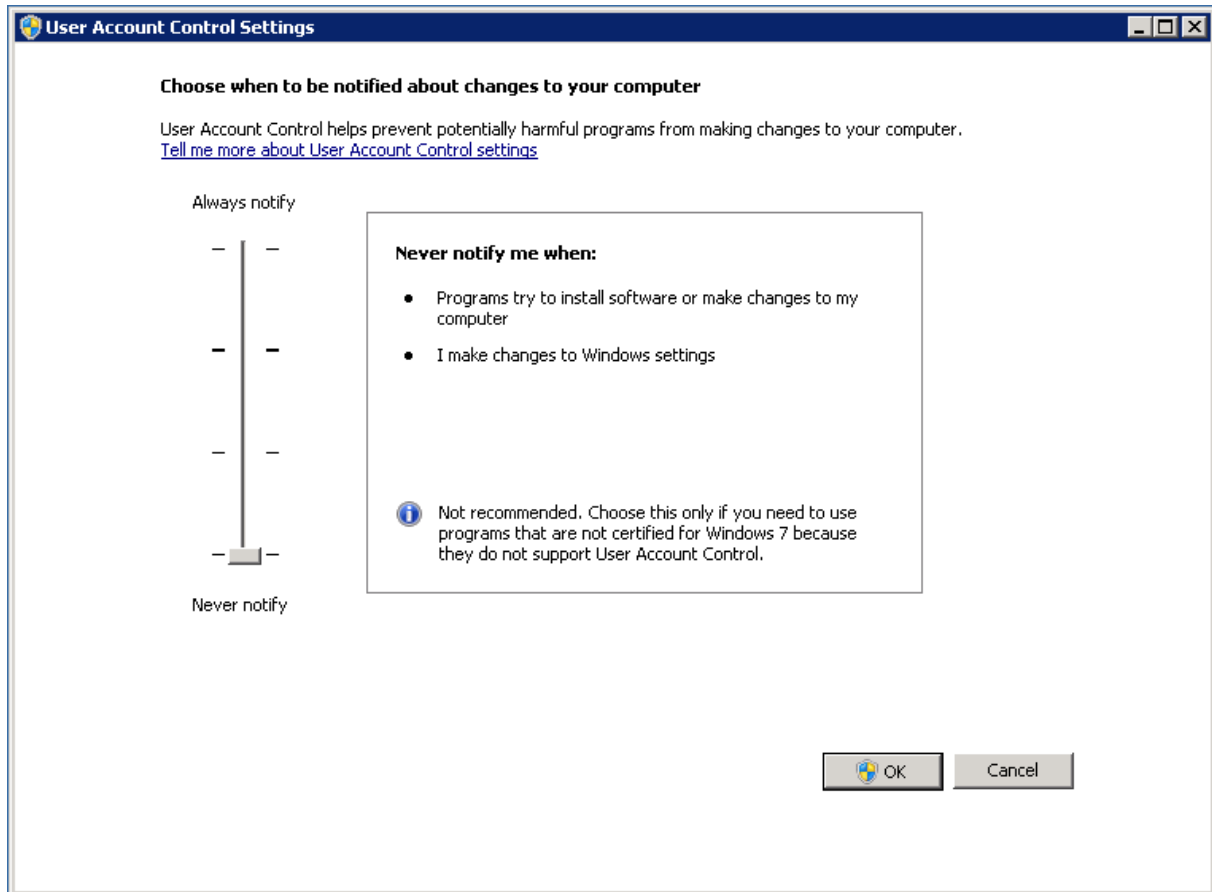


Figure 45 Disabling UAC

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).

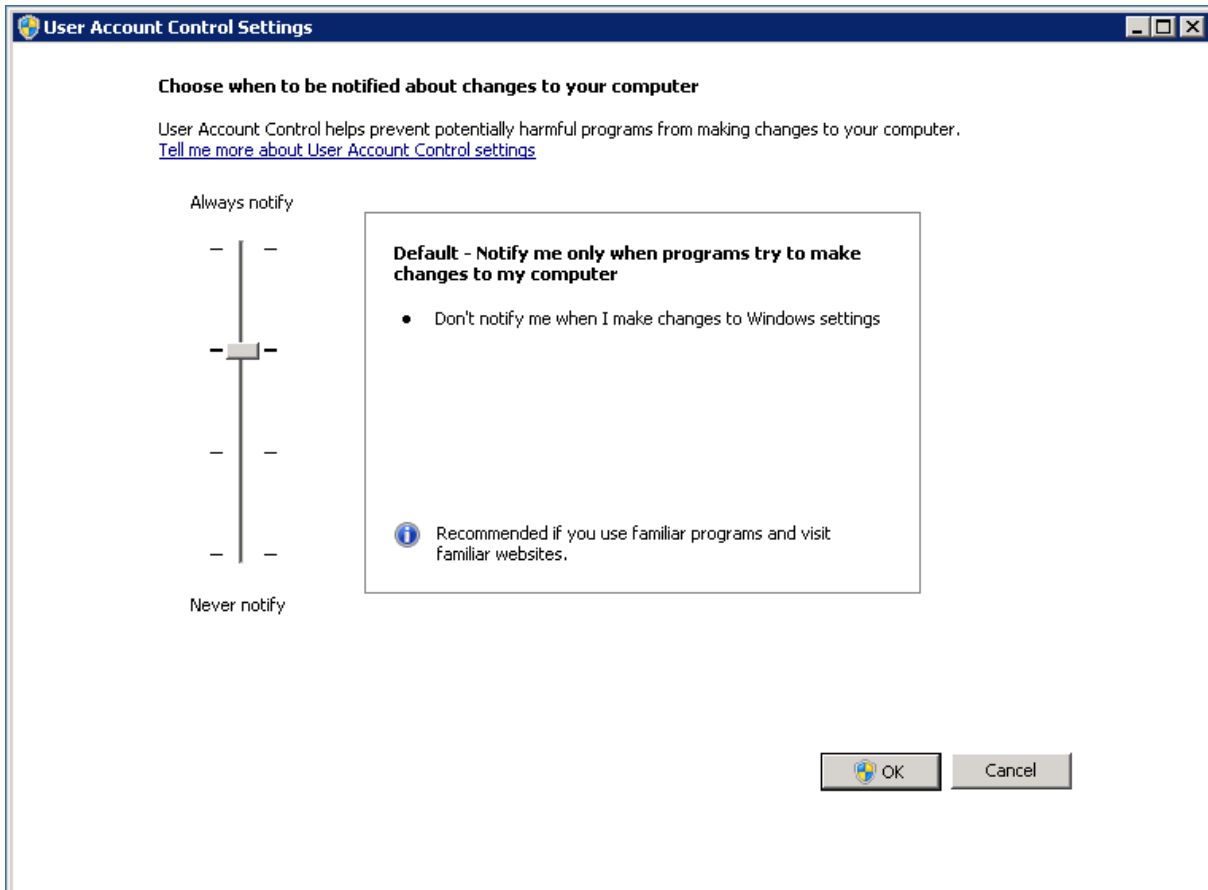


Figure 46 Enabling UAC

Note: Enabling and disabling UAC requires a reboot.

7.5.2. 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.

8. 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.

8.1. The Confirmit Monitoring Utility

The Confirmit On-Demand Operations team has built a companion utility for the built-in monitors that extracts data to a database on an aggregate level, presents the results in a nice 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 On-Demand Operations team.

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

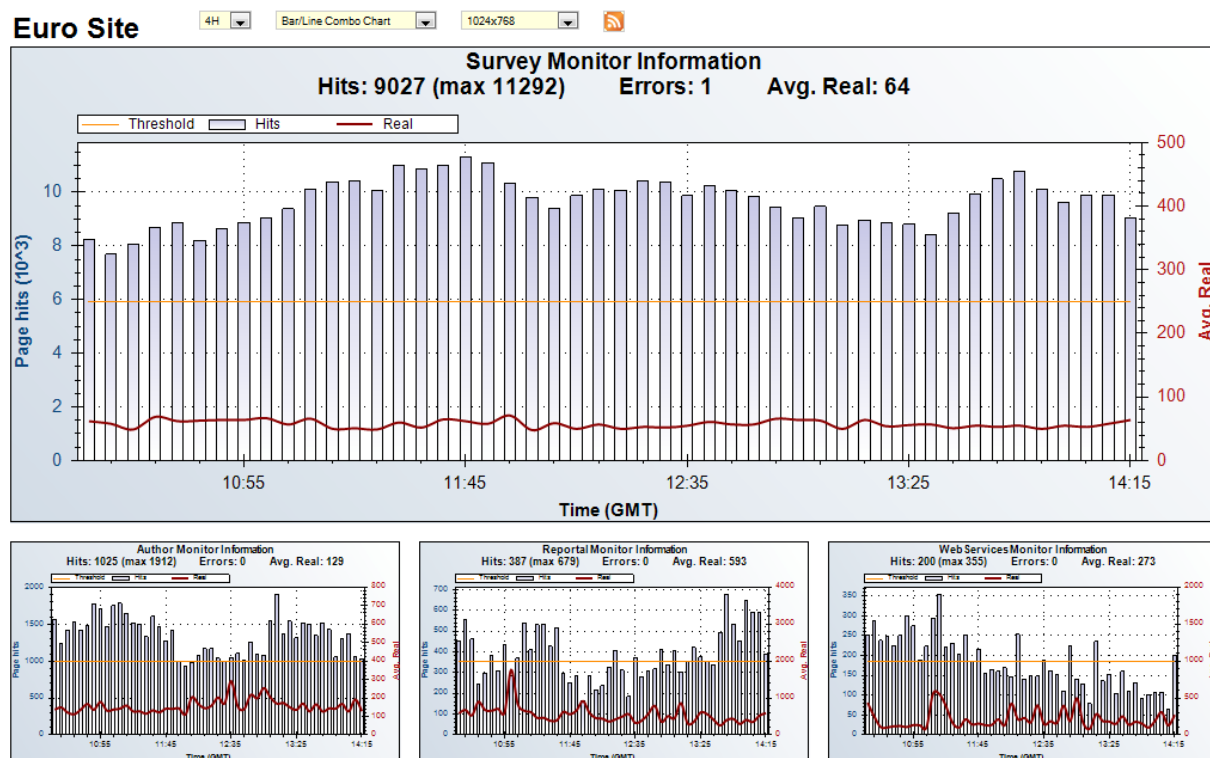


Figure 47 Example of the monitoring utility screen layout

8.2. 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.

8.3. 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 and Web Services.

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

Confirmit Authoring only supports Internet Explorer, but Confirmit Express may be used for monitoring purposes using a different browser engine.

8.4. Confirmit-Specific Monitoring and Recommended Sensors

Monitoring of certain Confirmit components should be set up for any production installation. 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 linked below follows:

8.4.1. Confirmit System Service Monitoring

Confirmit installs various system services when installed with certain roles selected. 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 will 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.

8.4.2. 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.

8.4.3. 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.

8.4.4. 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())
```

8.4.5. 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())
```

8.4.6. 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;
```

8.4.7. 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());
```

8.4.8. Monitoring Memory Usage for the Confirmit Task System Service

The Confirmit Task System service includes calls to third party code that runs in unmanaged memory, which may end up using up all available memory (2GB total including private and virtual bytes) after running for a prolonged time. We have written a script that can be set up to run as a scheduled task on any Confirmit server within a given site, that will monitor the memory usage of the Task System Service on all registered servers running the Confirmit Task System service, and restart the service if the memory has reached a certain threshold. The script can be obtained by contacting support@confirmit.com. Some parameters are configurable within the script itself. Note: The script must be run using the 32-bit version of Windows Script Host, or it will fail creating remote server connections. The script requires access to WMI on remote servers, access to the main Confirmit SQL Server, and must be run under a user context with permissions to control system services remotely.

Note: As this script 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 On-Demand Operations team.

8.4.9. 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 if possible. 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. 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;
```

8.5. 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 redundant, all hosts should have separate monitoring. Setting up an "isalive" page which monitoring systems can poll to verify IIS connectivity is highly recommended.







INTERNAL MONITORING		EXTERNAL MONITORING	APPLICATION MONITORING	
RESOURCE MONITORING <ul style="list-style-type: none"> Diskspace CPU (100%) 	APPLICATION MONITORING <ul style="list-style-type: none"> IIS (Isalive.aspx) Author Login (Localhost) SMTP Service (Caching Service) 	AUTHOR 	<ul style="list-style-type: none"> Author.company.com (isalive.aspx) Author.company.com (Forms Login) Latency 	
<ul style="list-style-type: none"> Diskspace CPU (100%) 	<ul style="list-style-type: none"> IIS (Isalive.aspx) Reportal Login (Localhost) SMTP Service 	REPORTAL 	<ul style="list-style-type: none"> Reportal.company.com (isalive.aspx) Reportal.company.com (Forms Login) Latency 	
<ul style="list-style-type: none"> Diskspace CPU (100%) 	<ul style="list-style-type: none"> IIS (Isalive.aspx) WIX AppPools (local IP) Monitoring Survey (Limited) SMTP Service 	DEPLOYMENT 	<ul style="list-style-type: none"> Survey.company.com (isalive.aspx) Survey.company.com (Survey - WIX AppPools) Latency 	
<ul style="list-style-type: none"> Diskspace CPU (100%) 	<ul style="list-style-type: none"> Task System Service Memory Usage Task Queue (SQL) SMTP Service SMTP Queue (BADMAIL\ Pickup) 	BATCH 	<ul style="list-style-type: none"> SFTP\FTP Service (If applicable) SMTP Service (Port) 	
<ul style="list-style-type: none"> Diskspace CPU (100%) I/O Service Thread count 	<ul style="list-style-type: none"> FastObjects service 	FO 	<ul style="list-style-type: none"> N/A 	
<ul style="list-style-type: none"> Diskspace CPU (100%) Custom SQL Perflogs I/O 	<ul style="list-style-type: none"> Task Queue (Project\Type) Bugreports (Monitor Tables) Performance Info (Latency) 	SQL 	<ul style="list-style-type: none"> N/A 	

Figure 48 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.

Index

.NET applications, 2

A

- Account, 31
- Activation Info, 44
- Additional Components, 19
 - Per Server Role, 20
- Additions and Patches, 19
- Add-ons, 4, 42
- Advanced System Configuration, 60
- Anti-Virus Recommendation, 57
- Application
 - Data Storage, 4
 - Dataflow, 5
 - Pool Recommendations, 63
- Architecture, 2
- Archiving Task, 50
- ASP
 - Response Size Limit, 60
 - Scripts, 2
- aspnet_setreg.exe Tool, 49
- Assigning
 - New Task Types, 48
 - Tasks to a Task System Server, 45
- Attached Survey Databases, 84
- Authoring, 2
- Authoring and Express Features, 23
- Autoclose, 28
- Autogrow, 27
- Automatic Detachment of Inactive Survey Databases, 56
- Autoshrink, 27
- Availability, 9

B

- Backend, 10
- Backup, 30, 53
 - and Recovery, 52
 - Confirmit System Databases, 30
 - FastObjects Database, 30
 - Incremental, 54
 - Offline, 53
 - Online, 54
- Basics, 2
- Before You Start, 9
- BitStream, 4, 8, 47
 - Performance Counters, 67
 - Server Operation Modes, 46
 - Service, 82
- browscap.ini
 - Replacing, 42
- Building a Confirmit Site, 9

C

- Caching Service, 82
- Changing the Location of PGP Folders, 29
- Clearing Contents of Log Tables, 57

- Cluster Nodes, 69
- Collation, 73
- Command Line Management Tools, 49
- Community Panels, 26
- Compression, 64
- Concepts, 53
- Configuration, 48, 60
 - System Concepts, 48
- Configuration Tool, 35
- Configure Database Encryption, 74
- Configuring
 - IIS, 60
 - the Model Database, 27
- Confirmit
 - add-ons, 4
 - Architecture, 2
 - Authoring, 2
 - Authoring Dataflow, 5
 - BitStream, 46
 - BitStream Dataflow, 8
 - BitStream Service, 4
 - BitStream Services, 31
 - Configuration, 48
 - Configuration Tool, 35
 - Data Export Dataflow, 7
 - Data Share, 45
 - File Library, 31
 - Setting up, 47
 - Flex Extension, 4
 - Installation, 32
 - Licensing, 29
 - Maintenance, 52
 - Monitoring Utility, 81
 - Multimode, 4
 - Reportal, 3
 - Reportal Dataflow, 7
 - SDK License, 45
 - Survey Dataflow, 6
 - System Service Monitoring, 82
 - Task System, 3, 31
 - Task System Helper Service, 31
 - Task System Service, 83
 - Translator, 4
 - Upgrades, 55
 - Web Services, 4
 - XML Web Services Development, 26
- ConfirmitAdmin Tool, 50
- ConfirmitAdmin.exe, 50
- ConfirmitAdministrator Tool, 49
- Creating
 - Aliases, 74
 - Data Folders
 - SQL, 30

D

- DAS, 72
- Data
 - Export, 7
 - Storage, 4
- Data Files Generated by Confirmit, 52
- Database, 73
 - Cleanup Task, 51

- Encryption, 74
 - Confirmit, 75
 - SQL, 74
- Server Version Support, 26
- Servers, 10
- Shrinking, 57
- System Redundancy, 69
- Databases, 73
- Dataflow, 5, 6, 7, 8
- Defragmenting Indexes, 57
- Deleting Survey Test Databases, 57
- Deploying Language and Resource Files, 46, 48
- Deployment, 3
- Disable FastObjects Incremental Backup, 30
- disable UAC, 76, 78
- Dundas
 - Web Chart/Gauge Components, 23, 25
 - Web Chart/Gauge Components in Reportal, 26

E

- Enable
 - FILESTREAM on SQL, 28
 - UAC, 77, 79
- Encryption, 74
- Enforce Compatibility Mode, 60
- Enterprise Instrumentation Framework, 19
- Entry-Level Multi-Server Setup, 13
- Errors
 - from Authoring, 82
 - from Reportal, 83
 - from Web Services, 83
- Event Logs, 57
- Example Scenario, 13
- Existing Settings, 30
- Express Features, 23
- Extension, 4
- External Location Monitoring, 82

F

- Failing Launch Survey Tasks, 83
- FastObjects, 3, 4, 13, 30, 55, 69
 - Database, 30, 53
 - Database Server, 11
 - Incremental Backup, 30
 - Server, 26
- Features, 2
 - and Modules, 2
- File Library
 - Files, 31
 - Setting up, 47
- File System Permissions, 45
- FILESTREAM
 - Enable on SQL, 28
- Finding the Site ID, 43
- Firewalls, 11
- First Cluster Server, 69
- Flex Extension, 4
- FTP, 3
- Full Offline Backup, 53
- Full Online Backup, 54

G

- Generic Server Monitoring, 81

H

- Hardware Configuration, 9
- High-End Multi-Server Setup, 15

I

- ibackup, 72
- IDS, 11
- IE8 Browsers, 60
- IIS, 60
 - Advanced Configuration, 60
 - Compression Tuning, 64
 - Configuration, 58
 - Hardening, 58
 - Log File Maintenance, 58
 - Log Files, 58
 - Monitoring, 58
- IIS7, 58, 60
- Inactive Survey Databases
 - Detachment, 56
- Incremental Backup, 30, 54, 55
- Indexes
 - Defragmenting, 57
- Initiating Special Task Types, 50
- Install Sequence, 32
- Installation, 31, 32
 - Preparation, 18
- Installing
 - Confirmit, 29, 76
 - the License File, 44

K

- Keeping Existing Settings, 30
- Key Features, 18

L

- Language and Resource Files, 46, 48
- Layout, 12
- License
 - File
 - Installation, 44
 - Obtaining and Installing, 42
 - SDK, 45
- Licensing, 27, 29
 - Concepts, 42
- Linked Servers, 73
- Load Balancer, 11, 68
- Log Tables
 - Clearing Contents, 57
- Login, 73

M

- Maintenance, 52
- Microsoft
 - .NET Framework, 2
 - .NET Framework 1.1, 19
 - .NET Framework 2.0, 19
 - .NET framework 3.5 SP1, 19
 - Internet Information Server, 2
 - Office 2007, 25
 - Report Viewer Distributable 2008 SP1, 20
 - SQL Server, 2, 4

- 2008, 24
- 2008 CLR Types, 20
- 2008 Datamining Viewer Controls, 24
- 2008 Editions, 27
- 2008 Native client, 20
- 2008 Shared Management Objects, 20
- Visual C++ 2008 Redistributable Package, 20
- Visual J# 2.0 for .NET, 20
- Mid-Level, 14
- Model Database
 - Configuring, 27
- Modules, 2
- Monitoring
 - and Recommended Sensors, 82
 - and Systems Maintenance, 55
 - Confirmit, 81
 - Delays in Task Queue Execution, 83
 - Errors from Authoring, 82
 - Errors from Reportal, 83
 - Errors from Web Services, 83
 - Example, 84
 - External Monitoring, 82
 - Failing Launch Survey Tasks, 83
 - Generic Server, 81
 - Memory Usage, 83
 - Number of Attached Survey Databases, 84
 - Survey Engine Availability
 - Survey, 59
 - Surveys Returning Errors, 82
 - Utility, 81
 - Web Server Availability
 - Basic Page, 58
- Moving the Web Root, 58
- Multi Server
 - With Sync, 47
- Multimode, 4
 - Server, 11
- Multiple SQL Server Instances, 72
- Multi-Server, 14, 15
 - Setup, 13

N

- Network
 - and Hardware Configuration, 9
 - Equipment, 11
 - Load Balancing, 11
 - Traffic Requirements, 11, 12
- NLB, 11
- NTFS compression, 58

O

- Obtaining and Installing a License File, 42
- Offline Backup, 53
- OLAP, 26
- Online Backup, 54
- Open XML SDK 2.0, 20
- Operating System, 18
 - Additions and Patches, 19
- Optional Components, 4
- OutOfMemory, 61

P

- Panelist Reporting, 26

- Partial Restore, 55
- Performance, 9
 - Counters, 66, 67
- PGP, 3
 - Folders
 - Changing Location, 29
 - key rings, 29
- Pool
 - Performance Tuning, 60
 - Recommendations, 63
- Post-Installation Server Configuration, 41
- Pre-Installation, 30
- Preparing an Installation, 18
- Preventing Search Engine Indexing, 66
- Project Info Collector Task, 50
- ptadmin.exe tool, 53
- Publisher, 3

R

- Recommendations, 12
 - Anti-virus, 57
- Recommended Sensors, 82
- Recovery, 52
 - model, 28
- Redundancy, 69
- Replacing browscap.ini, 42
- Reportal, 7
- Required Update, 27
- Resource Files, 46, 48
- Response Data Storage, 4
- Restore, 55
 - Partial, 55
 - with Incremental Backups, 55
- Restoring Survey Databases, 55
- Role-Wide Settings, 49

S

- SAN, 72
- Scalability, 2, 9
- SDK
 - License, 45
 - Uninstall, 30
- Search Engine Indexing, 66
- Second Cluster Server, 69
- Secure Event Log Viewing, 58
- Server
 - Configuration
 - Post Installation, 41
 - Hardware, 9
 - Logins, 73
 - Maintenance
 - SQL Database, 55
- Servers
 - Database, 10
 - Task System, 10
 - Web, 9
- Server-Wide Settings, 49
- Service Packs and Updates, 19
- Services MMC, 30
- Setting Up
 - 'the Confirmit File Library, 47
 - User Accounts, 31
- Settings
 - Role-wide, 49

- Server-wide, 49
- Site-wide, 48
- Setup, 13, 14, 15
- Shrinking Databases, 57
- Single Server With Sync, 47
- Single Server, No Sync, 47
- Site
 - ID
 - Finding, 43
 - Layout, 12
- Site-Wide Settings, 48
- SQL
 - Database Server Maintenance, 55
 - Query, 82, 83, 84
- SQL Server
 - 2008 Native Client, 24
 - Client Tools, 24
 - Collation, 73
 - Components Version Support, 24
 - Databases, 52
 - in Windows Cluster, 72
 - Integration Services, 24
 - LOB Compaction, 57
 - Logins, 73
 - Reporting Services, 24
- SSIS, 24
- SSL Acceleration, 68
- SSLAcceleratorMode, 68
- SSLAcceleratorPort, 68
- SSLAuthorCertificate, 68
- SSRS, 24
- Stop the Confirmit Task System Service, 30
- Subsystems, 4
- Supported Operating Systems, 18
- Survey, 6
 - Databases, 73
 - Restoring, 55
 - Engine, 3
 - Test Databases
 - Deleting, 57
- SurveyPackageSlidingExpiration, 63
- Surveys Returning Errors, 82
- Switches, 11
- Synchronization, 57
- System
 - Concepts, 48
 - Configuration, 60
 - Databases, 73
 - Files, 52
 - Maintenance, 55
 - Partitions, 52
 - Service Monitoring, 82

T

- Task Isolation Level, 46
- Task Queue Execution, 83
- Task System, 3
 - Features, 24
 - Server, 45
 - Servers, 10, 48
 - Service, 82
 - Stop, 30
- Task Types, 50
- The [confirm_admin] Database, 73
- The Archiving Task, 50
- Time Synchronization, 57
- Translator, 4
- Troubleshoot Survey Problems, 66
- Tuning, 60, 64

U

- UAC, 76
- Uninstall Confirmit SDK, 30
- Universal Object Server, 4
- Upgrade Sequence, 32
- Upgrades, 55
- Upgrading
 - From a Previous Version, 48
 - From an Existing Version, 29
- User Account Control, 76
- User Accounts
 - Setting up, 31
- Using Confirmit, 80

V

- Versant FastObjects, 2
- Virtual Infrastructure, 17

W

- w3wp.exe, 61
- Web Root
 - Moving, 58
- Web Servers, 9
- Web Services, 4, 26
 - Enhancements 2.0 SP3, 19
 - Enhancements 3.0, 19
- Windows
 - 2008, 76
 - 2008 R2, 78
 - Cluster, 72
 - Server Editions, 18
 - Server Manager, 22
- Wintertree Software Spelling Server, 23