Workshare Compare Server 9.5.4

Admin Guide

February 2019 | 9.5.4. 4774

Table of Contents

Chapter 1: Product Overview 4
What is Workshare Compare Server? 5
What to expect with PDF comparison 5
System Requirements 6
Hardware 6
Supported operating systems 6
Chapter 2: Deploying Workshare Compare Server 7
Prerequisites 8
Installation 9
Verifying installation 18
Uninstalling 18
Chapter 3: Configuration 19
Configuration for RESTful APIs 20
Licensing 20
Change the target of the root URL 20
Queuing 20
DVJS comparison 22
Logging 24
Configuration for Legacy SOAP APIs 25
Licensing 25
Event Logging 25
Customizing the API 26
Chapter 4: Monitoring 27
Introducing Metrics 28
Enabling Metrics 28
Metrics Published 29
Format of metrics 29
Metric label 30
Types of metrics 30
Details of metrics 30
Chapter 5: Troubleshooting 33
General Checks 34

Installer fails 34
Swagger troubleshooting 34
PDF Comparison 34
Log file locations for RESTful APIs 34
Troubleshooting SOAP APIs 35
C# sample does not work 35
Compare Service page doesn’t display 35
Comparing large files 36
SOAP endpoint issues 36

Chapter 1: Product Overview

This chapter provides an overview of Workshare Compare Server as well as a list of system requirements. It includes the following sections:

- **What is Workshare Compare Server?**, page 5, introduces Compare Server and its key functionality.
- **System Requirements**, page 6, describes the system attributes required to run Compare Server.

What is Workshare Compare Server?

Workshare Compare Server is a web service that performs server-side document comparison and exposes APIs (Application Programming Interfaces) to allow a range of client applications to exploit this functionality. The APIs enable software developers to write custom software that compares two Microsoft Word, RTF, PDF, TXT or HTML documents and produces a redline document that describes the differences between the two documents. Compare Server is installed on a server.

Compare Server exposes simple APIs that allow for integration into a range of client applications and server-based solutions. For example, it can be used by custom solutions that run on your internet site, intranet site, internal network or your email system. Clients can be developed in Java, C#, C++, VB.NET and other languages to integrate Compare Server into your desktop, DMS or document applications.

What to expect with PDF comparison

Compare Server provides fast and accurate comparison of Word and PDF files. However, the comparison of PDF files may not be as accurate as the comparison of Word documents for the following reasons:

- In order to display a PDF comparison as a redline, Compare Server converts PDF documents back into Word documents, then applies traditional document comparison (Word to Word) to the resulting documents. Converting a PDF into a Word document is an imprecise process because there is not enough information available (in the PDF file) to pin down a unique Word document that is the one and only possible output for that particular PDF file.
- Scanned PDFs add another layer of complexity because in order to compare the text, Compare Server must first perform OCR on the file. The quality of the scan can have a big impact on the comparison results. The OCR engine used can also impact the resulting output and therefore accuracy of the comparison. No OCR engine will ever be 100% perfect with every scanned PDF – particularly low quality scans.

System Requirements

Workshare Compare Server is designed to run on both entry level and enterprise scale servers. Recommended specifications are given below.

Hardware

CPU	64-bit architecture-based computer with Intel or AMD processor with 4 cores
Memory	4GB RAM – more memory may be needed if you intend multiple users to regularly compare long and complex documents
Storage	1GB free disk space for installation; additional space required for storage of DVJS comparisons and queued comparisons

Supported operating systems

Microsoft Windows Server 2016 x64 Edition

***Note**: Compare Server can be installed on Microsoft Windows 10 for demonstration and evaluation purposes only.*

It is strongly recommended that you install Compare Server on a dedicated server instance.

Chapter 2: Deploying Workshare Compare Server

This chapter describes the installation procedure for Workshare Compare Server. It includes the following sections:

- **Prerequisites**, page 8, describes the software required before installation of Compare Server.
- **Installation**, page 9, describes the step-by-step procedure required to install Compare Server.
- **Uninstalling**, page 18, provides important information about uninstalling Compare Server.

Prerequisites

Workshare Compare Server is installed using a single installation file. It will automatically install the following required software (if not found) as part of the installation process.

- Microsoft .NET Framework 4.6.2
- Microsoft Visual C++ 2015 Redistributable Package (x64 and x86)
- Microsoft IIS (Internet Information Services)
- Windows Server Components: The install enables the following features:

Feature internal name	Feature display name
IIS-ASPNET45	ASP.NET 4.7
WCF-HTTP-Activation	Windows Communication Foundation HTTP Activation
WCF-Pipe-Activation45	WCF Named Pipe Activation
WCF-TCP-Activation45	WCF TCP Activation
WCF-HTTP-Activation45	WCF HTTP Activation
WAS-WindowsActivationService	Windows Process Activation Service
WAS-ConfigurationAPI	WAS Configuration APIs
WAS-ProcessModel	WAS Process Model
IIS-BasicAuthentication	IIS Basic Authentication
IIS-CustomLogging	IIS Custom Logging
IIS-HttpCompressionDynamic	IIS Dynamic Content Compression
IIS-HealthAndDiagnostics	IIS Health and Diagnostics
IIS-RequestFiltering	IIS Request Filtering
IIS-HttpTracing	IIS Tracing

***Note**: The install requires these features **and their dependencies** to be installed, so for instance IIS_ASPNET45 will cause many other IIS features to be installed as dependencies.*

Installation

Workshare Compare Server is provided as a standard installation file, with an additional company-specific file for licensing purposes:

Workshare.CompareServer Install-[build #].exe	The executable file used to install Compare Server.
Product.lic	The license file that contains the Compare Server license.

Note: *Both installation files (the EXE and the LIC file) must be on a local drive of the server machine and not on a mapped network drive or UNC path.*

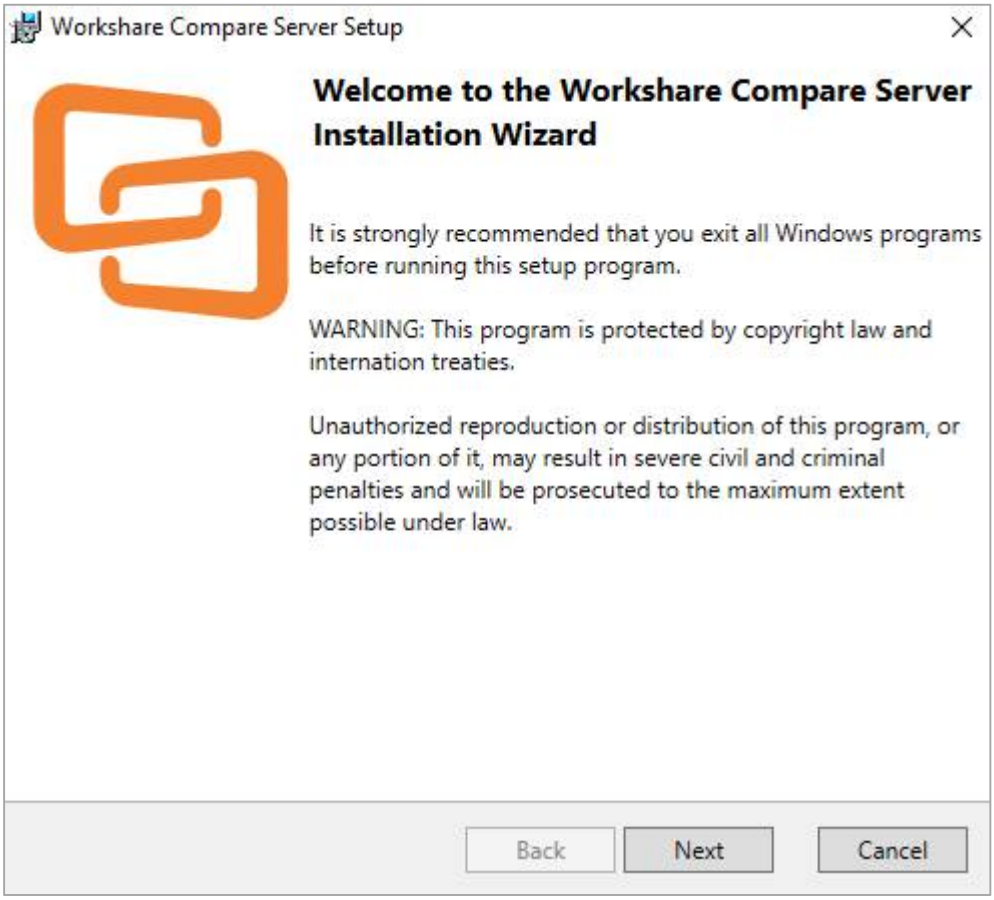
The RESTful APIs are now independent and can talk to Compare Server directly. By default, the SOAP APIs are not installed, but you can still choose to install them if you want.

During install or uninstall of Compare Server, information and error entries are written to log files. These plain text files are saved in the current user's temporary folder (for example, C:\Documents and Settings\Admin\Local Settings\Temp).

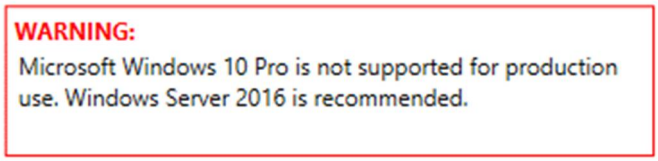
The following procedure explains how to install Compare Server on your server using the executable installer. Before beginning the installation procedure, make sure that all other programs are closed and disable any anti-virus software.

To install Compare Server:

1. Double-click the **Workshare.CompareServerInstall-9.5.XXXX.exe** file. The Workshare Compare Server Installation Wizard is displayed.

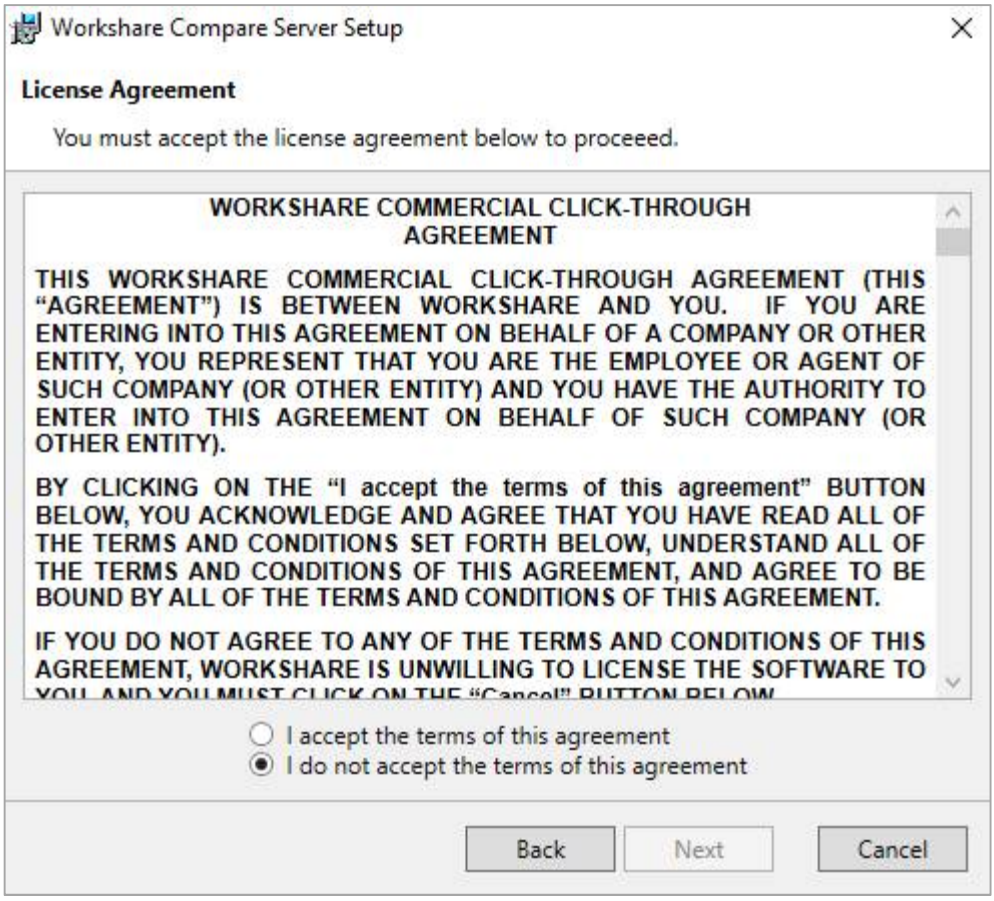


***Note**: If you are installing Compare Server on Windows 10, you will see a warning.*



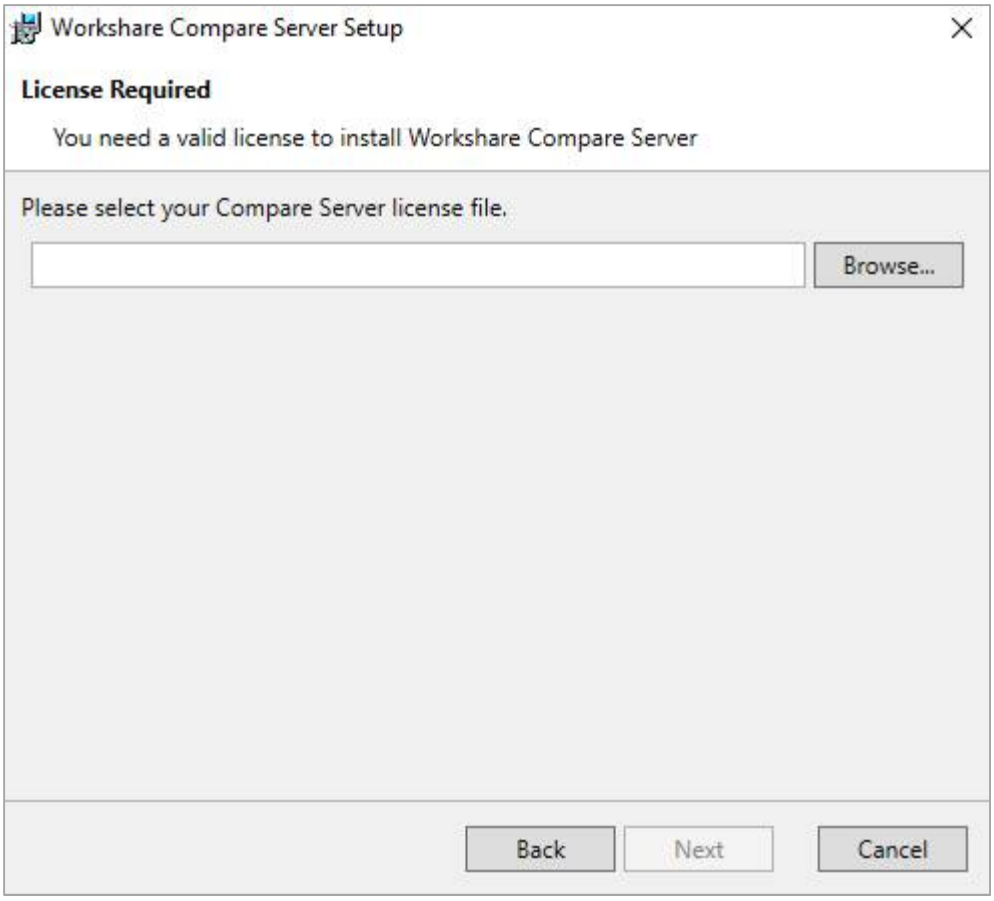
Compare Server should only be installed on Windows 10 for demonstration and evaluation purposes.

2. Click **Next**. The License Agreement screen is displayed.



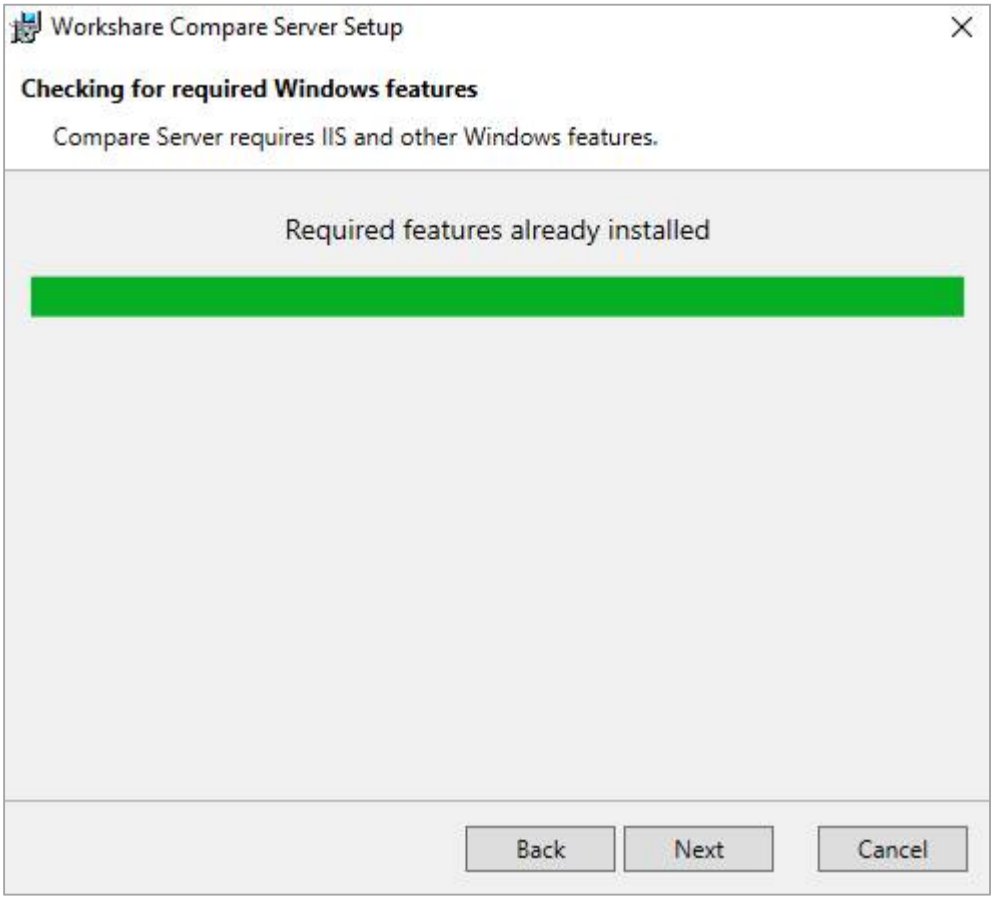
3. Read the agreement and select the **I accept the terms of this agreement** radio button.

4. Click **Next**.



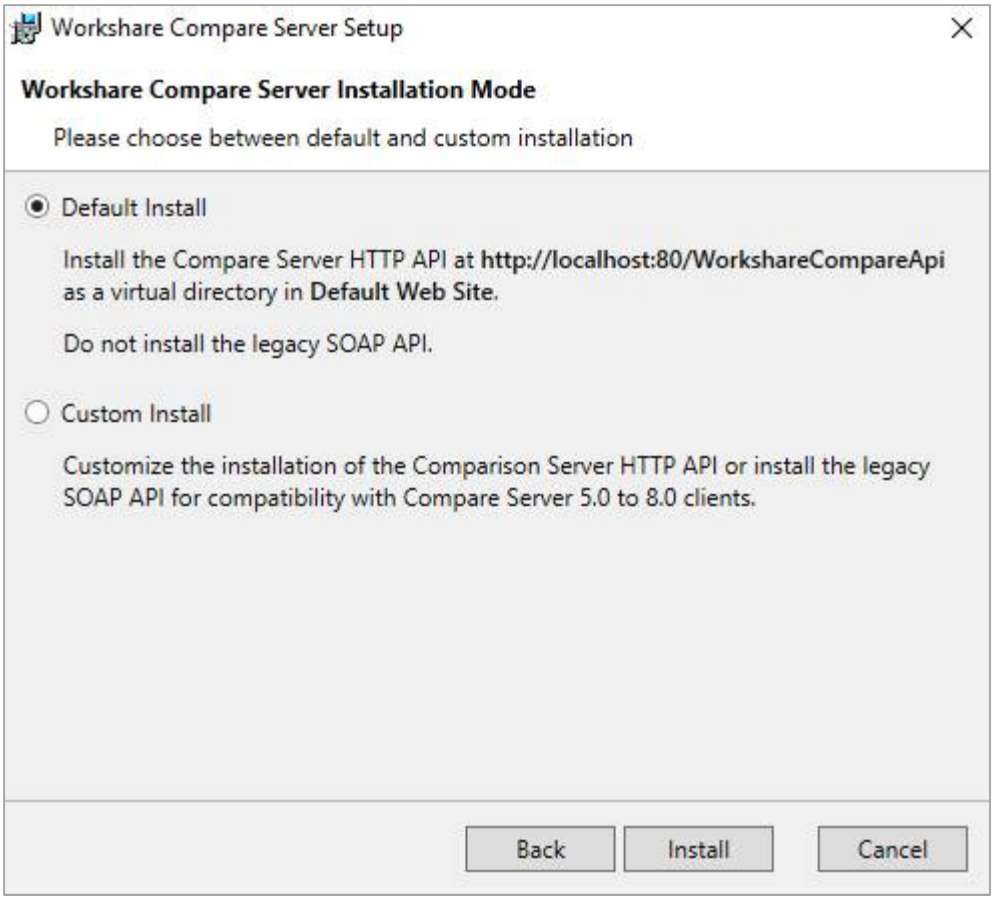
5. Workshare will automatically detect your license file if it is in the same location as the installation file. If necessary, browse to the location of your license (.lic) file. This location MUST be on the local machine.

6. Click **Next**. Workshare checks you have the required Windows components and other prerequisites. If any are not found, Workshare will install them.



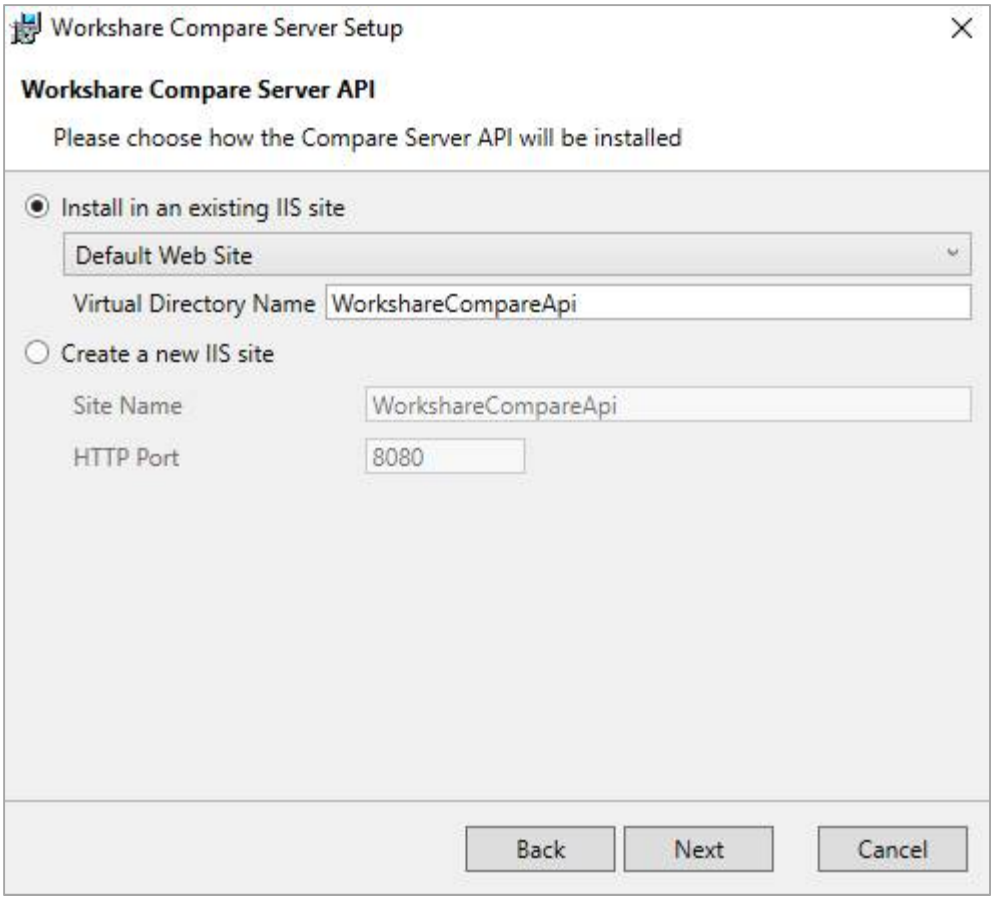
***Note**: If the installer cannot install the required components, it will display an error showing the command it tried to run. You can troubleshoot by reinserting your Windows Server installation media and rerunning the installer, or by running the command manually.*

7. Click **Next**.



8. Continue in one of the following ways:
   - Leave **Default Install** selected and click **Install**. Workshare Compare Server is installed as a web service with RESTful APIs. Go to step 12.
   - If you want to customize the installation of the RESTful APIs or you want to install the legacy SOAP APIs, select **Custom Install** and click **Install**.

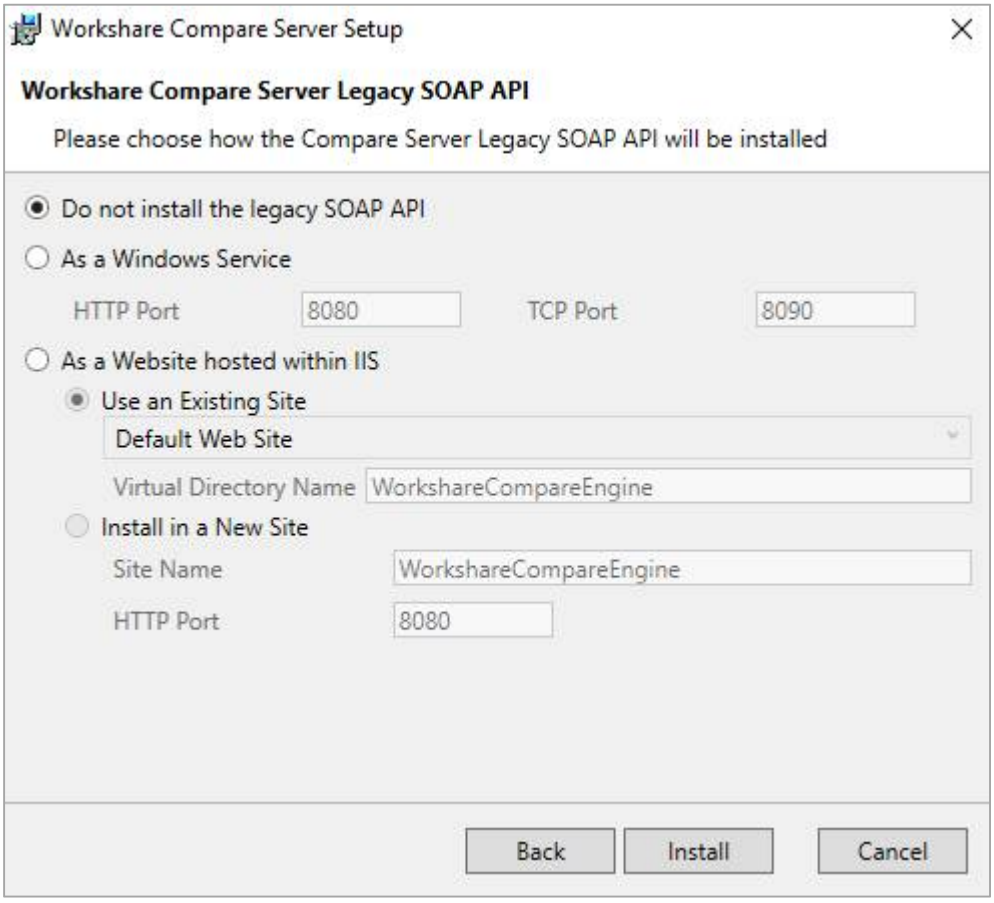
The Workshare Compare Server API screen is displayed.



9. Select an existing IIS site or create a new IIS site for the installation of the RESTful APIs.

> ***Note**: If you choose a site name that is already in use, you won't be able to proceed unless the site name is being used for a previous installation of Compare Server that you are upgrading.*

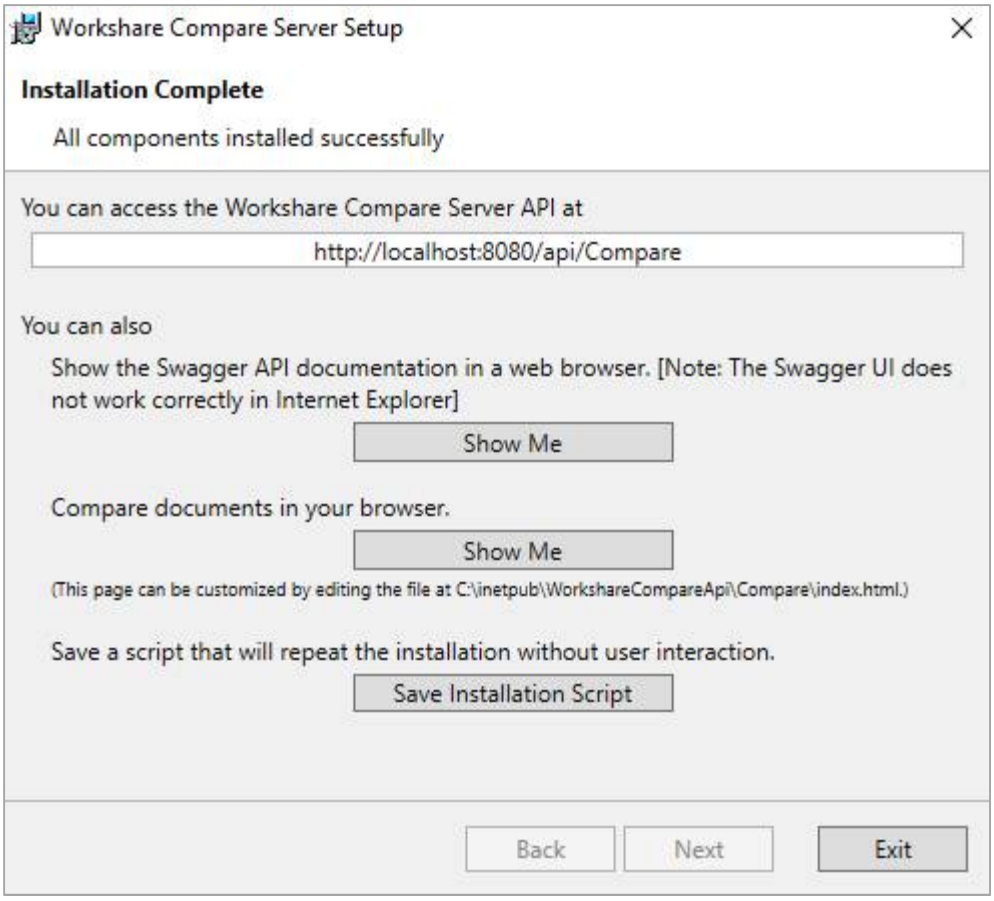
10. Click **Next**. The Workshare Compare Server Legacy SOAP API screen is displayed.



11. If you want to install the legacy SOAP APIs, select how to install them:
    - **As a Windows Service**: This provides access to the comparison engine via the HTTP, TCP and Named Pipe protocols. Specify the HTTP and TCP ports.
    - **As a Website hosted within IIS**: This provides access to the comparison engine via the HTTP and HTTPS protocols. If you want to configure the website further, use IIS Manager. Specify the website and port, or you can leave the default settings.

> ***Note**: Do not use a port that is already in use for a new website, i.e. port 80 if you have an existing website.*

12. Click **Install**. Compare Server is installed.



13. Once installation is complete, the confirmation dialog provides useful information about your install:
    - The URL for the API is provided.
    - You can display the Swagger API documentation in your browser. You can also try a test comparison from there. Refer to *Verifying installation*.

> ***Note**: If you are using the Microsoft Internet Explorer browser, the Swagger API documentation may not display correctly.*

    - You can run a comparison in your browser.
    - Click **Save Installation Script** if you want to save your installation settings for further Compare Server installs.

14. Click **Exit**.

Verifying installation

To verify your installation, run a test comparison as follows:

1. Click **Show Me** in the final page of the Installation Wizard to display the Swagger API documentation.
2. Click **Expand Operations** to show the options to perform comparisons via GET and POST.
3. Expand **Compare** (to test a classic comparison) or **UICompare** (to test a DVJS comparison).
4. Click **GET** and click the **Try It Out** button.
5. Scroll down and click **Execute**.

   If the configuration is correct and Compare Server is running, you should see the response to the request appear after a few seconds (look for a 200 code to indicate that the comparison completed successfully).

Additionally, once installation is complete, you can verify the installation as follows:

- When installed as a website hosted within IIS, go to IIS Manager, expand the Sites node and verify the Compare Server web service is there. By default, it will appear as a virtual directory called **WorkshareCompareApi** in the Default Web Site. The files will be located at c:\inetpub\WorkshareCompareApi.

***Note**: Compare Server continues to include SOAP and WCF-based APIs for compatibility with earlier versions. These legacy APIs are not recommended for new projects. Samples showing the use of these legacy APIs can be found at: https://github.com/workshare/compare-service.samples.*

Uninstalling

To upgrade from previous versions of Workshare Compare Server, you will need to uninstall the previous version. The installer of Compare Server 9.5.4 will detect earlier versions of Compare Server and stop the installation. It's necessary to manually uninstall previous versions of Compare Server before installing Compare Server 9.5.4.

You can uninstall previous versions from Add/Remove Programs or by rerunning the original installer.

When uninstalling Compare Server, some files, such as log files and configuration files, may not be removed from the machine even after completely uninstalling the product. If you want these files removed, you must do so manually.

Chapter 3: Configuration

This chapter outlines the various configuration changes that can be applied to Workshare Compare Server. Compare Server is a highly customizable tool and it is not the purpose of this chapter to describe the full potential of the APIs. However, this chapter does describe how to configure the Workshare-specific functionality. It includes the following sections:

- **Configuration for RESTful APIs**, page 20, describes how to configure the RESTful APIs.
- **Configuration for Legacy SOAP APIs**, page 25, describes how to configure the legacy SOAP APIs.

Configuration for RESTful APIs

The Compare Server RESTful APIs are configured via a configuration file located by default at **c:\inetpub\WorkshareCompareApi\web.config**.

After making changes to the web.config file, you must restart the Compare Server website in IIS.

Licensing

By default, the license file is saved in the c:\inetpub\WorkshareCompareApi\bin folder.

The license file must be specified in the **LicenseFileName** setting in the web.config file. If you move or rename the license file then the web.config setting must be updated accordingly.

When updating a license, if you replace the old license file with the new license file, there is no need to update the web.config file.

Change the target of the root URL

By default, the root page of the server displays the API documentation. Depending on the intended use of Compare Server, you may want to adjust the configuration so that the root page of the server redirects to the comparison document selection page.

To change the target of the root URL, change the **HomeUrl** setting in the web.config file from **swagger** to **Compare**.

Queuing

Comparisons can be run in an asynchronous queued manner on Compare Server. This approach is useful if you perform large numbers of comparisons and it is acceptable if the results are not returned immediately. For example, where comparisons are being performed by a background task in another system.

Set up queuing

Use of the queuing system in a production environment requires an SQL server database to be configured to store information on queued jobs. The connection string to contact the database must be stored in the **QueueDB** connection string in the web.config file and the **QueueStorage** setting must be changed from **memory** to **sql**. If these changes are not made, the comparison queue will be stored in an in-memory database and will be lost if the server, IIS or the application pool running the compare service is restarted.

Status information on the queuing system can be found by browsing to Compare Server's health check API endpoint – http://server.name/path_to_root/api/healthcheck .

Further details about the queue can be found at the queue status endpoint (http://localhost/path_to_root/queue). This endpoint can only be accessed from the server itself.

Queuing process

The queuing mode of Compare Server works as follows:

- A client sends a request to start a comparison.
- The server responds immediately with a URL which the client can use to request the status of the comparison.
- The requested comparison is placed in a queue and runs when it reaches the head of the queue.
- The client can poll repeatedly to find the status of the comparison.
- Once the comparison is complete, the polling result will be a redirect response (HTTP response status code **302 Found**) giving the URL from which the comparison result can be downloaded.

The result of the queued comparison must be stored until the client downloads it and the source documents must be stored until the comparison reaches the top of the queue. Therefore, use of the queuing system involves both documents to be compared and comparison results being stored on the disk of the Compare Server machine. Refer to DVJS comparison for storage location details and ways to customize it.

Comparison source documents are deleted from storage when the comparison itself is completed. Comparison results are deleted when the first of the following criteria is met:

- One hour after the comparison is completed
- Fifteen minutes after the first attempt to download the result of the comparison is initiated

These retention times are fixed and cannot be configured.

***Note**: The deletion process is managed by the queuing system, so if an SQL database has not been configured to store queue entries and the service restarts then some files may not be properly deleted.*

Maximum concurrent comparisons

Depending on the workload expected of Compare Server, it may be useful to set limits on the maximum number of comparisons that the server will attempt to run concurrently. By default, the maximum number of concurrent comparisons is processor count (number of logical processors on CPU) times two.

You can set this to a lower number by entering a value for the **MaxConcurrentComparisons** setting in the web.config file.

***Note**: You can enter a number as the value or in the format "CPUCount*1.5".*

The behavior of the server if the concurrency limit is reached can also be configured by adjusting the **RejectIfTooBusy** setting. When this is set to **false**, the server will accept new comparison requests if the limit is reached and wait until comparisons already in progress have finished before attempting to fulfill them. When set to **true**, the server will respond with HTTP response status code **503 Too Busy** to any requests that are received while the maximum concurrency limit is reached.

Additionally, if **RejectIfTooBusy** is set to **true** and **BusyRejectDelaySeconds** has a value, then the server will wait for this amount of time before responding with HTTP response status code **503 Too Busy**. This delay provides an opportunity for a comparison in progress to finish and for the new comparison request to proceed. The default value for this setting is **1**.

DVJS comparison

Compare Server includes functionality to allow comparisons to be viewed in the browser. This technology is called DVJS (DeltaView JavaScript).

The DVJS functionality is enabled by default, but may be disabled using the **enableDVJS** setting in the web.config file. If you don't require this functionality with Compare Server, set **enableDVJS** to false.

***Note**: The result of a comparison performed using DVJS is a URL which can be navigated to in any modern web browser to show the results of the comparison. The URL contains a securely generated 192 bit comparison ID, which means that it is effectively impossible to guess the URL to see another user's comparison. However, sharing the results URL will allow any user who receives the URL to see the result of the comparison in their browser.*

Where to store results

Classic comparison requests return the comparison result directly to the caller and do not store the results on the Compare Server machine. However, DVJS comparisons require the results of the comparison to be stored on the server to allow them to be viewed in the browser. By default these files are stored in a cache folder located at:

C:\inetpub\WorkshareCompareApi\App_Data\cache

The location used to store these files can be customized by changing the **comparisonStorageConnectionString** setting in the web.config file. This setting should include the full path of the folder (or network share) to be used to store comparison results and associated resources, including queued comparisons.

***Note**: Changing this path will invalidate any recently run DVJS comparisons and any queued comparisons (see Queuing).*

How long to store results

Workshare Compare server automatically deletes the comparison results of DVJS comparisons at a fixed time after they were created. This time interval may be adjusted by altering the **comparisonRetentionTime** setting in the web.config file. This setting indicates the number of hours after which the comparisons are to be deleted from the server's storage. The minimum valid value for this setting is 1 hour (lower values will be treated as 1 hour).

***Note**: The cleanup runs on an intermittent basis (once every 15 minutes), so deletion may not occur immediately upon termination of the storage period for a particular comparison.*

Once a comparison result has been deleted, any user who is viewing the comparison will see errors if they try to refresh the browser page or perform actions such as:

- Saving the comparison result in any format
- Swapping the documents used in the comparison

Encrypting result files

By default, the comparison result files for DVJS comparisons as well as asynchronous queued comparisons are stored unencrypted on disk. Since the comparison IDs used to identify completed comparisons are cryptographically secure random strings with 192 bits of entropy, this does not pose a risk of one user accessing another user's comparison via the web site since the number of possible comparison IDs is so vast that the chance of guessing one correctly is effectively zero.

However, a user with desktop access to the Compare Server machine would be in a position to browse to the folder containing the results of completed comparisons and open or copy those comparison result files. If this risk is considered significant, you can enable encryption of the stored comparison result files by providing a value for the **comparisonEncryptionSalt** setting in the web.config file.

***Note**: Changing the EncryptionSalt setting will invalidate any recently run DVJS comparisons and any queued comparisons (see Queuing).*

If the **comparisonEncryptionSalt** setting is not blank, its value (along with the comparison ID) is used to generate an encryption key for each comparison performed. This encryption key is then used to encrypt the comparison results when stored to disk. Compare Server will automatically decrypt the results when the comparison is requested by a user's browser to show the correct results to the user.

***Note**: In theory, a highly-skilled attacker who has access to the Compare Server machine may be able to discover the algorithm used to derive the comparison encryption key from the ID and the **comparisonEncryptionSalt** setting. They could then recover the encryption keys for stored comparisons and decrypt their contents. Enabling this feature should therefore be considered as part of a well-designed 'defence in depth' strategy rather than a silver bullet to absolute security.*

Limits on redline size

If the comparison is more than 500 pages, the user will be offered links to download the comparison rather than viewing it in the browser. This page limit may be adjusted by altering the **dvjsComparisonPageLimit** setting in the web.config file.

Working out the number of pages in a very large document for the above check can be slow, so a quicker test is to look at the word count first. If that is clearly very high then there is no need to spend time counting pages. Business documents tend to have 200-300 words per page. Denser text like novels 300-400. There is a default of 500 words per page to give a first estimate of page count. If the estimate exceeds this limit then the document can be rejected quickly as too large without server CPU load. This words per page number can be adjusted by altering the **dvjsWordsPerPageEstimate** setting in the web.config file.

Logging

RESTful API log files

With a default install, the API log files are as follows:

Location	Purpose
C:\inetpub\logs\LogFiles\W3SVC1	Includes all requests coming into the REST endpoint
C:\inetpub\WorkshareCompareApi\App_Data\logs	Includes some requests and all errors

Analytics

By default, Workshare gathers anonymous usage statistics and environment information used towards product evolvement and improvement. If you want to disable this customer experience reporting, you can do so in the relevant config file.

Find the text `<add key="EnableCustomerExperienceReporting" value ="true"/>` in **web.config**.

Change the value to `false`.

Configuration for Legacy SOAP APIs

The Workshare Compare Server legacy SOAP APIs are configured via a configuration file. The location of this file depends on the type of installation:

- Where Compare Server is hosted in IIS: **C:\Program Files\Workshare\Compare Service\web.config**
- Where Compare Server is hosted as a Windows service: **C:\Program Files\Workshare\Compare Service\bin\Workshare.CompareService.ServiceHost.exe.config**

Note: *Workshare Support is available to assist you with any queries directly related to the Compare Server APIs. However, we do not provide support on issues related to third-party applications or customized add-ins.*

Licensing

Licensing for Compare Server is based on a service expiry date. The **product.lic** file, specified at the time of install, determines how the license is configured and when the license expires. The **product.lic** file is automatically copied into the Compare Server installation folder during setup and is required at runtime in order to authorize each comparison. Compare Server references this file via a path variable stored within the relevant config file. If you experience licensing errors, please check that the **product.lic** file is correctly installed in the Compare Service folder on the server and that the **web.config** or **Workshare.CompareService.ServiceHost.Exe.config** have a **LicenseFileName** key that correctly specifies the full path to the file.

For example:

```
<appSettings>
 <add key="LicenseFileName" value="C:\Program Files\Workshare\Compare Service\bin\product.lic" />
</appSettings>
```

Event Logging

On installation of Workshare Compare Server, the following log files are created:

- **compare_service_audit.log**: This log file contains information about the comparison requests sent to Compare Server, such as the size of the original and modified files, time taken to perform the comparison and so on.
- **compare_service_host.log**: This log file contains information such as when Compare Server started, when it was shut down, what protocols are available and so on.

- **compare_service_system.log**: This log records everything happening in Compare Server and includes information and error log entries.

Analytics

By default, Workshare gathers anonymous usage statistics and environment information used towards product evolvement and improvement. If you want to disable this customer experience reporting, you can do so in the relevant config file.

Find the text `<add key="EnableCustomerExperienceReporting" value ="true"/>` in **web.config** or **Workshare.CompareService.ServiceHost.Exe.config**.

Change the value to `false`.

Customizing the API

If you want to configure the API to use a different port or configure it to receive requests over https, you can make the required changes using IIS Manager.

You will find detailed Microsoft documentation for common IIS Manager tasks at: https://technet.microsoft.com/en-us/library/cc771341(v=ws.10).aspx. Although the documentation refers to IIS version 7, the instructions also apply to later versions of IIS.

Chapter 4: Monitoring

This chapter describes how to enable metrics on Workshare Compare Server and what metrics are published.

- **Introducing Metrics**, page 28, introduces the metrics exposed by Compare Server.
- **Enabling Metrics**, page 28, describes how to enable metrics on Compare Server.
- **Metrics Published**, page 29, describes the details of the metrics exposed.

Introducing Metrics

Workshare Compare Server includes a metrics endpoint to support integration with Prometheus. The metrics provide information about the activity of Compare Server such as the total number of comparisons performed, the number of comparisons currently in progress, the number of comparisons in a queue. This enables you to keep eye on Compare Server, monitor the load and identify potential problems.

Prometheus is open source software used to record real time metrics. Compare Server exposes metrics in a text-based format that Prometheus polls and stores enabling the data to be presented in graphs and charts.

Enabling Metrics

By default, this functionality is disabled and Compare Server doesn't publish metrics. You enable it in web.config.

Metrics can be retrieved by browsing to Compare Server's metrics API endpoint – http://server.name/path_to_root/api/metrics.

Typically, you can run a Prometheus server which will poll this API endpoint, get the data and graph it.

To enable metrics:

1. Find the text `<add key="EnableMetrics" value ="false"/>` in **web.config**.
2. Change the value to `true`.
3. Restart the Compare Server website in IIS.

***Note**: Restarting Compare Server will reset the counters. IIS may restart idle websites in the background to conserve resources. This would have the effect of resetting the metrics counters.*

Once enabled, Compare Server starts publishing metrics and Prometheus can poll the endpoint to retrieve them.

***Note**: Metrics are only published once the first health check or comparison is made, meaning the metrics endpoint will return a blank page until then.*

Metrics Published

Metrics are published in a text-based format. For example:

The metrics include the total number of comparisons run through Compare Server, the total number of PDF comparisons, the number of currently active comparisons, and more.

Format of metrics

The format of the string is:

```
[metric name]{[key]=”[value]”,[key]=”[value]”} [metric value]
```

Where

- Metric name: The name of the property being measured.
- Metric label: A key value pair where key is the variable name and value is the value of the variable.
- Value: The measure of this metric.

For example:

```
cs_comparisons_total{instance_name="examplename1",end_state=”success”} 7
```

Metric label

The following key value pairs may be used in the metrics:

- **instance_name**

  The name of the machine the comparison's running on. This is the hostname unless running on Azure App Svc in which case it will be the instance environment variable - WEBSITE_INSTANCE_ID. This is the id representing the VM that the site is running on. If the site runs on multiple instances, each instance will have a different id.

- **end_state**

  "success" when the operation completes successfully or "failure" when it doesn't.

Types of metrics

The metrics are one of two types:

- **Counter**: A cumulative metric whose value can only increase or be reset to zero on restart.
- **Gauge**: A metric whose value can go up and down.

Details of metrics

The different metrics exposed are described below:

cs_comparisons_total

Description	The total number of comparisons performed on this Compare Server since the last restart.
Type	Counter
Metric label	instance_name end_state
Example	cs_comparisons_total{instance_name="examplename1", end_state="success"} 7 cs_comparisons_total{instance_name="examplename1", end_state="failure"} 1

cs_pdfconversions_total

Description	The total number of PDF conversions performed on this Compare Server since the last restart. A PDF is converted to RTF before being compared.
Type	Counter
Metric label	instance_name end_state
Example	cs_pdfconversions_total{instance_name="examplename1", end_state=”success”} 25 cs_pdfconversions_total{instance_name="examplename1", end_state=”failure”} 2

cs_comparisons_active

Description	The number of comparisons currently active.
Type	Gauge
Metric label	instance_name
Example	cs_comparisons_active{instance_name="examplename1", 1

cs_pdfconversions_active

Description	The number of PDF conversions currently active. A PDF is converted to RTF before being compared.
Type	Gauge
Metric label	instance_name
Example	cs_pdfconversions_active{instance_name="examplename1", 1

cs_comparisons_queued

Description	The number of comparisons waiting in a queue to be compared.
Type	Gauge
Metric label	instance_name
Example	cs_comparisons_queued{instance_name="examplename1", 4

cs_workers_available

Description	The number of processors available to run comparisons.
Type	Gauge
Metric label	instance_name
Example	cs_workers_available{instance_name="examplename1", 8

cs_pdfconversionworkers_available

Description	The number of processors available to convert PDFs to RTF. A PDF is converted to RTF before being compared
Type	Gauge
Metric label	instance_name
Examples	cs_pdfconversionworkers_available{instance_name= "examplename1", 5

Chapter 5: Troubleshooting

This chapter describes how to troubleshoot Workshare Compare Server. It includes the following sections:

- **General Checks**, page 34, describes some general checks to perform when troubleshooting Compare Server.
- **Troubleshooting SOAP APIs**, page 35, describes how to deal with issues that might arise when using the legacy SOAP APIs.

General Checks

This section describes some basic checks to carry out in some general situations.

Installer fails

- Ensure any previous Workshare Compare Server installs are uninstalled.
- Is IIS started?
- Check the install logs in the current user's Temp folder, for example, C:\Documents and Settings\Administrator\Local Settings\Temp.

Swagger troubleshooting

If you are unable to access Swagger:

- Ensure that Javascript is enabled in your web browser.
- Use Chrome, or Firefox.
- Do not use Internet Explorer.

PDF Comparison

If you see display problems after comparing PDF files, you can try setting on the **Single Column Conversion** rendering set parameter.

PDF files are converted before comparison and sometimes columns are detected when there are none so that text ends up displayed vertically in columns. If the **Single Column Conversion** rendering set parameter is set (by default, it is off), columns are not detected in PDF source documents. This prevents potential inaccuracies. With this parameter set, if text looks like columns, an alternate approach is used to lay it out (such as a table or tabs).

Log file locations for RESTful APIs

The IIS logs at **C:\inetpub\logs\logfiles\W3SVC1** contain a basic record of all requests to the compare service.

The **C:\inetpub\WorkshareCompareApi\App_Data\logs** folder may contain the following files:

- Workshare.CompareApi.log: This is the main log file recording any information or errors in the Compare Server API module.
- Compare_service_engineworker.log. This includes logging from the actual comparison subsystem.

- Compare_service_audit.log. This records basic auditing information about each comparison performed.
- Compare_service_pdftoword.log. This includes logging from the conversion of PDF source documents to Word format for comparison.
- Compare_service_system.log. This inlcudes subsidiary logging from the modules supporting the API.

The most recent copy of each log file will have the ".log" extension. Older log files will have a date appended to the file extension.

Troubleshooting SOAP APIs

C# sample does not work

- Did you install legacy SOAP APIs?
- Check that the service address is correct and is appended with **/compare5**.
- If **Test Connection** returns an error, the issue may be security related:
  - Does the Java sample work? (The basic HTTP endpoint does not use client credentials.)
  - Are the user credentials valid on the server machine? You can test this by setting up a new, temporary user account and using those credentials.
  - Is there a firewall enabled on the server?
- If **Test Connection** works but **Compare** returns an error, the issue may be document related.
  - Do the default documents shipped in the **samples\data** folder compare correctly?
  - Are the compared documents very large? (try smaller)
- In all instances check the event logs (all keys) for error info.

Compare Service page doesn't display

If http://localhost/wcs/comparewebservice.svc does not display:

- Did you install legacy SOAP APIs?
- Is IIS started?
- Is the default website started? If the default website will not start, try changing the TCP port. If this is successful there is a port conflict with another server.

Comparing large files

The default configuration should support comparison of most source files up to 100MB in size. For extremely large and complicated files, you may need to increase the quotas and limits configured in the comparison engine and API.

To do this, edit the config files:

- For the comparison engine, either the **web.config** or **Workshare.CompareService.ServiceHost.Exe.config**
- For the API, the config file at (by default) C:\inetpub\WorkshareCompareApi

In these files, search for and replace:

`314572800`, which is 300Mb (ie 300x1024x1024). Replace this with a larger value to allow larger documents to be processed. This value is used in a number of places in each file as a size limit for message requests to the comparison engine.

`00:20:00`, ie 20 minutes. Replace this with a longer timeout to allow documents which take a very long time to be compared. This value is used in a number of places in each file as a timeout for requests to the comparison engine.

Note: *Even though an individual file may be smaller than the supported maximum, the total packaged transaction (original file + modified file + message details) may exceed this limit.*

SOAP endpoint issues

Client-side issues

The default maximum message size quota, generated by the Visual Studio Service Reference Wizard, is 64KB. This means that, although the server is configured to allow for large files, failures may occur if the **web.config** or **app.config** file of the client application is not updated.

When the client-side buffers are too small, the transport channel may throw one of the following exceptions:

`System.ServiceModel.CommunicationException` - “The maximum message size quota for incoming messages has been exceeded.”

`System.ServiceModel.FaultException` - “The maximum string content length quota has been exceeded.”

`System.ServiceModel.FaultException` - “The maximum array length quota has been exceeded.”

To resolve this issue, you should edit the **web.config** or **app.config** file for the client application and increase the binding's `maxReceivedMessageSize`, `maxStringContentLength` and `maxArrayLength` values.

***Note**: **web.config** applies if you have made a web site to use the SOAP endpoint; **app.config** applies if you have made a non-website application.*

Timeouts

Sometimes large files may fail to compare because the session has timed out. To extend the session, perform the following configuration:

1. Update the bindings section in the **web.config** file (where Compare Server is hosted in IIS) or in the **Workshare.CompareService.ServiceHost.Exe.config** file (where Compare Server is hosted as a Windows service) with the following.

   Note: The important part is the addition of **receiveTimeout** to "00:10:00" to both bindings.

```
    <bindings>
      <basicHttpBinding>
        <binding name="UnsecureBinding" closeTimeout="00:10:00"
openTimeout="00:10:00" sendTimeout="00:10:00"
receiveTimeout="00:10:00" maxBufferSize="67000000"
maxBufferPoolSize="67000000" maxReceivedMessageSize="67000000"
messageEncoding="Text" transferMode="Buffered">
          <readerQuotas maxStringContentLength="67000000"
maxArrayLength="67000000" maxBytesPerRead="67000000" />
        </binding>
      </basicHttpBinding>
      <wsHttpBinding>
        <binding name="SecureBinding" closeTimeout="00:10:00"
openTimeout="00:10:00" sendTimeout="00:10:00"
receiveTimeout="00:10:00" maxBufferPoolSize="67000000"
maxReceivedMessageSize="67000000" messageEncoding="Mtom">
          <readerQuotas maxDepth="67000000"
maxStringContentLength="67000000" maxArrayLength="67000000"
maxBytesPerRead="67000000" maxNameTableCharCount="67000000" />
          <reliableSession ordered="true" enabled="true" />
        </binding>
      </wsHttpBinding>
    </bindings>
```

2. Update the host section inside the **web.config** or **Workshare.CompareService.ServiceHost.Exe.config** file and increase the timeouts.

```
        <host>
          <timeouts closeTimeout="00:10:00" openTimeout="00:10:00"
/>
        </host>
```

3. Change the request execution in the IIS configuration:
   - In IIS go to the ASP.Net tab and select **Edit Configuration**.
   - Select the Application tab.
   - Change the **Request execution timeout(seconds)** to 600.

Workshare Ltd.
© 2019. Workshare Ltd. All rights reserved.

Copyright
Workshare Professional and Workshare DeltaView are registered trademarks of Workshare Ltd. Workshare Compare, Workshare Protect, Workshare 3, Workshare DeltaServer, SafetyGain, and the Workshare logo are trademarks of Workshare Ltd. All other trademarks are those of their respective holders.

Trademarked names may appear throughout this guide. Instead of listing these here or inserting numerous trademark symbols, Workshare wishes to state categorically that no infringement of intellectual or other copyright is intended and that trademarks are used only for editorial purposes.

Disclaimer
The authors/publishers of this guide and any associated help material have used their best efforts to ensure accuracy and effectiveness. Due to the continuing nature of software development, it may be necessary to distribute updated help from time to time. The authors would like to assure users of their continued best efforts in supplying the most effective help material possible.

The authors/publishers, however, make no warranty of any kind, expressed or implied, with regard to Workshare programs or help material associated with them, including this guide. The authors/publishers shall not be liable in the event of incidental or consequential damages in connection with, or arising out of, the programs or associated help instructions.

Revisions
Published for Workshare Compare Server 9.5: 14/9/18
Revised for Workshare Compare Server 9.5.1: 8/1/18
Revised for Workshare Compare Server 9.5.2: 17/5/18
Revised for Workshare Compare Server 9.5.3: 6/12/18
Revised for Workshare Compare Server 9.5.4: 18/2/19

Workshare Ltd., 20 Fashion Street, London E1 6PX www.workshare.com