



White Paper

Fabasoft on Microsoft Windows - Installation of Fabasoft Folio Services

Fabasoft Folio 2021 Update Rollup 1

Copyright © Fabasoft R&D GmbH, Linz, Austria, 2021.

All rights reserved. All hardware and software names used are registered trade names and/or registered trademarks of the respective manufacturers.

No rights to our software or our professional services, or results of our professional services, or other protected rights can be based on the handing over and presentation of these documents.

Contents

1 Introduction	6
2 Fabasoft Reference Architecture	6
3 Software and Hardware Requirements	7
3.1 Requirements for Installing Microsoft Windows Server	7
3.2 Requirements for Installing Fabasoft Folio Backend Services and a Fabasoft Folio Domain	7
3.3 Requirements for Installing Fabasoft Folio Web Services	8
3.4 Requirements for Installing Fabasoft Folio Conversion Services	8
3.5 Requirements for Installing Fabasoft Folio AT Services	8
4 Prerequisites	9
4.1 Prerequisites for Microsoft Windows Server	9
4.2 Prerequisites for Fabasoft Folio Backend Services and a Fabasoft Folio Domain	10
4.3 Prerequisites for Fabasoft Folio Web Services	11
4.4 Prerequisites for Fabasoft Folio Conversion Services	11
4.5 Prerequisites for Fabasoft Folio AT Services	12
5 Installation of Fabasoft Folio Backend Services and a Fabasoft Folio Domain	13
5.1 Installation of Fabasoft Folio Backend Services and a Fabasoft Folio Domain	13
5.2 Update of a Fabasoft Folio Domain	17
5.2.1 Migration	17
5.2.2 Update	18
5.3 Management of a Fabasoft Folio Domain	20
5.4 Load additional Fabasoft Folio Software Products	20
5.5 Removing a Fabasoft Folio Domain	22
5.6 Default Database Indexes	22
5.7 Necessary Permissions and Privileges	23
6 Installation and Configuration of Fabasoft Folio Web Services	24
6.1 Installation of Fabasoft Folio Web Services	24
6.2 Removing a Fabasoft Folio Web Service	27
7 Installation and Configuration of Fabasoft Folio Conversion Services	27
7.1 Overview	27
7.2 Installation Tutorial for Third-party Products	27
7.2.1 Enable Kerberos Authentication	27
7.2.2 Adjust the Regional and Language Options	28
7.2.3 Install and Configure Microsoft Office	29
7.2.4 Install and Configure Adobe Acrobat Pro	30
7.2.5 Install and Configure Ghostscript	31
7.2.6 Install and Configure FFmpeg	32

7.2.7 Install and Configure ExifTool	32
7.2.8 Install and Configure LibreOffice.....	32
7.2.9 Start Conversion Tools.....	33
7.2.10 User Settings.....	33
7.3 Installation of Fabasoft Folio Conversion Services	33
7.4 Configuring a Fabasoft Folio Conversion Service	33
7.4.1 Tools Host Configuration.....	34
7.4.2 Conversion Configuration	36
7.4.3 Default Web Services Configuration	38
7.4.4 Logging Mode	39
7.4.5 Tools Host State.....	39
7.4.6 Necessary Configuration Steps	40
7.4.7 Additional Configuration Options	42
7.5 Web Service Interface	42
7.6 Security Requirements	43
7.6.1 DCOM Security	43
7.6.2 File System Security Requirements.....	43
7.6.3 Security Requirements of Microsoft Office.....	44
7.6.4 Fabasoft Folio Security Requirements.....	44
7.7 Licensing Issues	44
7.8 Troubleshooting	45
7.8.1 Changing the Password.....	45
7.8.2 Changing the Web Service Identity.....	45
7.8.3 HTTP Client Issues	45
7.8.4 Kerberos Authentication Issues	46
7.8.5 Simultaneous Use as a Regular Web Service.....	47
7.8.6 Transaction Context	47
7.8.7 Enable the Logging Mode	48
7.8.8 Obtaining the Tools Host State	48
7.8.9 LibreOffice Issues	48
7.8.10 IIS Time Out Issues.....	48
7.8.11 Microsoft Office Issues.....	48
7.9 References.....	48
8 Installation and Configuration of Fabasoft Folio AT Services	49
8.1 Preparing Operations.....	49
8.1.1 User Account.....	49
8.1.2 Launch Permission in Microsoft Windows	49
8.2 Installation of Fabasoft Folio AT Services	51
8.3 Create Further Fabasoft Folio AT Services.....	53

8.4 Start a Fabasoft Folio AT Service.....	55
8.4.1 Manual Start.....	55
8.4.2 Automatic Start.....	56
8.5 Processing Automated Tasks.....	56
8.5.1 Prerequisites this Chapter is Based on.....	56
8.5.2 Settings for Processing Automated Tasks.....	57
8.5.3 List of Automated Tasks and Automated Tasks.....	58
8.5.4 Process.....	63
9 Uninstallation of Fabasoft Folio Services	64
10 Administration Tasks	64
10.1 Management of Fabasoft Folio Services.....	64
10.2 Management of the Fabasoft Folio Server Cache.....	64
10.3 Execution of Fabasoft app.ducx Expressions.....	65
10.4 Additional Administration Tasks.....	65
11 Troubleshooting	66
11.1 Fault Tolerant Heap (FTH).....	66

1 Introduction

This document describes preparing the installation of Fabasoft Folio Services as well as the installation and configuration of Fabasoft Folio Services in a Microsoft Windows Environment.

Chapter 2 “Fabasoft Reference Architecture” describes the Fabasoft reference architecture, including amongst others Fabasoft Folio Backend Services, Fabasoft Folio Web Services, Fabasoft Folio Conversion Services and Fabasoft Folio AT Services.

Chapter 3 “Software and Hardware Requirements” deals with assumed system environment and supported platforms as well as software the descriptions in this document are based on.

Chapter 4 “Prerequisites” describes how to lay the foundations for installing the software of which the installation is described in this document.

Chapter 5 “Installation of Fabasoft Folio Backend Services” describes the installation of Fabasoft Folio Backend Services and the steps necessary to create a Fabasoft Folio Domain.

Chapter 6 “Installation and Configuration of Fabasoft Folio Web Services” describes the installation and configuration of a Fabasoft Folio Web Service.

Chapter 7 “Installation and Configuration of Fabasoft Folio Conversion Services” describes how to install and configure a Fabasoft Folio Conversion Service. It outlines the technical background and ends with a troubleshooting section.

Chapter 8 “Installation and Configuration of Fabasoft Folio AT Services” describes how to install and configure Fabasoft Folio AT Services.

2 Fabasoft Reference Architecture

The Fabasoft product family is implemented as a distributed web application. The functionality of Fabasoft software products can be used via commercially available standard web browsers.

The incoming user queries are forwarded to the Fabasoft Folio Web Service by the Microsoft Internet Information Service. The Fabasoft Folio Web Service represents the business logic.

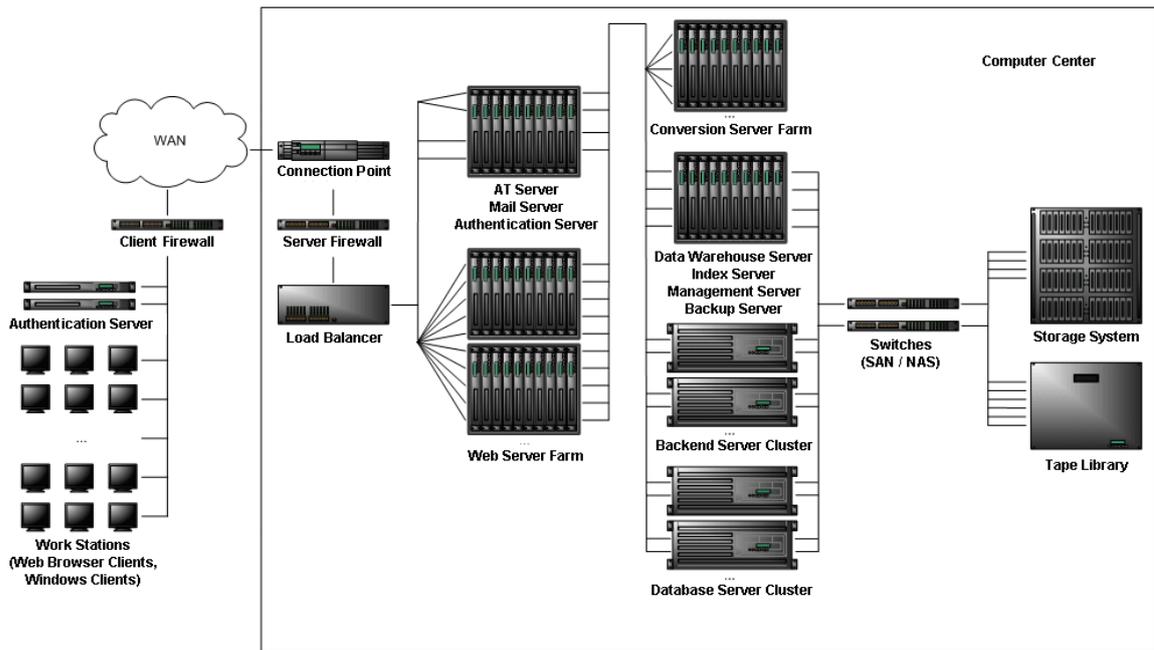
By means of the Fabasoft Folio Kernel, in the Fabasoft Folio Web Service the object model is available, object placement and access protection are defined and checked, properties are buffered, method calls are allocated and transactions are performed.

Fabasoft Folio Web Services process user queries by accessing the data managed by the Fabasoft Folio Backend Services.

Fabasoft Folio Backend Services (Fabasoft Folio COO Services and Fabasoft Folio MMC Services) are used for accessing and storing data in Fabasoft Folio Stores. Fabasoft Folio Stores are logical data stores storing business objects in a relational database system and in the file system.

Fabasoft Folio Conversion Services allow converting contents. Conversion queries are forwarded by the Fabasoft Folio Web Services to the Fabasoft Folio Conversion Services.

A Fabasoft Folio Domain consists of Fabasoft Folio Web Services, Fabasoft Folio Conversion Services and Fabasoft Folio Backend Services logically belonging together. Optionally Fabasoft Folio AT Services, Fabasoft iArchive Services and/or Fabasoft Folio application integration services (e.g. Fabasoft iArchiveLink Services) can also be included into the Fabasoft Folio Domain.



3 Software and Hardware Requirements

System environment: All information contained in this document implicitly assumes a Microsoft Windows environment.

Supported platforms: For detailed information on supported operating systems and software see the software product information on the Fabasoft distribution media.

Descriptions in this document are based on the software described in the following chapters.

3.1 Requirements for Installing Microsoft Windows Server

Third-party products:

- Microsoft Windows Server 2016 Standard (x64)
or
Microsoft Windows Server 2019 Standard (x64)

3.2 Requirements for Installing Fabasoft Folio Backend Services and a Fabasoft Folio Domain

Third-party products:

- Microsoft Windows Server 2016 Standard (x64)
or
Microsoft Windows Server 2019 Standard (x64)
- Microsoft .NET Framework 4.8
- AdoptOpenJDK OpenJDK 8 Update 282 (JRE, HotSpot)
Add `<ADOPTOPENJDKDIR>\bin` to the system environment variable `PATH`

3.3 Requirements for Installing Fabasoft Folio Web Services

Third-party products:

- Microsoft Windows Server 2016 Standard (x64)
or
Microsoft Windows Server 2019 Standard (x64)
- Microsoft .NET Framework 4.8
- Microsoft Internet Information Services (version corresponding to the operating system)
 - Including Common HTTP Feature: "Static Content"
 - Including Common HTTP Feature: "Default Document"
 - Including an appropriate Security Authentication
 - Including IIS Management Console
- AdoptOpenJDK OpenJDK 8 Update 282 (JRE, HotSpot)
Add `<ADOPTOPENJDKDIR>\bin` to the system environment variable `PATH`

3.4 Requirements for Installing Fabasoft Folio Conversion Services

Third-party products:

- Microsoft Windows Server 2016 Standard (x64)
or
Microsoft Windows Server 2019 Standard (x64)
- Microsoft .NET Framework 4.8
- Microsoft Internet Information Services (version corresponding to the operating system)
 - Including Common HTTP Feature: "Static Content"
 - Including Common HTTP Feature: "Default Document"
 - Including an appropriate Security Authentication
 - Including IIS Management Console
- AdoptOpenJDK OpenJDK 8 Update 282 (JRE, HotSpot)
Add `<ADOPTOPENJDKDIR>\bin` to the system environment variable `PATH`
- Microsoft Office 2016 (64 bit)
or
Microsoft Office 2019 (64 bit)
- Adobe Acrobat DC Pro (32 bit)
or
Ghostscript 9.53 (64 bit)
- LibreOffice 6.4.6 (32 bit)
- FFmpeg 4.3 (x64)
- QCAD 3.25 (x64)
- ExifTool 12.06 (x64)

3.5 Requirements for Installing Fabasoft Folio AT Services

Third-party products:

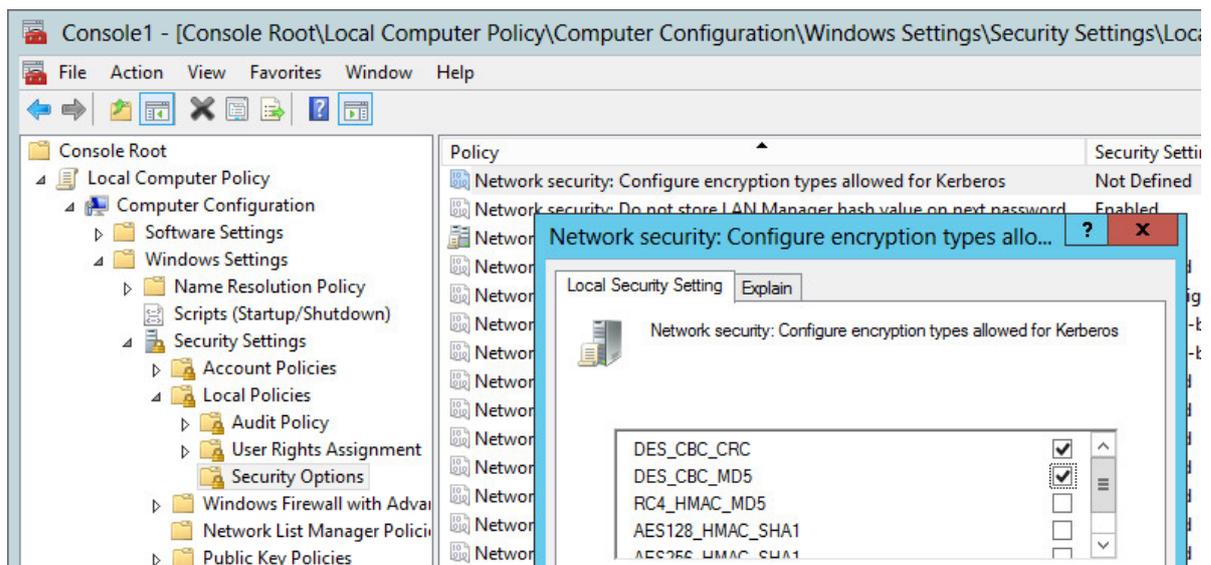
- Microsoft Windows Server 2016 Standard (x64)
or
Microsoft Windows Server 2019 Standard (x64)
- Microsoft .NET Framework 4.8
- AdoptOpenJDK OpenJDK 8 Update 282 (JRE, HotSpot)
Add <ADOPTOPENJDKDIR>\bin to the system environment variable PATH

4 Prerequisites

4.1 Prerequisites for Microsoft Windows Server

- This document assumes the utilization of a Microsoft Windows Active Directory domain controller.
- Make sure that the BIOS option to first try to start from CD-ROM is enabled.
- Make sure that the NTFS file system is selected during the Installation of Microsoft Windows
- Prepare the following information necessary during the installation and/or preparation of Microsoft Windows Server before beginning the installation:
 - IP address of the computer Microsoft Windows Server is installed on
 - Host name of the computer Microsoft Windows Server is installed on
 - IP address of the gateway server
 - IP address(es) of the DNS server(s)
 - Domain name
 - IP address of the domain controller
- For running Fabasoft Folio Services with Kerberos authentication, ensure that the enabled encryption types match the domain controller security configuration.

Note: Encryption types are not enabled by default but can be enabled in the Group Policy Object Editor (Network security: Configure encryption types allowed for Kerberos).

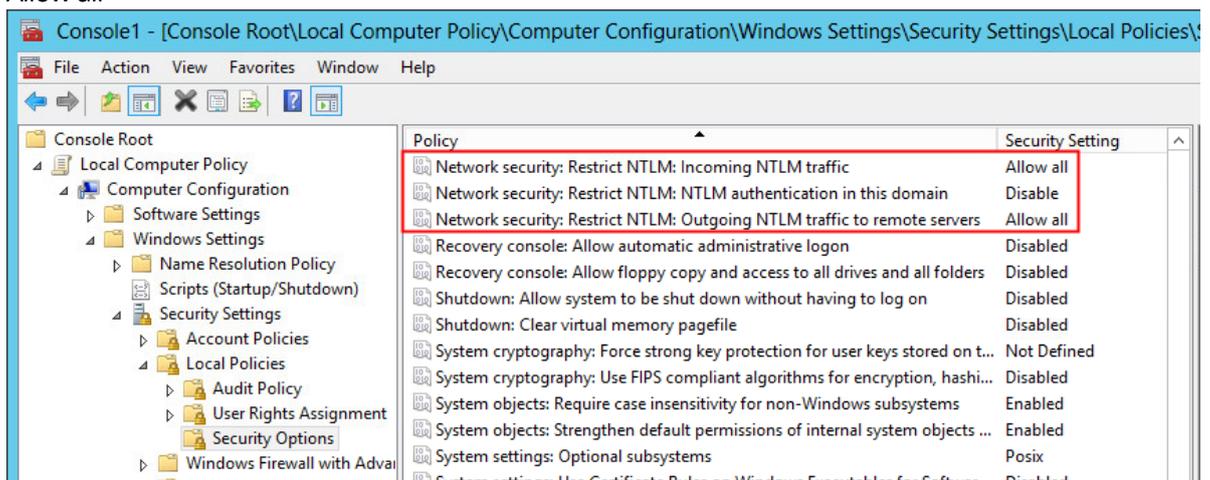


4.2 Prerequisites for Fabasoft Folio Backend Services and a Fabasoft Folio Domain

- Make sure that all third-party software products listed in chapter 3.2 “Requirements for Installing Fabasoft Folio Backend Services” have been installed.
- All Fabasoft Folio software packages from the distribution media have to be available on a file system local on the machine (hard disk, CD, DVD etc.) or on a network share.
- A valid Fabasoft Folio license key is required as well.
- A user account (e.g. fscsv) for running the Fabasoft Folio Backend Services has to be created in the operating system environment.

Note: If integrated authentication is used (default) and Kerberos is required, an SPN for the ADERPC service must be set for the Fabasoft Folio Backend Service user account. If NTLM authentication is required or sufficient, NTLM must be enabled in the Group Policy Object Editor.

- Network security: Restrict NTLM: Incoming NTLM traffic
Allow all
- Network security: Restrict NTLM: NTLM authentication in this domain
Disable
- Network security: Restrict NTLM: Outgoing NTLM traffic to remote servers
Allow all



Databases

- This document describes the installation when using Microsoft SQL Server databases. Therefore a preconfigured working database server (as described in the document “Installation and Configuration of Microsoft SQL Server”) is required.
- Alternatively Oracle Database can be used on the database server. Information about installation and configuration of Oracle Database can be found in the white paper “Fabasoft on Microsoft Windows and Linux - Oracle Database Preparation Guide”. Make sure that the NLS_LANG parameter used by the Oracle Instant Client is set to “AMERICAN_AMERICA.AL32UTF8”. Most commonly this is done via the following registry key:
HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE\<home ID, e.g. KEY_Oradb12c_home1>\NLS_LANG
Type: REG_SZ
Value: AMERICAN_AMERICA.AL32UTF8
More information about the NLS_LANG parameter can be found here:
<http://www.oracle.com/technetwork/products/globalization/nls-lang-099431.html>

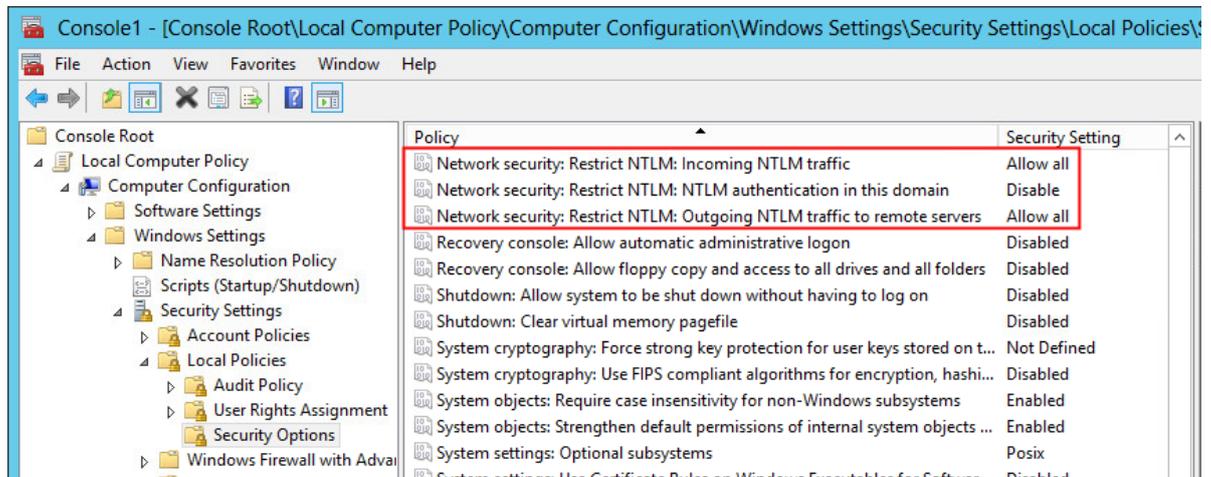
4.3 Prerequisites for Fabasoft Folio Web Services

- Make sure that all third-party software products listed in chapter 3.3 “Requirements for Installing Fabasoft Folio Web Services” have been installed.
- Working Fabasoft Folio Backend Services as described in chapter 5 “Installation of Fabasoft Folio Backend Services” have to be available.
- A user account (e.g. fscwebsrv) for running the Fabasoft Folio Web Services has to be created in the operating system environment and in the Fabasoft Folio Domain.

Note: If the Fabasoft Folio Web Services are installed at the same computer as the Fabasoft Folio Backend Services and the same user is performing the installation, for the user in whose context the Fabasoft Folio Web Services will be running a user object is created automatically in the Fabasoft Folio Domain.

Note: If integrated authentication is used and Kerberos is required, an SPN for the HTTP service must be set for the Fabasoft Folio Web Services user account. If NTLM authentication is required or sufficient, NTLM must be enabled in the Group Policy Object Editor.

- Network security: Restrict NTLM: Incoming NTLM traffic
Allow all
- Network security: Restrict NTLM: NTLM authentication in this domain
Disable
- Network security: Restrict NTLM: Outgoing NTLM traffic to remote servers
Allow all

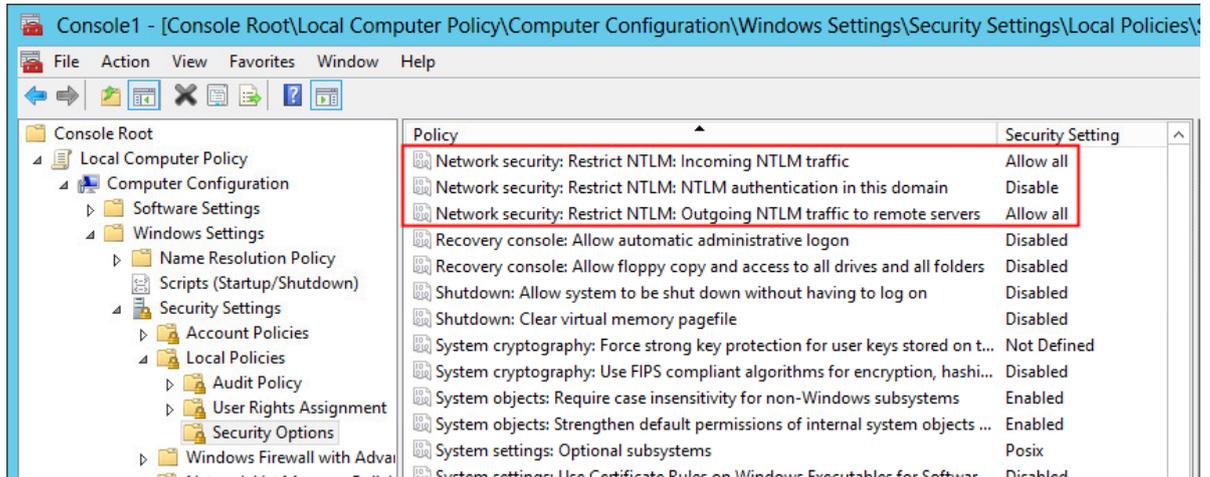


4.4 Prerequisites for Fabasoft Folio Conversion Services

- Make sure that all third-party software products listed in chapter 3.4 “Requirements for Installing Fabasoft Folio Conversion Services” have been installed.
- Working Fabasoft Folio Web Services as described in chapter 6 “Installation and Configuration of Fabasoft Folio Web Services” have to be available.
- A user account (e.g. fscconvsrv) for running the Fabasoft Folio Conversion Services has to be created in the operating system environment and in the Fabasoft Folio Domain.

Note: If integrated authentication is used and Kerberos is required, an SPN for the HTTP service must be set for the Fabasoft Folio Web Services user account. If NTLM authentication is required or sufficient, NTLM must be enabled in the Group Policy Object Editor.

- Network security: Restrict NTLM: Incoming NTLM traffic
Allow all
- Network security: Restrict NTLM: NTLM authentication in this domain
Disable
- Network security: Restrict NTLM: Outgoing NTLM traffic to remote servers
Allow all



4.5 Prerequisites for Fabasoft Folio AT Services

- Make sure that all third-party software products listed in chapter 3.5 “Requirements for Installing Fabasoft Folio AT Services” have been installed.
- A user account (e.g. fscatsrv) for running the Fabasoft Folio AT Services has to be created in the operating system environment and in the Fabasoft Folio Domain.
Note: If the Fabasoft Folio AT Services are installed at the same computer as the Fabasoft Folio Backend Services and the same user is performing the installation, for the user in whose context a Fabasoft Folio AT Service will be running a user object is created automatically in the Fabasoft Folio Domain.
- *Fabasoft Folio/AT* consists of a software product package and a setup for the installation of a Microsoft Windows system service (see chapter 8 “Installation and Configuration of Fabasoft Folio AT Services”).

Note: Make sure the software product *Fabasoft Folio/AT* has been installed in the Fabasoft Folio Domain. To check whether the software product *Fabasoft Folio/AT* has been installed, perform the following steps:

1. In the *Domain Administration*, navigate to the *Domain Objects* and click the “Software Components” tab.
2. If the software component *Fabasoft Folio/AT* has been installed, an *Automated Tasks* entry is available.

Note: To refresh the content of this list open the “Edit” menu and click “Refresh”.

5 Installation of Fabasoft Folio Backend Services and a Fabasoft Folio Domain

5.1 Installation of Fabasoft Folio Backend Services and a Fabasoft Folio Domain

To install Fabasoft Folio Backend Services via the automated setup, perform the following steps:

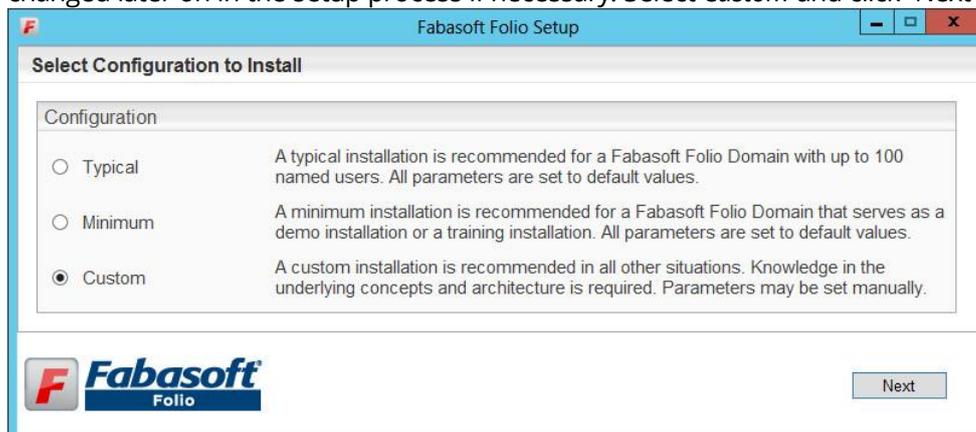
1. Insert the Fabasoft product DVD in the DVD drive of your server. The Fabasoft Server Setup dialog will be displayed automatically.

Note: If this dialog is not displayed automatically, run `setup.exe` in the root directory of the Fabasoft product DVD.

2. Accept the Fabasoft Folio legal terms and click "Next".

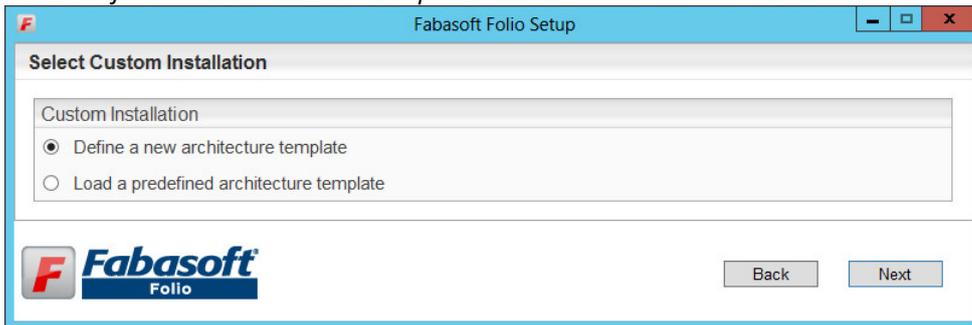


3. The configuration of the installed Fabasoft Folio system is based on a template. The template defines how many Fabasoft Folio COO Services and Fabasoft Folio MMC Services are installed. Additionally default values for the Fabasoft Folio Services are provided. The settings may be changed later on in the setup process if necessary. Select *Custom* and click "Next".

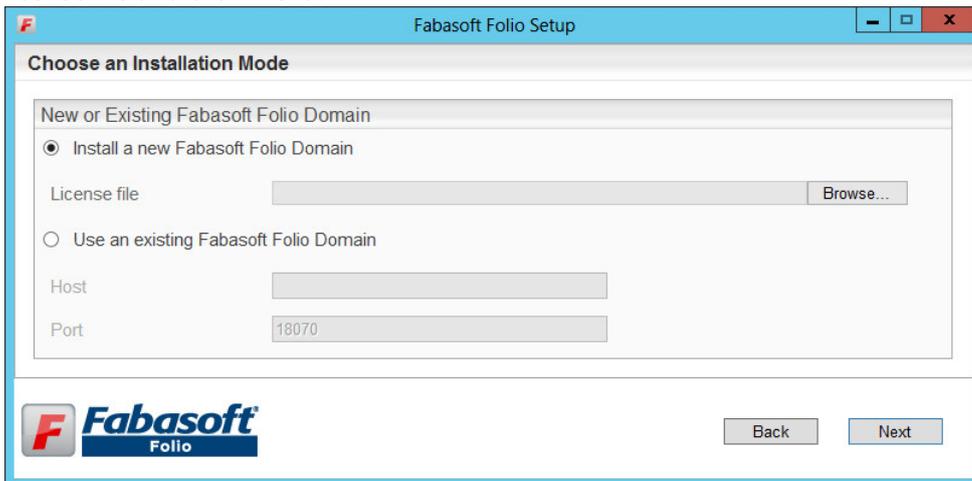


- o *Typical*
Select this option to install one Fabasoft Folio COO Service (with four Fabasoft Folio COO Stores) and one Fabasoft Folio MMC Service (with one Fabasoft Folio MMC Store).

- *Minimum*
Select this option to install one Fabasoft Folio COO Service (with one Fabasoft Folio COO Store) and one Fabasoft Folio MMC Service (with one Fabasoft Folio MMC Store).
 - *Custom*
Select this option to install by default one Fabasoft Folio COO Service (with four Fabasoft Folio COO Stores) and one Fabasoft Folio MMC Service (with two Fabasoft Folio MMC Stores).
4. The option *Define a new architecture template* can be selected to create a new template based on default values. *Load a predefined architecture template* can be used to select a saved template file of a previous installation. This step is omitted if a typical or minimum installation is made. Select *Define a new architecture template* and click "Next".



5. In the next dialog two options are available. Select *Install a new Fabasoft Folio Domain*, specify a license file and click "Next".



- *Install a new Fabasoft Folio Domain*
Select this option to install Fabasoft Folio Backend Services. In the *License file* field, type the path to the license file or click "Browse" to browse the file system.
- *Use an existing Fabasoft Folio Domain*
Select this option to use an existing Fabasoft Folio backend installation.

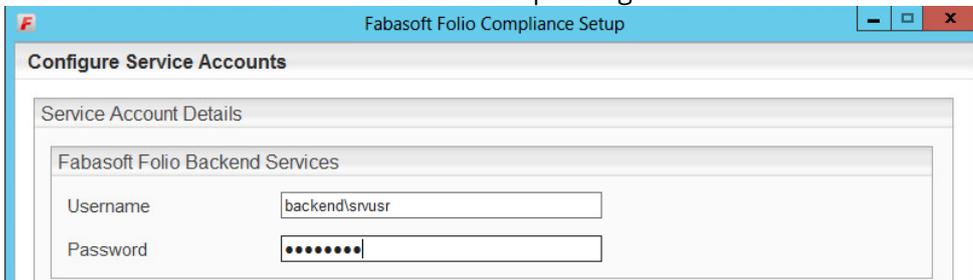
- Clear all check boxes except of *Fabasoft Folio Server Management*, *Fabasoft Folio Backend Services*, *Fabasoft Folio Domain* and *Fabasoft Folio Kernel* and click "Next".



- A check on all prerequisites is performed. Its result is presented in an overview. Ensure that your system complies with all requirements (indicated by a green tick) and click "Next".

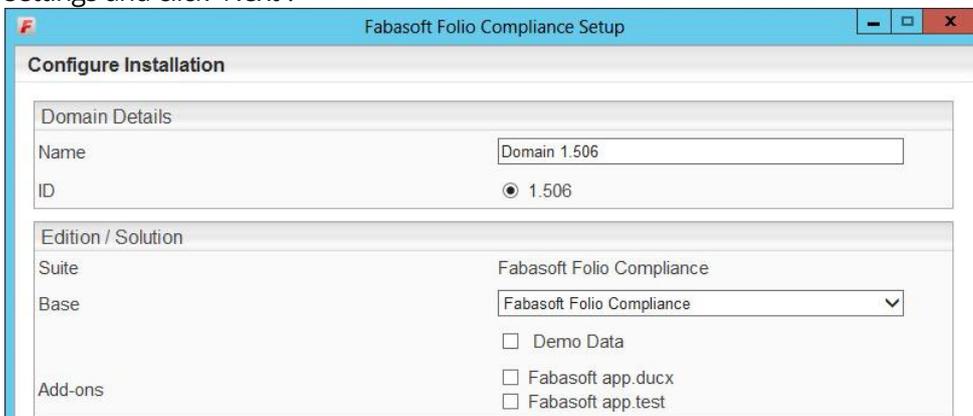


- Type the *Username* and the *Password* of the user account that should be used for running the Fabasoft Folio Backend Service in the corresponding fields and click "Next".



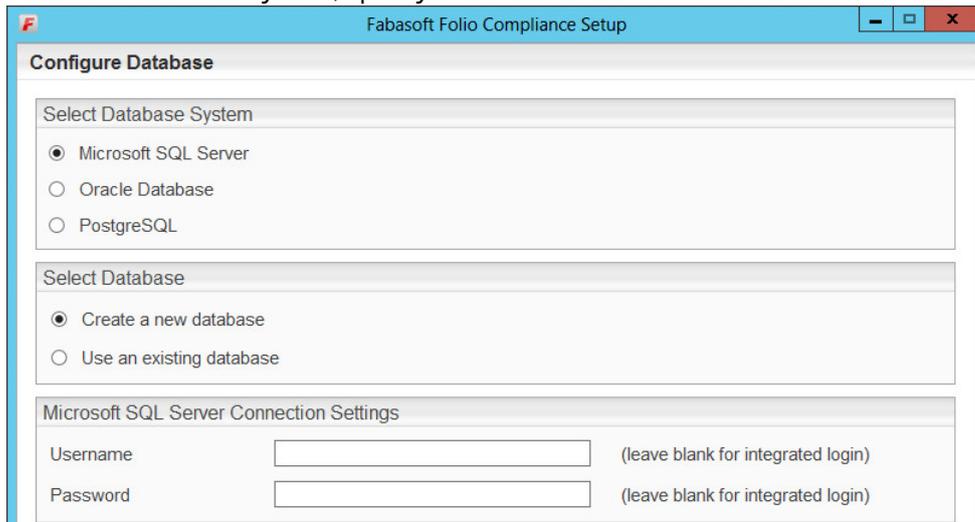
Note: This service user must exist as an operating system user.

- Specify the settings for the created Fabasoft Folio Domain. Specify your Fabasoft Folio Domain settings and click "Next".



- Under *Edition*, in the *Edition/Solution* dropdown list select a pre-configuration for the Fabasoft Folio Domain.
- Under *Domain Details* a *Name* for the Fabasoft Folio Domain can be specified and an *ID* can be selected.
Note: The available domain IDs depend on the license file used.
- In the *Load demo data* field specify whether demo data is loaded or not.
- In the Add-ons section specify an additional Fabasoft product if needed.

10. Select the database system, specify whether to create a new database and click "Next".



11. In the *Review Configuration*, the Fabasoft Folio Services can be configured. To change the installation directory, click "Edit" next to *Common Product Settings* and type in the desired installation path. If you want to create additional Fabasoft Folio MMC Services and COO Services, click "Edit" next to *Fabasoft Folio Backend Services*. Otherwise click "Install" to start the installation. In this case the next steps can be omitted.



12. The following options are available to configure the Fabasoft Folio Domain. Click the name of a Fabasoft Folio COO Service, to edit the settings.

- A Fabasoft Folio Backend Service can be edited by clicking the name of the service or by clicking "Edit".
- A Fabasoft Folio Backend Service can be deleted by clicking "Remove".
- A Fabasoft Folio Backend Service can be created by clicking the plus icon.
- Under *Fabasoft Folio COO Service*, *Service name* and *Service port* of a Fabasoft Folio COO Service can be edited.

Using Microsoft SQL Server, the following options are available:

- *Datasource*: Logical database interface.
- *Database*: The name of the database.
- *Create new database*: Specifies whether to create a new database.
- *Data directory*: The path to the data directory of the database.
- *Log directory*: The path to the log directory of the database.

- Under *Fabasoft Folio COO Stores*, the currently available Fabasoft Folio COO Stores are displayed.
A Fabasoft Folio Store can be deleted by clicking “Remove”.
To add a Fabasoft Folio COO Store, a name and a corresponding Fabasoft Folio MMC Store have to be specified. To create a Fabasoft Folio Gateway, select *Gateway*.
13. To close the dialog for configuring the Fabasoft Folio Backend Services, click “OK”.
 14. Click “Install” to start the installation.
The Fabasoft Folio Backend Services are installed.

5.2 Update of a Fabasoft Folio Domain

5.2.1 Migration

When updating Fabasoft Folio from a version below 2012 Summer Release to a version equal or higher 2012 Summer Release following migration steps have to be considered.

From version 2012 Summer Release numerical values larger than $2^{32}-1$ (or $2^{31}-1$ for negative numbers) are supported. In previous versions numbers are stored as 32 bit values. Since 1 bit is used for the sign (+ or -), for the number range only 31 bits remain. In Fabasoft Folio it can be defined for number properties, whether they use a sign or not. If no sign is used, the negative value range is used for storing large numbers.

In the course of the conversion from 32 bit to 64 bit, those large values that are stored as negative numbers must be migrated. The identification of affected properties is done via the tool `upgr122`. As part of an update by the Fabasoft Folio Setup this tool will run automatically.

Attention: In the following chapter two scenarios are described that need further manual steps.

5.2.1.1 Manual Steps

Scenario 1: Affected values exist only in versions or in the archive

The automated identification only considers integer properties that contain values of the affected number range in the current version. For integer properties that potentially contain values of the affected number range only in versions or in the archive, a migration CSV file has to be created manually.

The CSV file must have the following format:

```
version;upgradetype;objidold;objidnew;tabidold;tabidnew
1223;9;<object address 1>;<object address 1>;1;1
1223;9;<object address 2>;<object address 2>;1;1
and so on
```

The object address of the integer property can be set as 64 bit ID or as COO address.

Example:

```
1223;9;285774255974402;285774255974402;1;1
or
1223;9;COO.1.1001.1.223234;COO.1.1001.1.223234;1;1
```

The CSV file has to be stored in the migration folder (`<server directory>/CooSrv/files/migration` OR `/var/opt/fabasoft/instances/files/migration`) on all backend servers before running the setup, such that the migration is carried out as part of the COO service start-up.

Scenario 2: Fabasoft Folio Backend Services are distributed on several clusters

If Fabasoft Folio Backend Services are distributed across multiple clusters, a manual execution of the tool `Setup\ComponentsBase\Services\ is necessary on each cluster. After each run, the output file (phase=1: attridfile; phase=2: csvfile) has to be used as input file for the following execution:`

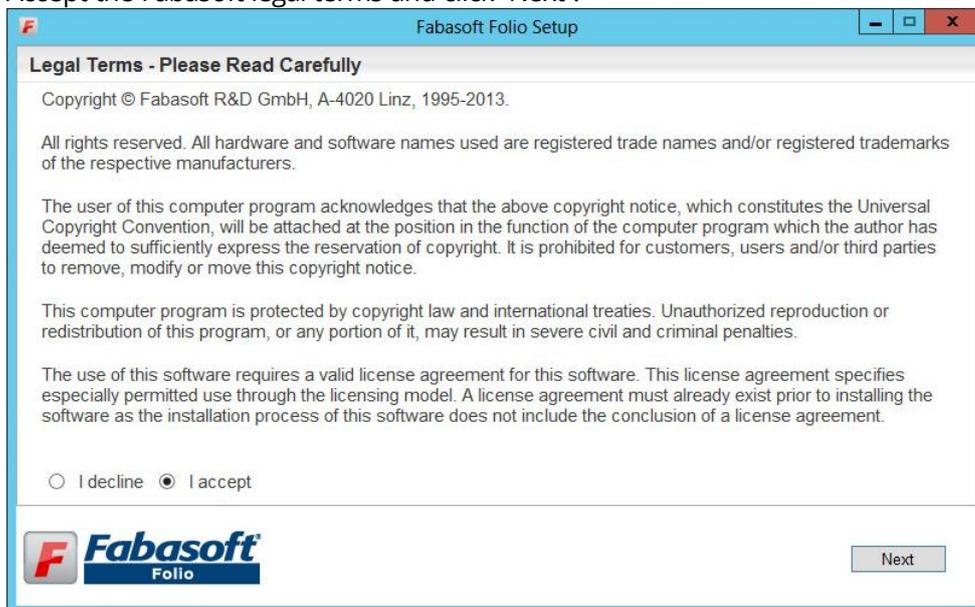
1. Determining a list of potentially affected properties (execute on each cluster):
`upgr122 phase=1 attridfile=attrids.txt`
2. Determine which properties are actually values in the numerical range of $2^{31} - 2^{32}-1$ (execute on each cluster):
`upgr122 phase=2 attridfile=attrids.txt csvfile=UINT322INT64.csv`

The result file (`UINT322INT64.csv`) has to be stored in the migration folder (`<server directory>/CooSrv/files/migration`) on all backend servers before running the setup, such that the migration is carried out as part of the COO service start-up.

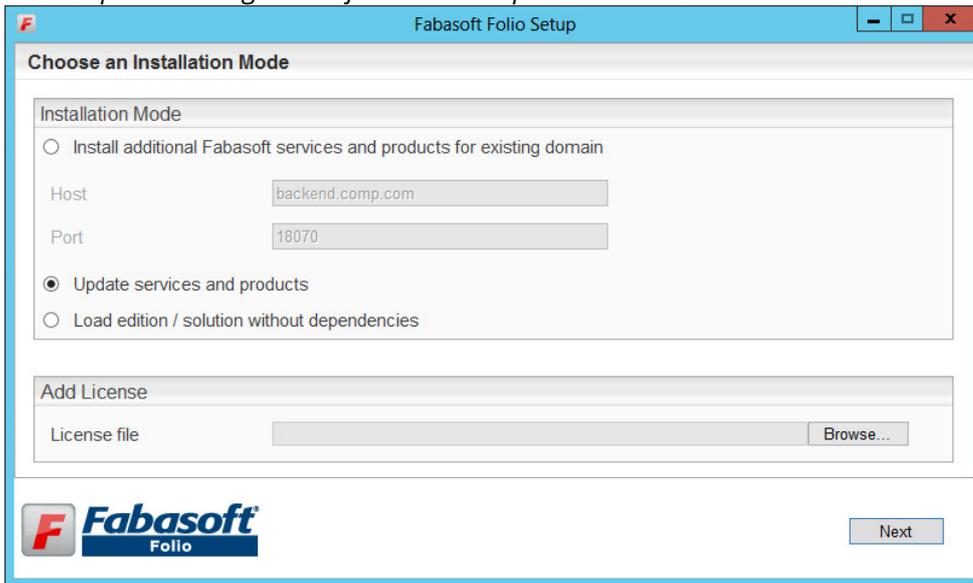
5.2.2 Update

To update a Fabasoft Folio Domain via the automated setup, perform the following steps:

1. Insert the Fabasoft product DVD in the DVD drive of your server. The Fabasoft Server Setup dialog will be displayed automatically.
Note: If this dialog is not displayed automatically, run `setup.exe` in the root directory of the Fabasoft product DVD.
2. Accept the Fabasoft legal terms and click "Next".



3. Select *Update existing Fabasoft services and products* and click "Next".



If an additional license file should be loaded, specify the path to the license file and click "Next".

4. Select the Fabasoft Folio Services which are to be updated.



5. A check on all prerequisites is performed. Its result is presented in an overview. Ensure that your system complies with all requirements (indicated by a green tick) and click "Update".



The Fabasoft Folio Domain is updated.

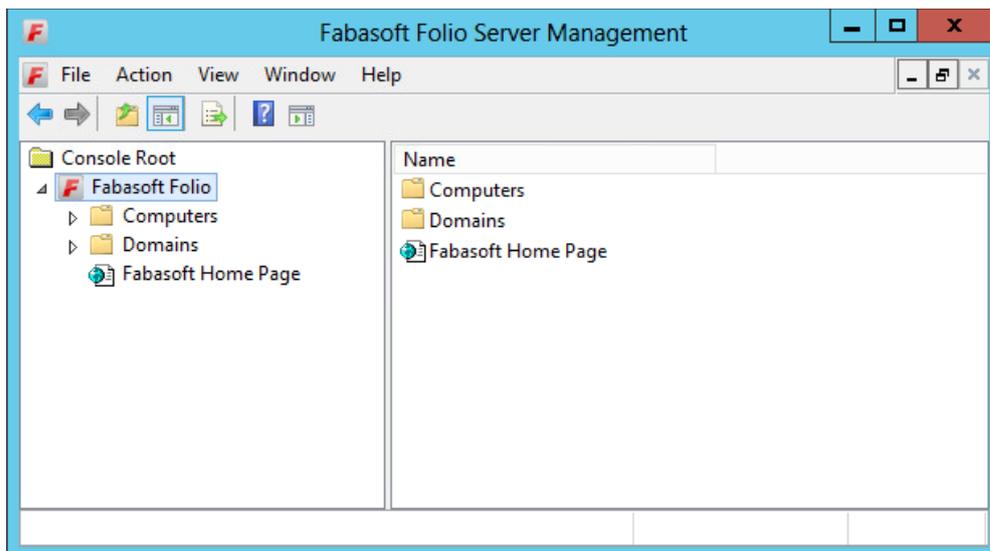
5.3 Management of a Fabasoft Folio Domain

A Fabasoft Folio Domain can be created and managed using the Fabasoft Folio Server Management MMC snap-in.

Run the “Fabasoft Folio Server Management”.



The Fabasoft Folio Server Management is opened.

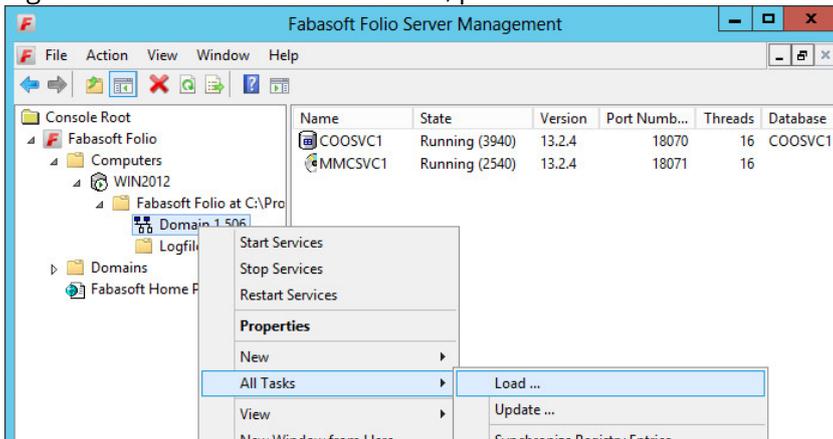


5.4 Load additional Fabasoft Folio Software Products

The Fabasoft Folio Server Management allows loading additional Fabasoft Folio software products into an existing Fabasoft Folio Domain. Perform the following steps:

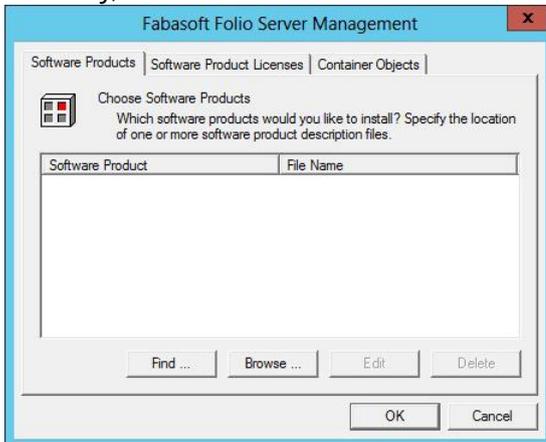
1. Run the “Fabasoft Folio Server Management”.
2. Navigate to “Fabasoft Folio” > “Computers” > “<computer name>” > “Fabasoft Folio at <installation directory>” > “<domain name>”.

- Right-click the Fabasoft Folio Domain, point to "All Tasks" and click "Load".



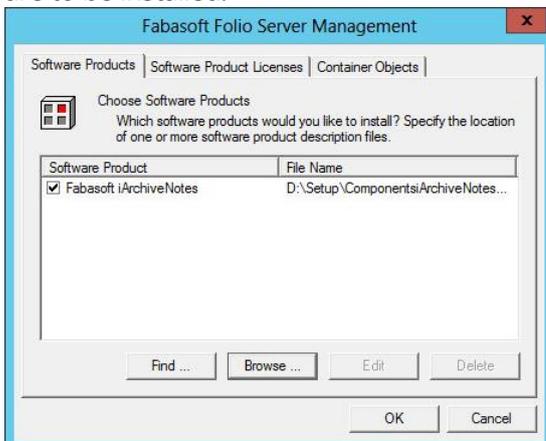
Loading Software Products

- Click "Browse" to search for a software product file in the file system. Click "Find" to specify a directory, which should be searched for software product files.

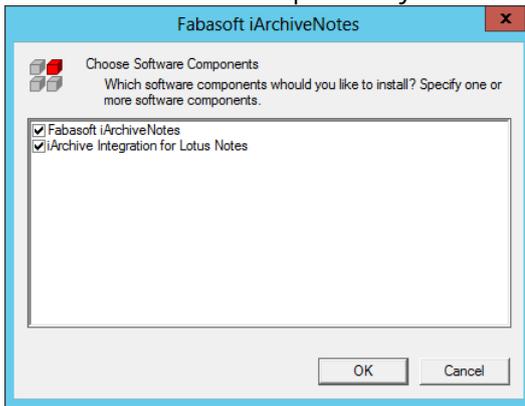


Note: Select *Look for installed software products only* to search for software product files of already installed software products only.

- Double-click a software product entry to define which of the contained software components are to be installed.



3. Select the software components you want to install and click "OK".



Loading Software Product Licenses

1. Click the "Software Product Licenses" tab.
2. Click "Find" or "Browse" to browse for a software product license file in the file system.

Loading Container Objects

1. Click the "Container Objects" tab to add single container objects (COO files), for example software components that do not belong to a software product.
2. Click "Browse" to search for the container object file in the file system.
3. Click "OK" to load the selected software products, software product licenses or container objects.
4. Click "Yes" to confirm that objects should be loaded into the Fabasoft Folio Domain.

5.5 Removing a Fabasoft Folio Domain

To remove a Fabasoft Folio installation the Fabasoft Folio Domain and the installed Fabasoft Folio stores and gateways have to be deleted.

Note: When deleting a Fabasoft Folio Domain the Fabasoft Folio stores and gateways are destroyed irrevocable.

To delete a Fabasoft Folio Domain perform the following steps:

1. Run the "Fabasoft Folio Server Management".
2. Navigate to "Fabasoft Folio" > "Computers" > "<computer name>" > "Fabasoft Folio at <installation directory>" > "<domain name>".
3. Right-click the Fabasoft Folio Domain and click "Delete".
4. Click "Yes" to confirm that the selected Fabasoft Folio Domain should be deleted.

The Fabasoft Folio Domain is deleted as well as all contained Fabasoft Folio stores and Fabasoft Folio gateways.

5.6 Default Database Indexes

Each Fabasoft Folio COO Service creates default database indexes which are documented in the `indexes.txt` file located in `<installation directory>\CooSrv\db`.

The index usage for columns of type VARCHAR or NVARCHAR depends on the property “Queries Are Case-Insensitive” (COOSYSTEM@1.1:domaincisqry) which is configurable in the current domain object. When changing the value of this property, the indexes should be changed accordingly.

5.7 Necessary Permissions and Privileges

Logon as a Service

The Fabasoft Folio Backend Service account must hold the `SeLogonServiceRight` privilege (Logon as a service).

Registry

The Fabasoft Folio Backend Service account needs the following registry permissions:

Resource	Minimum resource permission required
HKEY_LOCAL_MACHINE\Software\Fabasoft\Fabasoft Components Server	Read (This key and all sub keys and values)
HKEY_LOCAL_MACHINE\Software\Fabasoft\Fabasoft Components Server\ Domain <domainmajorid>.<domainminorid>\ Service <serviceinstance>	Full access (This key and all sub keys and values)

File System

The Fabasoft Folio Backend Service account needs the following file system permissions:

Resource	Minimum resource permission required
<Fabasoft Folio Installation Folder>	Read and execute (This folder and all subfolders and files)
<Fabasoft Folio Installation Folder>\coosrv\files	Full access
%ProgramData%\Fabasoft	Full access

Microsoft Windows Active Directory

The Fabasoft Folio Backend Service account needs the following Microsoft Windows Active Directory permissions:

Resource	Minimum resource permission required
ForeignSecurityPrincipals	Read All Properties (Child objects only)
System	Read All Properties (Child objects only, Apply these permissions to objects and/or containers within this container only)
System\FileLinks\VolumnTable	Read All Properties (This object and all child objects)

System\WinsockServices	Read All Properties, Write All Properties, Create All Child Objects, Delete All Child Objects (This object and all child objects)
------------------------	---

Note: In the Microsoft Windows Active Directory the Microsoft SQL service user needs the right “Read Group Membership” on the user account of the Fabasoft Folio Backend Services.

Cluster

The Fabasoft Folio Backend Service account must have full access to the cluster if global MSDTC coordinated transactions are configured in a clustered environment.

Fabasoft Folio COO Service Specific Permissions

The Fabasoft Folio COO Service account must have database access permissions and must be the owner of the database objects created by the Fabasoft Folio COO Service.

Fabasoft Folio MMC Service Specific Permissions

The Fabasoft Folio MMC Services account needs the following additional file system permissions:

Resource	Minimum resource permission required
<MMC Service Data Area Folder>	Full access (This folder and all subfolders and files)

Fabasoft Folio Server Management Service

The Fabasoft Folio Server Management Service impersonates the logged-on user and must run under the “Local System Account”.

6 Installation and Configuration of Fabasoft Folio Web Services

6.1 Installation of Fabasoft Folio Web Services

To install Fabasoft Folio Web Services via the automated setup, perform the following steps:

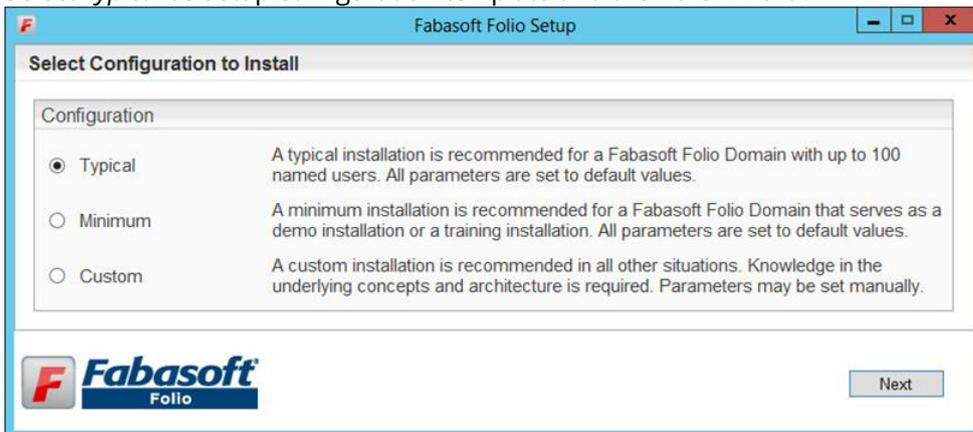
1. Insert the Fabasoft product DVD in the DVD drive of your server. The Fabasoft Server Setup dialog will be displayed automatically.

Note: If this dialog is not displayed automatically, run `setup.exe` in the root directory of the Fabasoft product DVD.

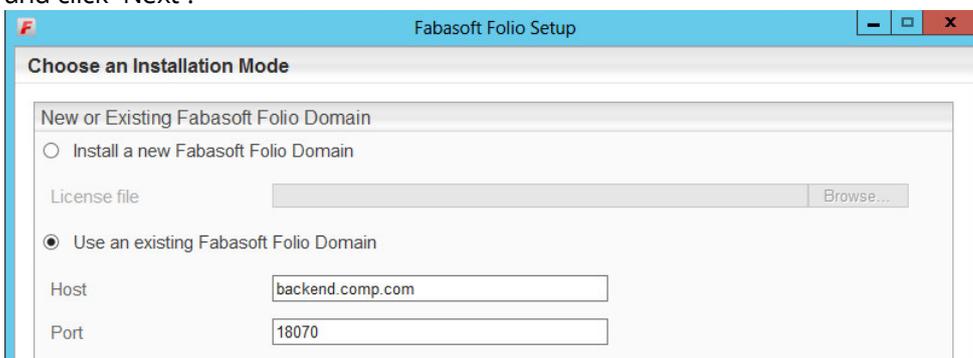
2. Accept the Fabasoft legal terms and click "Next".



3. Select *Typical* as setup configuration template and then click "Next".



4. In the next dialog two options are available. Select *Use an existing Fabasoft Folio Domain*, in the *Host* field type the hostname of the Fabasoft Folio Backend Server, in the *Port* field type its port and click "Next".

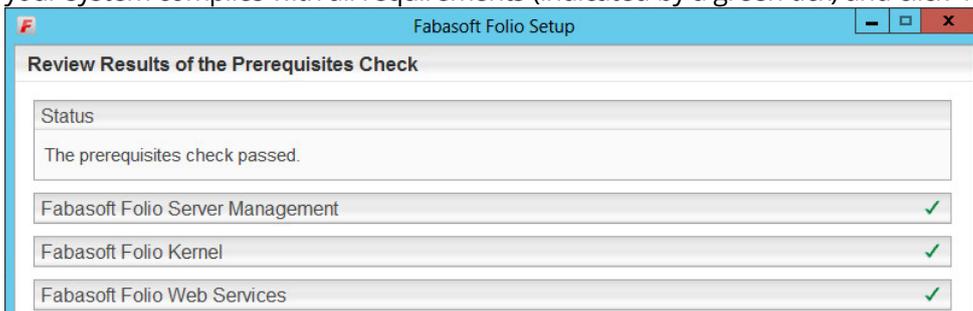


- *Install a new Fabasoft Folio Domain*
Select this option to install Fabasoft Folio Backend Services. In the *License file* field, type the path to the license file or click "Browse" to browse the file system.
- *Use an existing Fabasoft Folio Domain*
Select this option to use an existing Fabasoft Folio backend installation.

5. Select *Fabasoft Folio Web Services*, clear the checkboxes of all other services and click "Next".



6. A check on all prerequisites is performed. Its result is presented in an overview. Ensure that your system complies with all requirements (indicated by a green tick) and click "Next".

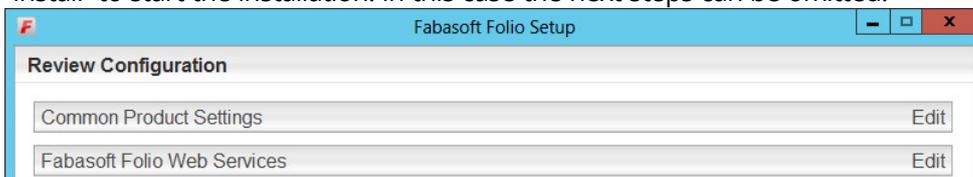


7. Type the *Username* and the *Password* of the user account that should be used for running the Fabasoft Folio Web Service in the corresponding fields and click "Next".



Note: This service user must exist as an operating system user and must have a corresponding user account in the Fabasoft Folio Domain.

8. In the *Review Configuration*, the Fabasoft Folio Services can be configured. If you want to change the settings, click "Edit" next to *Fabasoft Folio Web Services*. Otherwise click "Install" to start the installation. In this case the next steps can be omitted.



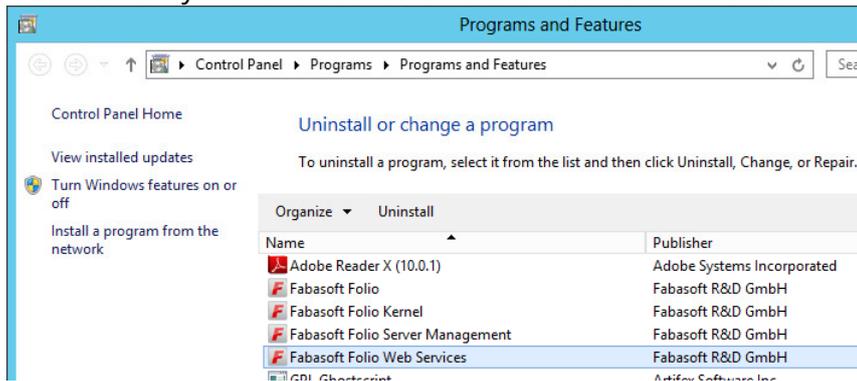
9. The following options are available to configure the Fabasoft Folio Web Services. Click the name of a Fabasoft Folio Web Service, to edit the settings. In the *Virtual Directory* field the name of the virtual directory can be specified.
 - o A Fabasoft Folio Web Service can be edited by clicking the name of the service or by clicking "Edit".
 - o A Fabasoft Folio Web Service can be deleted by clicking "Remove".
 - o A Fabasoft Folio Web Service can be created by clicking the plus icon.
10. Click "OK", to close the dialog for configuring the Fabasoft Folio Web Services.

11. Click "Install" to start the installation.
The Fabasoft Folio Web Services are installed.

6.2 Removing a Fabasoft Folio Web Service

To remove a Fabasoft Folio Web Service installation, perform the following steps:

1. Open the Microsoft Windows "Programs and Features" control panel.
2. Select the entry "Fabasoft Folio Web Services" and click "Uninstall".



7 Installation and Configuration of Fabasoft Folio Conversion Services

7.1 Overview

The Fabasoft Folio Conversion Service is based on the Fabasoft Folio Web Service. The conversion of documents is implemented as an XML web service. Other Fabasoft Folio Web Services may use that conversion service for converting documents.

The Fabasoft Folio Conversion Service hosts third-party software tools required for converting documents. Microsoft Office Word for example is used to convert Microsoft Office Word documents to HTML documents. Usually it is not possible to run Microsoft Office Word within a web service. It is designed as a client application only. Supposing special conditions, it is possible to run such applications within a service (see Microsoft Knowledge Base Article - 257757).

The Fabasoft Folio Conversion Service provides an infrastructure for client applications to be executed in a service environment. For example, the applications are started and invoked in the security context of the Fabasoft Folio Conversion Service account and not in the requesting client's security context. This is necessary for the Microsoft Office applications that rely on a user profile.

Another important aspect is to cache these applications. This results in a significant performance gain. The caching mechanism can be controlled by defining the maximum number of instances and the maximum run time of a pooled application. This allows for avoiding known memory leaks or stability issues. The conversion service also checks whether a pooled application is responding. If not, the application will be restarted automatically.

7.2 Installation Tutorial for Third-party Products

7.2.1 Enable Kerberos Authentication

If the application pool identity of the Fabasoft Folio Conversion Service is configured as a domain user and Integrated Microsoft Windows Authentication should be used, some extra tasks have to

be performed to enable the Kerberos authentication protocol (see Microsoft Knowledge Base Article - 832769):

1. In a command line interface, type the following command (<FQDNConvMachine> is the fully qualified domain name of the Fabasoft Folio Conversion Server and <ConvUser> is the Fabasoft Folio Domain user account):

```
setspn.exe -A HTTP/<FQDNConvMachine><ConvUser>
```

Example:

```
setspn.exe -A HTTP/convserver.fabasoft.com convserver\convuser
```

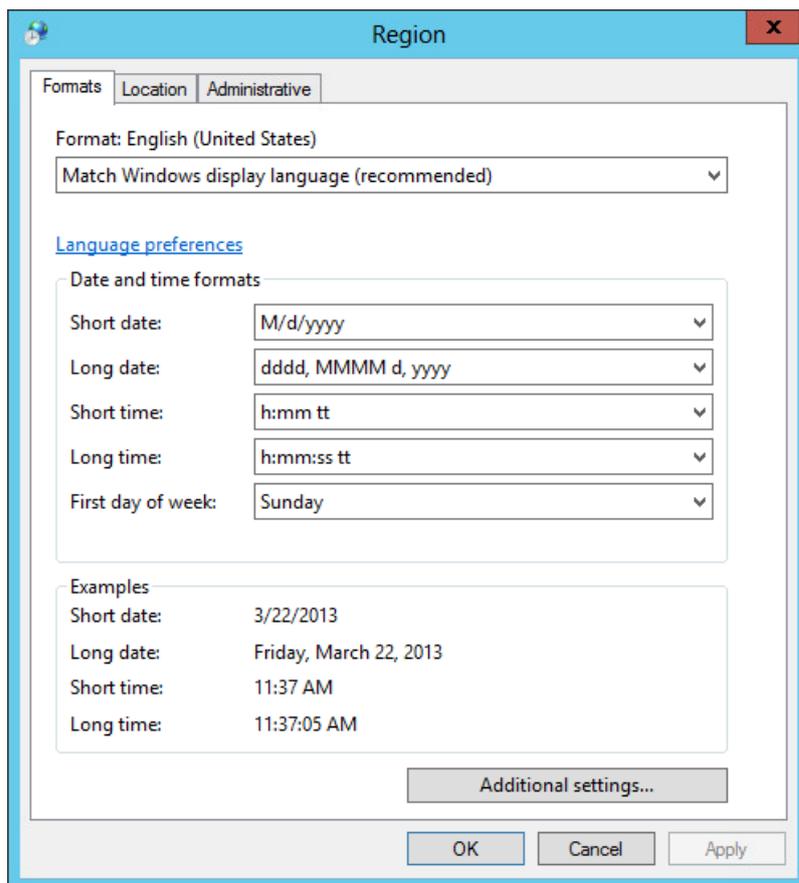
Important: Use the fully qualified domain name of the Fabasoft Folio Conversion Server, not the NetBIOS name. Make sure that the service principal name is unique.

2. Reboot the computer that hosts the Fabasoft Folio Conversion Service (required for enabling the *Trust this computer for delegation* security option).

Note: Use "Basic Authentication" for the Fabasoft Folio Conversion Service to avoid these critical administrative measures. See chapter 7.8.4 "Kerberos Authentication Issues" for details.

7.2.2 Adjust the Regional and Language Options

On the Fabasoft Folio Conversion Server, log on as the user specified in the Fabasoft Folio Web Service setup. Adjust the regional settings as required. This setting will have an effect on how some kinds of information (date, currency etc.) are displayed in the converted documents.



7.2.3 Install and Configure Microsoft Office

To install and configure Microsoft Office for converting documents and to grant launch and access permissions for Microsoft Office applications to the Fabasoft Folio Web Service user account, perform the following steps:

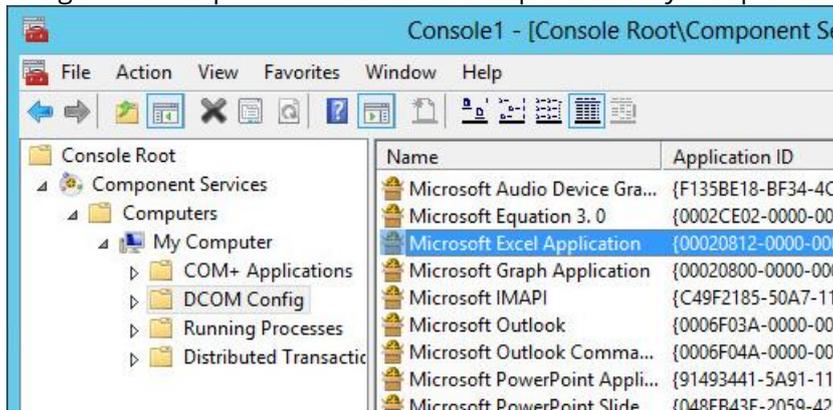
On the Fabasoft Folio Conversion Server, log on as the user specified in the Fabasoft Folio Web Service setup.

Install Microsoft Office

1. Perform a full installation of Microsoft Office. Otherwise Microsoft Office may cause the Microsoft Windows Installer service to pop-up an invisible dialogue box for browsing to the Microsoft Windows Installer package during the conversion of a document. As a result, the conversion is stopped and does not respond.
2. Afterwards start each Microsoft Office application. This initializes the user's profile for the Microsoft Office applications.
3. Disable any automatic checks (e.g. automatic field updates or spelling and grammar checks). This will speed up opening and converting documents.

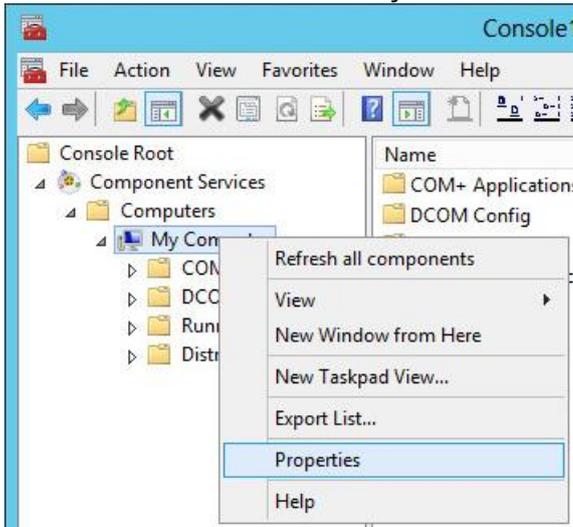
Adjust the DCOM security settings for Microsoft Office applications

1. Open the "Component Services" MMC snap-in.
2. Navigate to "Components Services" > "Computers" > "My Computer" > "DCOM Config".



3. Grant access and launch permissions to the Fabasoft Folio Web Service user for the DCOM applications "Microsoft Word Document", "Microsoft Excel Application" and "Microsoft PowerPoint Presentation".

Note: The default DCOM security is accessible through the properties of the computer node.



7.2.4 Install and Configure Adobe Acrobat Pro

To install and configure Adobe Acrobat Pro for converting documents, perform the following steps:

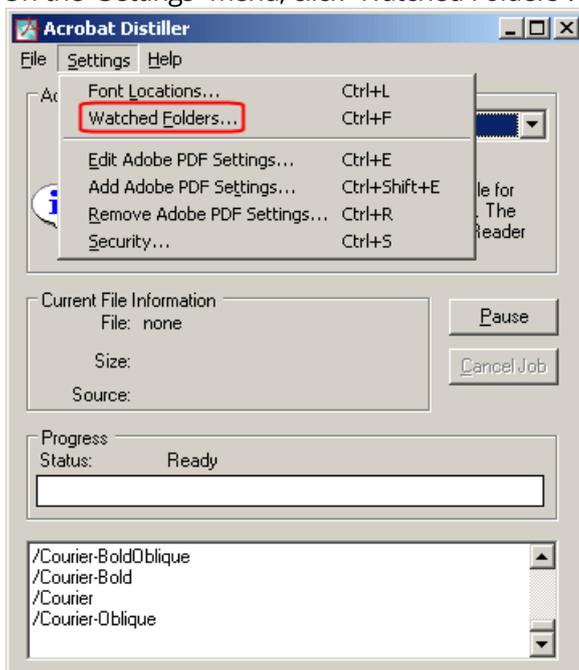
On the Fabasoft Folio Conversion Server log on as the user specified in the Fabasoft Folio Web Service setup.

Install Adobe Acrobat Professional

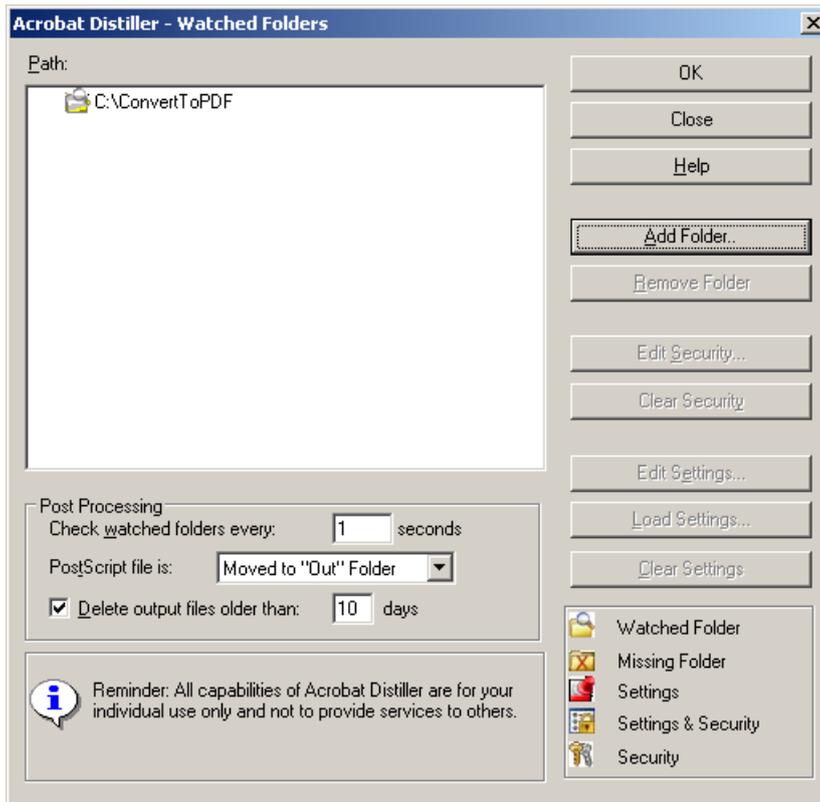
Install Adobe Acrobat Pro by executing the installation file.

Adjust Acrobat Distiller

1. Start the application "Acrobat Distiller".
2. On the "Settings" menu, click "Watched Folders".



3. Click "Add Folder" to add a local folder to the list, change the *Check watched folders every* frequency to "1" second, select the *Delete output files older than* check box and click "OK".



Note: If the client version of Adobe Acrobat Distiller is used, the distiller application must be running on the Fabasoft Folio Conversion Server to generate PDF documents. For that purpose you have to log on interactively to the Fabasoft Folio Conversion Server and start the Adobe Acrobat Distiller application. The server version of Adobe Acrobat Distiller does not require this step.

7.2.5 Install and Configure Ghostscript

To install and configure Ghostscript for converting documents, perform the following steps:

On the Fabasoft Folio Conversion Server log on as the user specified in the Fabasoft Folio Web Service setup.

Install Ghostscript

1. Download and install the supported Ghostscript version. A repository of all available Ghostscript products can be found under the following link:
<https://github.com/ArtifexSoftware/ghostpdl-downloads/releases>
2. Install Ghostscript by executing the installation file.

Configure PDF/A Conversion with Ghostscript

The following steps are only necessary if you want to convert documents to PDF/A file format.

1. Edit the PDF_A_def.ps configuration file (<ghostscript-directory>\lib\PDF_A_def.ps).
2. In the /ICCPfile line replace srgb.icc with the path to the ICC file iccprofiles\srgb.icc in the standard Ghostscript installation path: /ICCPfile <ghostscript-directory>\iccprofiles\srgb.icc

Note: Backslashes in the path must be escaped using a backslash.

7.2.6 Install and Configure FFmpeg

The following steps are necessary if you want to convert video contents.

1. Build or download a static build of FFmpeg: <http://ffmpeg.org/>
2. Move the static FFmpeg tool to a location which is referenced by the PATH environment variable.

7.2.7 Install and Configure ExifTool

The following steps are necessary if image properties should be available.

1. Download ExifTool from <http://search.cpan.org/dist/Image-ExifTool/>.
2. Extract the file `exiftool` and the directory `lib` to the directory where you want to install ExifTool.
3. Rename `exiftool` to `exiftool.pl`.
4. Add this directory to the `PATH` environment variable.

The following step is only required if Perl is not already installed on your system:

1. Download and install StrawberryPerl from <http://strawberryperl.com/>.

7.2.8 Install and Configure LibreOffice

To install and configure LibreOffice for converting documents, perform the following steps:

On the Fabasoft Folio Conversion Server log on as the user specified in the Fabasoft Folio Web Service setup.

Install LibreOffice

1. Download the supported LibreOffice version from the URL <http://www.libreoffice.org>.
2. Extract the downloaded ZIP archive and install LibreOffice by executing the installation file. During the setup, select the following options:
 - o In the "File Type" dialogue, clear all check boxes to avoid that Microsoft Office documents are automatically opened with LibreOffice. If they are opened with LibreOffice automatically, they are not converted.

Setting the environment variable

After finishing the installation, the LibreOffice installation path has to be specified in the environment variable `PATH`, otherwise the conversion fails.

Example: `PATH=<existing path>;<libreoffice-directory>\program`

Setting the registry key

In order that conversion works, after rebooting the computer the following registry key has to be set:

```
[HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run]
REG_SZ LibreOfficeConversionKey = <libreoffice-directory>\program\soffice.exe -
headless -nofirststartwizard -norestore -invisible -
accept=socket,host=localhost,port=8100;urp
```

7.2.9 Start Conversion Tools

Adobe Acrobat Distiller and LibreOffice have to be started explicitly on the Fabasoft Folio Conversion Server. All other conversion tools are started automatically.

7.2.9.1 Starting Adobe Acrobat Distiller

Execute the `acrobat.exe` file. Then Adobe Acrobat Distiller is polling the configured “watched folder”.

7.2.9.2 Starting LibreOffice

Verify that LibreOffice is not already running (`soffice.exe` process in the process manager). In a command line interface, switch to the LibreOffice installation directory and enter the following command:

```
soffice -headless -accept=socket,host=localhost,port=8100;urp
```

Thus LibreOffice is running as a service and is able to receive requests on IP port 8100. Several instances LibreOffice may be started, each of them listening on a unique IP port.

Note: Be sure to use the same port numbers as configured in the Tools Host configuration (see chapter 7.4 “Configuring a Fabasoft Folio Conversion Service”).

7.2.10 User Settings

The user account which is used to run Fabasoft Folio Conversion Services must be a member of the group “Print Operators”.

7.3 Installation of Fabasoft Folio Conversion Services

For software requirements and prerequisites see chapter 3.4 “Requirements for Installing Fabasoft Folio Conversion Services” and chapter 4.4 “Prerequisites for Fabasoft Folio Conversion Services”.

Note: The machine that hosts the Fabasoft Folio Conversion Service should be a machine that is dedicated for conversion purposes only. If this installation is also used as a regular Fabasoft Folio Web Service, serious problems may be encountered (see chapter 7.8.5 “Simultaneous Use as a Regular Web Service” for details).

To install Fabasoft Folio Conversion Services via the automated setup, perform the following steps.

1. Insert the Fabasoft product DVD in the DVD drive of your server. The Fabasoft Server Setup dialog will be displayed automatically.
Note: If this dialog is not displayed automatically, run `setup.exe` in the root directory of the Fabasoft product DVD.
2. The installation works as described in chapter 6.1 “Installation of Fabasoft Folio Web Services”. Only instead of selecting to install *Fabasoft Folio Web Services* select the *Fabasoft Folio Conversion Services* check box.

7.4 Configuring a Fabasoft Folio Conversion Service

Open a Fabasoft Folio Web Client, navigate to the *Domain Administration* and click the “Configuration Objects” tab. There are three configuration objects that contain configuration data for the Fabasoft Folio Conversion Service:

- *Tools Host Configuration* (FSCTLH@1.1001:StdOleAutomationHost)
This object manages hosting of external applications used for converting documents.
- *Conversion Configuration* (FSCCONV@1.1001:ExternalConversionConfiguration)
This object is the main configuration object.
- *Default Web Services* (FSCCONV@1.1001:DefaultWebServices)
This object defines the location of the Fabasoft Folio Conversion Services.

7.4.1 Tools Host Configuration

The *Tools Host* software component manages conversion tools that support the Microsoft OLE Automation interface. It provides the infrastructure for client applications to run in a server environment. The default settings should be sufficient in most cases.

7.4.1.1 Application Configuration

The FSCTLH@1.1001:StdOleAutomationHost configuration object lists the available conversion tools. The following applications are configured by default:

- Microsoft Office Word
- Microsoft Office Excel
- Microsoft Office PowerPoint
- Microsoft Office Outlook

The *Tool Configuration* compound property specifies parameters used for managing the configured application:

Field	Description
<i>OLE Automation Server</i>	Specifies the ProgID or ClassID of the OLE Automation server (e.g. "Word.Application").
<i>Test</i>	Specifies an OLE Automation Dispatch Identifier (DISPID) of a test function. This function will be called when checking if the application is still alive. Usually this is a function that reads a global property, for example the version number of the application.
<i>Quit</i>	Specifies an OLE Automation Dispatch Identifier (DISPID) of a function that terminates the application.
<i>Restart after (Calls)</i>	Specifies the maximum number of uses of the application. The current instance will exit and a new instance of the application will be created when exceeding this limit.
<i>Maximum life time (in seconds)</i>	Specifies the maximum lifetime of an application. The current instance will exit and a new instance of the application will be created when this limit is exceeded.
<i>Monitoring time interval (in seconds)</i>	If the tool has a separate process, then this process can be monitored. This attribute defines a time interval for calculating average CPU utilization.

<i>Maximum CPU utilization (in percent)</i>	The maximum CPU utilization of the process associated with the tool. If the monitored value exceeds this configured value then the process will be terminated.
<i>Wait time for graceful process termination (in seconds)</i>	The associated process of a tool should terminate automatically when it is no longer in use. To avoid hanging tool processes specify a time interval within which the process can terminate automatically. If the specified amount of time is exceeded, the process will be terminated.
<i>Software Component</i>	This field is used when adding new applications to the <i>Tools Host</i> configuration object. Specify a software component if the settings should be kept after updating.

To ensure optimal stability, the reuse of instances is deactivated by default. Therefore, no applications are cached after they have been used in conversion operations. To speed up the conversion significantly the *Restart after (Calls)* value has to be increased. The *Tools Host* invokes the specified test function to determine if the application is still alive and responding. The *Restart after (Calls)* and *Maximum life time (in seconds)* fields are maximum thresholds for application usage. This helps to avoid known memory leaks and stability problems.

7.4.1.2 Detection of Application Crashes

Sometimes, a process of a conversion tool crashes and is caught in an endless-loop. The *Tools Host* component is able to handle such situations.

The *Monitoring time interval (in seconds)* and *Maximum CPU utilization (in percent)* settings of the *Tool Configuration* compound property define additional conditions for tools that have separate processes. If a process exceeds the configured threshold value, it will be terminated and a Microsoft Windows event log is written to denote failure of the tool.

The average CPU utilization of the conversion tool processes also can be traced. If the `HKEY_LOCAL_MACHINE\SOFTWARE\Fabasoft\FSTLH@1.1001` registry key contains the named `REG_DWORD`-value `TraceWatchedProcesses` and its value is set to "1", then the currently used conversion processes are traced to the Fabasoft DUCX Tracer. This may be useful for determining an appropriate threshold value for the *Maximum CPU utilization (in percent)* for a specific conversion tool.

7.4.1.3 Detection of Blocking Windows

A conversion tool may not be able to continue when it is displaying a modal dialogue box. The configuration object `FSTLH@1.1001:StdOleAutomationHost` provides a list of window descriptions of potential blocking windows (`FSTLH@1.1001:blockingwindows` property). The boxes of the aggregate *Blocking windows* describe a window:

Field	Description
<i>Window title</i>	Title of the window
<i>Window class</i>	Class name of the window

<i>Window style</i>	Style flags of the window
<i>Window process</i>	File name of the window process
<i>Software Component</i>	This field is used when adding new entries to this compound property. Specify a software component if the settings should be kept after updating.

All windows in the invisible web service desktop are compared to the values specified in the fields of this compound property. The fields are linked together using a logical AND operator. If a field is omitted it will not be taken into account when detecting a window.

The detection of potential blocking dialogue boxes is done at regular intervals. The *Polling frequency for detecting blocking windows (in seconds)* property (FSCTLH@1.1001:blockingwindowspollinginterval) specifies this interval. In this field a value greater than zero must be typed in order to take effect, otherwise no detection will occur.

A Microsoft Windows event log will be generated if a blocking dialogue box has been detected and closed. The event log contains the window title, window class, style flags, window process, and a list of the controls within this window so that the reason for the failed conversion can be identified.

The tools host state report contains a list of currently open windows (see chapter 7.4.5 “Tools Host State”). This is very helpful when encountering a hanging conversion operation and suspecting a blocking dialogue box. The window descriptions written to the state report may be taken to the *Blocking windows* aggregate of the *Tools Host* configuration object.

Be careful when extending the *Blocking windows* compound property. It may lead to unexpected side effects if an added entry identifies a normal window instead of a blocking dialogue box.

7.4.1.4 Conversion Timeout

The maximal conversion duration can be configured via the property *Conversion Timeout (in Seconds)* (FSCTLH@1.1001:conversiontimeout). The limit will be enforced using Microsoft Office or LibreOffice as external conversion tool. If the timeout is expired the conversion will be cancelled.

7.4.2 Conversion Configuration

The configuration object FSConv@1.1001:ExternalConversionConfiguration contains the supported document formats and additional parameters needed for the conversion operations.

The supported document format conversions are described in the *Conversion table* field. One line in this compound property defines a single conversion step. It consists of the following fields:

Field	Optional	Description
<i>Source Type</i>	No	Source types separated by a semicolon.
<i>Destination Type</i>	No	Destination types separated by a semicolon.
<i>Priority</i>	No	Priority of the current aggregate line. It is possible to have different implementations for the same document format. In this case the row with the highest priority (lowest number) will be selected.

<i>Conversion Action</i>	No	The conversion request will be passed to the specified action.
<i>Check conversion action</i>	Yes	Validation action for checking the conversion operation. Checks the availability of required tools and configuration.
<i>Application ProgID</i>	Yes	ProgID or ClassID of an application that supports the Microsoft OLE Automation interface (IDispatch).
<i>Printer</i>	Yes	Name of the printer when using printer-based conversions (e.g. PDF conversion).
<i>Web Service</i>	Yes	Specifies the web service end-point(s) of remote conversion services. If this box is empty or in the specified web service object no URL is specified then the conversion takes place locally.
<i>Software Component</i>	Yes	Helps to preserve configured values beyond updates.

A conversion can also be “multi-staged”. The shortest way from the source type to the destination type will be found automatically using Dijkstra’s “single source - shortest path” algorithm.

Example: A Microsoft Office Word document should be converted to a PDF document and the conversion configuration has the following settings:

Source Type	Destination Type	Priority	Action
doc	pdf	100	Use activePDF for PDF generation.
doc	ps	40	Use Microsoft Office Word and the Adobe Acrobat Distiller printer driver to create a postscript file.
ps	pdf	40	Use Adobe Acrobat Distiller to convert a postscript file to PDF.
ps	pdf	35	Use Ghostscript to convert a postscript file to PDF.

In this example the conversion from Microsoft Office Word format (“doc”) to PDF format is done in two steps (doc →ps, ps→pdf), because it is less expensive than using the direct conversion (40 + 35 < 100).

The configuration object `FSCCONV@1.1001:ExternalConversionConfiguration` is already populated with default values for HTML and PDF conversions. Ghostscript is the default application for creating PDF documents.

The *Content Converter* component is also able to do conversions by executing command lines. The conversion configuration can be extended by adding the `FSCCONV@1.1001:CmdLineConvert` action. The printer field is used to configure the command line. It contains place holders for input and output files.

Example: “fop.bat -opt <~src~><~dest~>”

7.4.3 Default Web Services Configuration

The configuration object *Default Web Services* (FSCCONV@1.1001:DefaultWebServices) stores end-points for remote web services. By default, all conversion formats that require a remote web service reference the *Default Web Services* object.

Create new web service objects that contain other end-points. By assigning different web service objects to conversion formats in the main conversion configuration, server affinity can be achieved: certain document formats will be converted on dedicated conversion servers.

The *Default Web Services* object and other instances of the FSCOWS@1.1001:HttpConnectors object class contain connection information for remote web services. Any number of end-points can be specified in such objects.

Field	Optional	Description
<i>URL</i>	No	Specifies the URL of the remote conversion service. The syntax is <code>http://<webserver>/<vdir></code> .
<i>User</i>	Yes	Basic Authentication only: user name
<i>Password</i>	Yes	Basic Authentication only: user password
<i>Proxy User</i>	Yes	Basic Authentication only: user name for proxy authentication
<i>Proxy Password</i>	Yes	Basic Authentication only: user password for proxy
<i>Client Certificate</i>	Yes	Client certificate for HTTPS connections (SSL). The certificate must be located in the local machine store.
<i>Software Component</i>	Yes	Helps to preserve configured values after updating.

If no values are defined in this compound property conversion takes place on the local kernel. In most cases this is no appropriate solution. The conversion tools would be activated and invoked on the local machine using the security context of the client user. This is not applicable for *Fabasoft Folio/Web* with a potentially high number of Fabasoft Folio Web Clients. As a result, the conversion tools (e.g. Microsoft Office) would have to be installed and configured for each client user on the Fabasoft Folio Web Server.

If one or more entries are specified in the *Default Web Services* configuration object then the conversion operations will be executed on these web services. The conversion tools are invoked in the security context of the web service user of the remote web service. This is the user account that has been specified during the installation of the remote conversion service.

If there are multiple entries in the *Default Web Services* configuration object then a simple load-balancing algorithm is used to select an appropriate end-point. This algorithm keeps records of response times and throughput for each conversion service. This statistical information forms the basis for choosing a web service end-point on a subsequent conversion request.

7.4.4 Logging Mode

All essential input and output parameters of a conversion operation can be logged to the file system. Two kinds of logging modes are available: the conversion service logging mode and the web service logging mode.

7.4.4.1 Conversion Service Logging

The conversion service logging mode can be enabled by creating the following registry key:

```
HKEY_LOCAL_MACHINE\Software\Fabasoft\FSCCONV@1.1001
```

Furthermore, the named values `LogDetailFolder` and `LogDetailCount` need to be created within this key. The `REG_SZ` value `LogDetailFolder` specifies an existing directory on the file system for storing the logging information. The `REG_DWORD` value `LogDetailCount` specifies the maximum number of logging entries to be made. This value is decremented each time a conversion operation is logged. If the counter reaches zero, logging will be disabled.

Example:

```
[HKEY_LOCAL_MACHINE\Software\Fabasoft\FSCCONV@1.1001]
REG_SZ LogDetailFolder = "C:\Temp\fsconconv"
REG_DWORD LogDetailCount = 10
```

7.4.4.2 Web Service Logging

The web service logging mode is similar to the conversion service logging mode. It can be enabled by creating the following registry key:

```
[HKEY_LOCAL_MACHINE\Software\Fabasoft\FSCOWS@1.1001]
REG_SZ LogDetailFolder = <destinationfolder>
REG_DWORD LogDetailCount = <maximum number of logs>
```

Note: The Fabasoft Folio Conversion Service is implemented as a web service. Hence, each conversion corresponds to a SOAP request.

7.4.5 Tools Host State

State and statistical information about the tools host on the conversion server can be queried using the `FSCTLH@1.1001:GetStateReport` action. Such a report is available in an XML or HTML format and contains the following information:

- System information
Hardware and software inventory, list of installed conversion tools.
- Cached tools
List of conversion tools with current state.
- Statistics
Runtime statistics of the conversion operations.
- Warnings and errors
List of warnings and errors occurred so far.
- Detailed history
List of all events occurred so far.

- Open windows

List of open windows and dialogue boxes owned by the conversion tools, very useful for detecting blocking dialogue boxes.

Call the `FSCCTLH@1.1001:GetStateReport` action via a HTTP/GET request using a web browser. The URL would be as follows:

`http://<webserver>/<vdir>/tlhstatus`

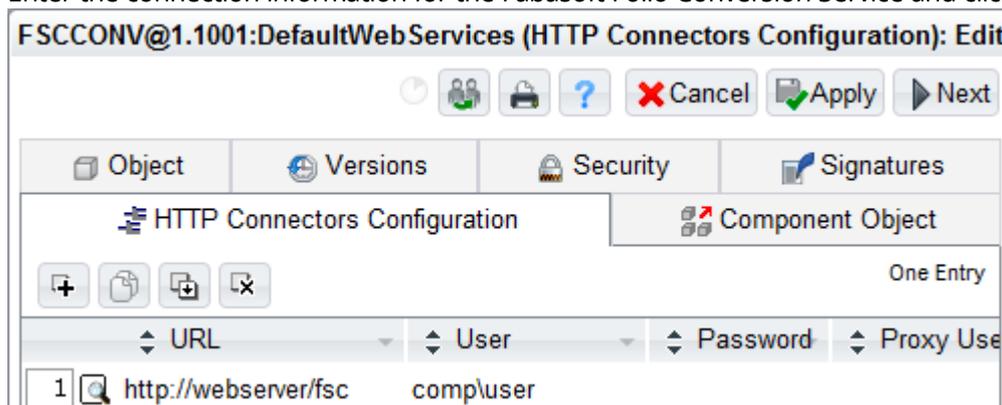
7.4.6 Necessary Configuration Steps

The Fabasoft Folio Conversion Service has to be configured according to the system and the conversion tools used.

7.4.6.1 Conversion Server URL

To specify the connection information for the Fabasoft Folio Conversion Service, perform the following steps:

1. Right-click the `FSCCONV@1.1001:DefaultWebServices` and click "Properties".
Note: If the object is opened in read-only mode, click "Edit".
2. Enter the connection information for the Fabasoft Folio Conversion Service and click "Next".

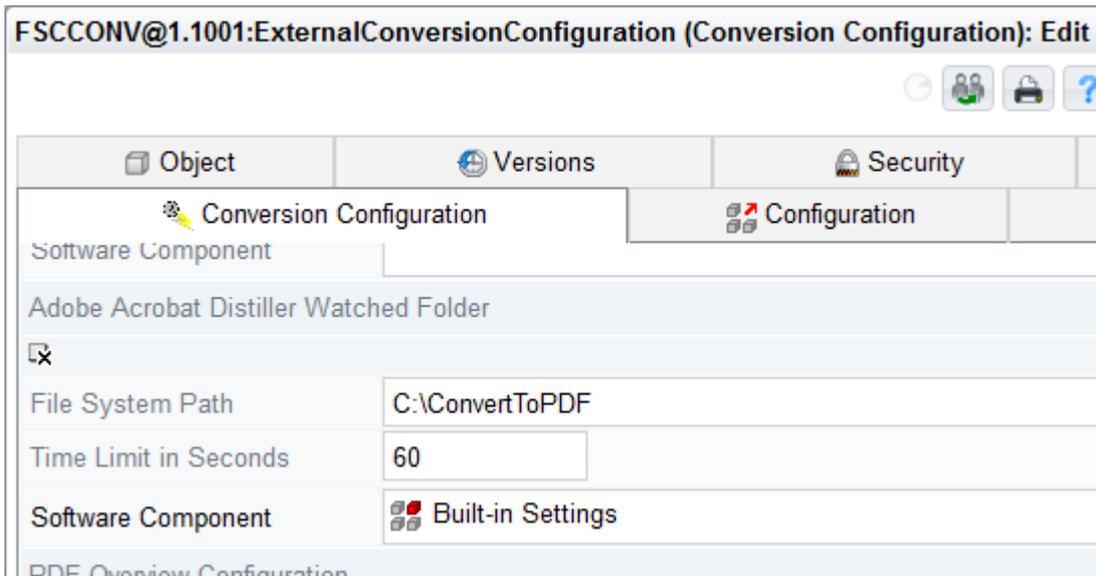


7.4.6.2 Adobe Acrobat Watched Folder

If Adobe Acrobat Distiller is used as PDF generator, a watched folder has to be configured. Perform the following steps:

1. Right-click `FSCCONV@1.1001:ExternalConversionConfiguration` and click "Properties".
Note: If the object is opened in read-only mode, click "Edit".
2. On the "Conversion Configuration" tab under *Acrobat Distiller watched folder* specify the "watched folder", which has already been configured during the configuration of Adobe Acrobat

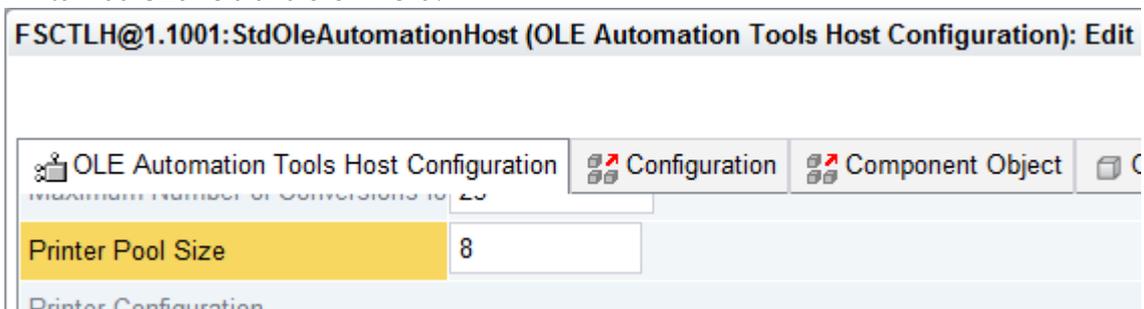
Distiller and click "Next".



7.4.6.3 Printer Pool Size

The pre-defined size of the printer pool is 8, which should be sufficient for most cases. To change this setting, perform the following steps:

1. Right-click `FSCTLH@1.1001:StdOleAutomationHost` and click "Properties".
Note: If the object is opened in read-only mode, click "Edit".
2. On the "Ole Automation Tools Host Configuration" tab type a size for the printer pool in the *Printer Pool Size* field and click "Next".

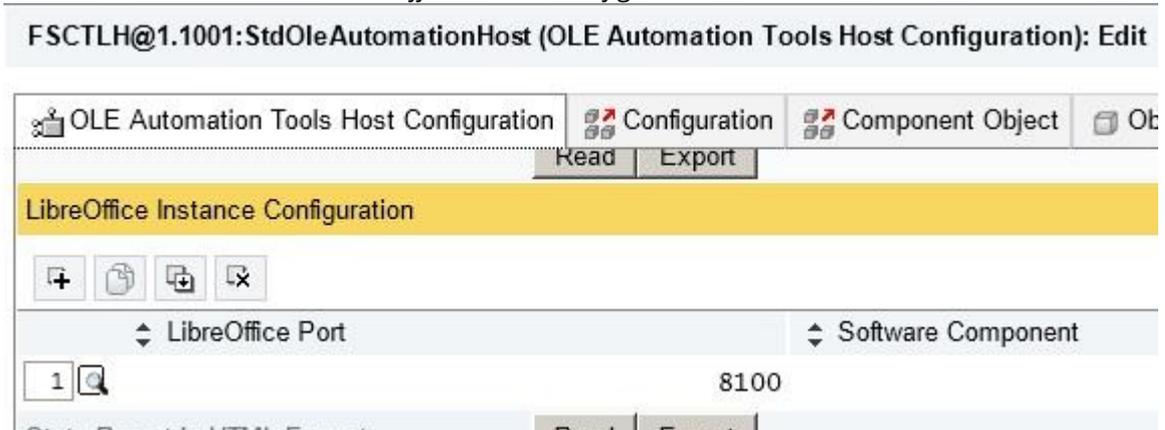


7.4.6.4 LibreOffice Instances

If LibreOffice is used as PDF generator, perform the following steps to customize the `FSCTLH@1.1001:StdOleAutomationHost` configuration object:

1. Right-click `FSCTLH@1.1001:StdOleAutomationHost` and click "Properties".
Note: If the object is opened in read-only mode, click "Edit".

- On the “Ole Automation Tools Host Configuration” tab define the listening ports of the LibreOffice instances in the *LibreOffice instance configuration* field and click “Next”.



7.4.7 Additional Configuration Options

Additional conversion options can be specified in the registry path

HKEY_LOCAL_MACHINE\Software\Fabasoft\FSCCONV@1.1001:

- PreferMSOfficePDF/A:** A value greater than 0 specifies that a PDF/A document will be created when converting via Microsoft Office, even though the target format is “PDF”. By default, this option is not enabled.
- PreferOpenOfficePDF/A:** A value greater than 0 specifies that a PDF/A document will be created when converting via LibreOffice application, even though when the target format is “PDF”. By default, this option is not enabled.
- DisableWordFillIn:** A value greater than 0 disables all Fill-In fields when converting a Microsoft Word document. Enabling this option has a performance impact. By default, this option is not enabled.
- UpdateFieldsAtPrint:** If the value is 1 fields in the document will be updated, if the value is 0 the fields will not be updated.

7.5 Web Service Interface

The Fabasoft Folio Conversion Service is implemented as a web service. Fabasoft Folio Web Services use this web service for submitting conversion requests.

The web service definition `FSCCONV@1.1001:ConversionService` describes a SOAP service that contains the following web methods:

- `FSCCONV@1.1001:SOAPConvert`
Executes a conversion action on the target machine.
- `FSCCONV@1.1001:SOAPCheckConvert`
Checks whether a conversion operation can be done on the target machine.
- `FSCTLH@1.1001:GetStateReport`
Returns a report in HTML or XML format that contains status information and statistics about the tools host on the target machine.

Other (external) applications may also use this web service. The WSDL definition for the conversion service is available via the following URL:

`http://<webserver>/<vdir>/fscdav/wsd1?WEBSVC=FSCCONV_1_1001_ConversionService`

7.6 Security Requirements

7.6.1 DCOM Security

The Fabasoft Folio Conversion Service user account starts external applications using the DCOM interface. Ensure that this user account has permissions to launch the DCOM servers of the conversion tools. Start the `dcomcnfg.exe` configuration program to modify the security of the DCOM servers. Grant launch permissions for each individual application or for all applications by adjusting the default launch permissions. The initial default launch security includes the following users and groups:

- Local administrators
- The interactively logged-on user
- Local system account (operating system account)
- The anonymous accounts of the Internet Information Server: `IUSR_machine`, `IWAM_machine`

Be careful to check the permissions of each conversion application. If custom permissions are selected, the default launch permissions will have no effect.

A set of conversion applications is already configured in the *Tools Host* configuration object (`FSCTLH@1.1001:StdOleAutomationHost`). Launch permissions for the following applications must be granted to the Fabasoft Folio Conversion Service user in order to be used for conversion operations:

Name or AppID	ProgID of COM Server
Microsoft Excel Application	Excel.Application
Microsoft Word Document	Word.Application
Microsoft PowerPoint Presentation	PowerPoint.Application
Outlook Message Attachment	Outlook.Application
{000C101C-0000-0000-C000-000000000046}	IMsiServer

Note:

- DCOM servers are COM servers associated with so-called AppIDs, where additional configuration settings like security, identity and other settings are stored.
- The DCOM server for the MSI Installer is used by Microsoft Office applications and is launched indirectly by the Fabasoft Folio Conversion Service user account.
- The Fabasoft Folio Web Service setup program does not currently change the DCOM security. It is the administrator's task to grant launch permissions.

7.6.2 File System Security Requirements

The Fabasoft Folio Conversion Service must be able to create temporary files. The directory for these temporary files was configured during setup. In order to be able to create temporary files, the Fabasoft Folio Conversion Service user account must have full access to this directory.

Each Fabasoft Folio Web Service instance contains a directory for static web content. This directory will be created and synchronized when the web service starts. Therefore the Fabasoft Folio Web Service user account needs full access to this directory. The default location of this directory is `<Program Files>/Fabasoft/Components/Web/<vdir>/ASP/content/tmp`, where the first part of the path (till `<vdir>`) specifies the physical location of the installed virtual directory.

Note: The Fabasoft Folio Web Service setup program adjusts the security on these directories.

7.6.3 Security Requirements of Microsoft Office

The Microsoft Office applications are installed either per user or per machine, depending on the privilege level of the user who installs or accesses these applications. If an administrative user starts a Microsoft Office application then the installation configuration is managed per machine. Otherwise, the installation configuration is stored per user in the user profile. Install Microsoft Office as an administrator of the local machine to ensure these applications are available for all users of the machine.

As most client applications, Microsoft Office relies on a user profile. Hence, the Fabasoft Folio Conversion Service loads a user profile before it uses these applications. This operation requires administrative privileges. For this reason the Fabasoft Folio Conversion Service user account must be a member of the local administrator group.

Note: The conversion is done using the macro security settings of the user who runs the Fabasoft Folio Conversion Service.

7.6.4 Fabasoft Folio Security Requirements

Final form views of file objects are created using templates, which contain Fabasoft DUCX Expressions for the content parts to be assembled. The evaluation operations are invoked in the security context of the configured Fabasoft Folio Conversion Service account. Thus this user account must have the appropriate permissions to read attributes and contents referenced by these Fabasoft DUCX Expressions.

It is highly recommended that the Fabasoft DUCX Expressions in the final form template are designed independently of the permissions of the Fabasoft Folio Conversion Service user account. Pass a "global scope dictionary" to the `FSCCONV@1.1001:Convert` action. This dictionary may contain any user related objects and attributes referenced by Fabasoft DUCX Expressions evaluated by the Fabasoft Folio Conversion Service.

7.7 Licensing Issues

Each client user that utilizes the Fabasoft Folio Conversion Service indirectly uses the conversion tools (Microsoft Office, Adobe Acrobat Distiller, activePDF). The license requirements for the conversion tools are as follows:

- Each client user requires a license for the Microsoft Office applications. Currently, there is no server-licensing model available for this product.
- The Adobe Acrobat Distiller provides both, client and server licensing.
- activePDF Server is licensed per server and is available as Professional version with an optional "Thread Upgrade" package, which improves the performance.

7.8 Troubleshooting

If problems occur when converting documents, take a look at the Microsoft Windows Event Viewer on both, the Fabasoft Folio Web Server and the Fabasoft Folio Conversion Server. In many cases the source of the problem can be found by viewing these log data.

7.8.1 Changing the Password

If the password for the web service user is changed, also the identity of the web service application pool has to be reconfigured. This can be done in the "Internet Services Manager" ("Start" > "Programs" > "Administrative Tools" > "Internet Services Manager").

7.8.2 Changing the Web Service Identity

For the Fabasoft Folio Web Service user account some security settings are required. If the identity of the Fabasoft Folio Web Service is changed, consider the following:

- The new Fabasoft Folio Web Service user account must be able to log on to the Fabasoft Folio Domain. Furthermore a user environment has to be specified for this user account.
- The new Fabasoft Folio Web Service user account must be a member of the local administrator group.
- The new Fabasoft Folio Web Service user account must be a member of the local group `IIS_WPG`.
- If the new Fabasoft Folio Web Service application pool identity is a domain user account, enable Kerberos authentication. See chapter 7.2.1 "Enable Kerberos Authentication".
- Adjust the DCOM security so that the new user account is able to start and execute the required Microsoft Office applications. See chapter 7.2.3 "Install and Configure Microsoft Office".
- Make an interactive logon with the new user account on the Fabasoft Folio Conversion Server and start each Microsoft Office application. This step creates a profile and initializes the Microsoft Office applications for the new user account.
- The Fabasoft Folio Web Service caches static content in the `<Program Files>/Fabasoft/Components/Web/<vdir>/ASP/content/tmp` directory. The new user account must have full file access permissions on this file system directory in order to synchronize the cached content during web service start-up.

7.8.3 HTTP Client Issues

The Fabasoft Folio Web Client does not communicate directly with the Fabasoft Folio Conversion Server. Server/server communication is implemented.

The XMLHTTP object of Microsoft XML (based on the Microsoft WinINet API) is used for client/server communications. All internet settings (security, proxy etc.) apply for this mode. However, WinINet is not server-safe. Hence, the ServerXMLHTTP object of Microsoft XML (based on Microsoft WinHTTP) is used for server/server communication.

The Microsoft WinHTTP client HTTP stack does not depend on the WinINet settings. The `proxycfg.exe` configuration tool is used to set up WinHTTP. For more details see the MSDN library article "How To Use ProxyCfg.exe".

The configuration object *Default Web Services* contains time out thresholds among other connection properties. Change these values, if the requirements differ from the default settings.

- *Resolve Time Out (in sec)* (default: unlimited)
In this field, a maximum time for resolving the DNS name can be defined.
- *Connect Time Out (in sec)* (default: 60 seconds)
In this field, a maximum time for connecting to the target web server can be defined.
- *Send Time Out (in sec)* (default: 5 minutes)
In this field, a maximum time for sending the request to the target web server can be defined.
- *Receive Time Out (in sec)* (default: 60 minutes)
In this field, a maximum time for receiving the response from the target web server can be defined.

7.8.3.1 Microsoft WinHTTP 5.0 and WinHTTP 5.1

The Fabasoft Folio Web Service uses Microsoft WinHTTP to communicate with the Fabasoft Folio Conversion Service. The supported versions are Microsoft WinHTTP 5.0 (*winhttp5.dll*) and Microsoft WinHTTP 5.1 (*winhttp.dll*), whereas version 5.1 is preferred.

Loading the operating system component Microsoft WinHTTP 5.0 will fail if more than 63 thread-local-storage (TLS) slots have already been allocated. The "Error in the DLL (0x800401F9)" error message is displayed. Typically this error occurs when the Fabasoft Folio Web Service tries to connect to the conversion service after it has been running for a long time while doing certain kinds of operations. This issue has been fixed in Microsoft WinHTTP 5.1.

7.8.4 Kerberos Authentication Issues

By default, Kerberos authentication cannot be used if the application pool identity of the Fabasoft Folio Conversion Service is configured as domain user. Contacting a Fabasoft Folio Conversion Service would result in a HTTP 401 status code. To solve or work around the problem, the following options are available.

7.8.4.1 Register a Service Principle Name (Recommended)

Register a Service Principle Name (SPN) for the domain user account that the application pool is running as. This is the proposed solution according to Microsoft Knowledge Base Article 832769. See chapter 7.2.1 "Enable Kerberos Authentication" for details.

On Microsoft Internet Information Service 7.0 following command needs to be executed for each application pool:

```
c:\Windows\System32\inetsrv>appcmd set config "Default Web Site/fsc" -  
section:windowsauthentication /useAppPoolCredentials:true
```

7.8.4.2 Disable Integrated Microsoft Windows Authentication

In the "*Internet Services Manager*" MMC snap-in, disable Integrated Microsoft Windows Authentication for the virtual directory of the Fabasoft Folio Conversion Service.

Disadvantage:

- Basic Authentication is not safe.

7.8.4.3 IP address forces Basic Authentication

If a TCP/IP address is configured instead of a NetBIOS name in the connection URL (in the *Default Web Services* configuration object), the HTTP XMLHTTP/ServerXMLHTTP client implementations will use Basic Authentication.

Disadvantages:

- Basic Authentication is not safe.
- Names are better than IP addresses because IP addresses change more frequently than names.

7.8.4.4 Local user account as application pool identity

Configure the application pool identity of the Fabasoft Folio Conversion Service as a local user account.

Disadvantage:

- Local user accounts have to be maintained, which results in a significant increase of administrative effort.

7.8.4.5 Connect with a local user account

Configure a local user account in the *Default Web Services* configuration object.

Disadvantages:

- Local user accounts have to be maintained, which results in a significant increase of administrative effort.
- Load balancing using a content switch will not work, because the selected target server may not know the supplied user account.

7.8.5 Simultaneous Use as a Regular Web Service

The usage of a Fabasoft Folio Conversion Service as a regular Fabasoft Folio Web Service may lead to a problem. Reading a certain attribute may trigger a "Get Action" that initiates a conversion request. This request would cause a web service round trip to the local Fabasoft Folio Kernel instance, which results in a dead lock.

Hence, it is recommended to install the Fabasoft Folio Conversion Service on a dedicated machine.

7.8.6 Transaction Context

Take into account that transactions cannot be shared between multiple Fabasoft Folio Kernel processes. Objects created in the current transaction do not exist unless the transaction is committed. This may impact the evaluation of Fabasoft DUCX Expressions on the Fabasoft Folio Conversion Server. In this case ensure that the Fabasoft DUCX Expressions within the source template do not depend on the current transaction context.

7.8.7 Enable the Logging Mode

If the conversion operation returns incorrect documents, enable the conversion logging mode (see chapter 7.4.4.1 “Conversion Service Logging”). The documents stored into the logging directory are very valuable for analyzing the problem.

7.8.8 Obtaining the Tools Host State

If conversion fails narrow down the error cause via the tools host status report. It contains occurred errors, pending conversions, statistics, and other information. The tools host status report can be obtained by navigating to the following URL:

```
http://<webserver>/<vdir>/tlhstatus
```

7.8.9 LibreOffice Issues

Be sure to specify the installation and URE (UNO Runtime Environment) path of LibreOffice in the environment variable `PATH`. Otherwise conversion fails and the following error message is displayed: “fscconv.dll not found”.

7.8.10 IIS Time Out Issues

The standard value for the server-side connection timeout of Microsoft Internet Information Services is set to 120 seconds. This means that if a conversion takes longer than two minutes, the Fabasoft Folio Web Server will cancel the request. Increase this value if longer conversion times are expected.

7.8.11 Microsoft Office Issues

Be sure that Microsoft Excel can use the printers of the Fabasoft Folio Conversion Service (Fabasoft-CONV-<no>) when running Microsoft Excel in the context of the Fabasoft Folio Conversion Service user. If the printers are not available in Microsoft Excel specify explicit security permissions for the Fabasoft Folio Conversion Service user on the security tab of the printers.

Note that the number of Microsoft Office processes, which can be started and used by the Fabasoft Folio Conversion Service is limited and depends on the system configuration.

7.9 References

- Microsoft Knowledge Base Article Q257757: “INFO: Considerations for Server-Side-Automation of Office”, <http://support.microsoft.com/default.aspx?kbid=257757>
- MSDN Library: “How To Use ProxyCfg.exe”, <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/winhttp/http/proxycfg.asp>
- ActivePDF online documentation for ActivePDF Server, <http://www.activepdf.com/en/Documentation/activePDF-server.pdf>
- Google Groups posting: ASP error 0x800401F9: <http://groups.google.com/groups?threadm=OyTcSQudCHA.620%40tkmsftngp12&group=microsoft.public.winhttp>

8 Installation and Configuration of Fabasoft Folio AT Services

8.1 Preparing Operations

Each user who wants to process tasks in the background needs an own Fabasoft Folio AT Service.

Note: For processing background activities and waiting actions in the workflow (*Fabasoft Folio/Wf* software component) one Fabasoft Folio AT Service can be used for all users.

Excursion: Changing the User Context

An action is processed in the security context of the user the Fabasoft Folio AT Service has been set up for. The Fabasoft Folio Kernel running in a Fabasoft Folio AT Service is able to change this user context. So for the workflow only one Fabasoft Folio AT Service dynamically changing the user context has to be set up. A user is allowed to change the user context if the user object is entered in the *Privileged Users* box of the *Current Domain*. The user in whose context the Fabasoft Folio AT Service is running, has to be entered as privileged user. Then activities can be processed by the Fabasoft Folio AT Service in the context of the user responsible for the activity.

8.1.1 User Account

In the operating system environment, create a user account (e.g. fscatsrv) for each user in whose context a Fabasoft Folio AT Service will be running.

Note: Do not use the user account of the administrator to set up a Fabasoft Folio AT Service.

For this user, a user object has to be created in the Fabasoft Folio Domain.

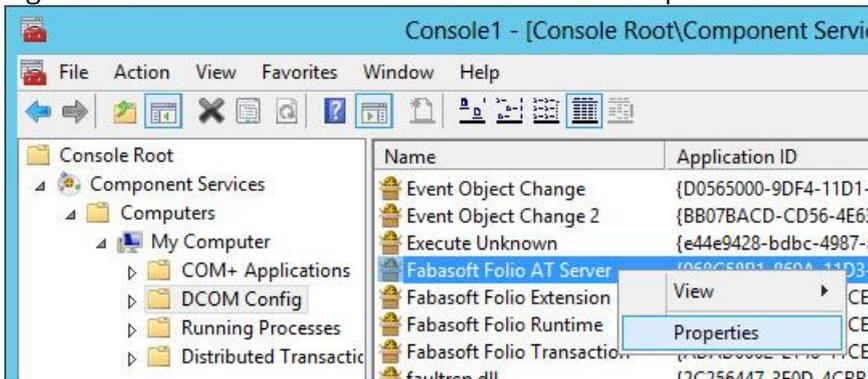
Note: If the Fabasoft Folio AT Services are installed at the same computer as the Fabasoft Folio Backend Services and the same user is performing the installation, for the user in whose context a Fabasoft Folio AT Service will be running a user object is created automatically in the Fabasoft Folio Domain.

8.1.2 Launch Permission in Microsoft Windows

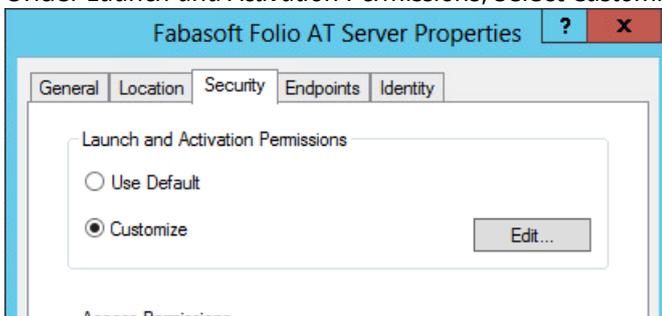
Each user, in whose context a Fabasoft Folio AT Service will be running, needs to have launch permissions for the "*Fabasoft Folio AT Server*" application. To grant these permissions, perform the following steps:

1. Open the "Component Services" MMC snap-in.
2. Open "Component Services" > "Computers" > "My Computer" > "DCOM Config", to display all available applications.

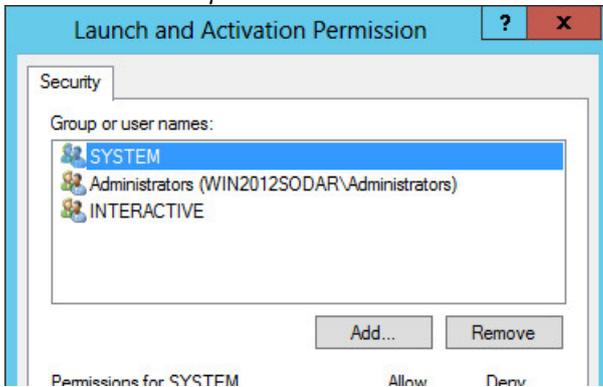
- Right-click "Fabasoft Folio AT Server" and then click "Properties".



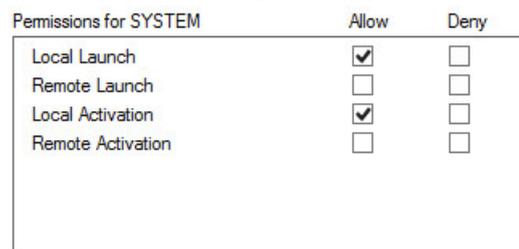
- Click the "Security" tab.
- Under *Launch and Activation Permissions*, select *Customize* and then click "Edit".



- Click "Add", to add the user in whose context the Fabasoft Folio AT Service will be running to users in the *Group or user names* box.



- In the *Enter the object names to select* box, type the user account in the context of which the Fabasoft Folio AT Service will be running and then click "OK".
- In the *Group or user names* box, select the user in whose context the Fabasoft Folio AT Service will be running and in the *Permissions for Administrators* box, in the lines *Local Launch* and *Local Activation* select the *Allow* check box.



- In the "Launch Permission" dialog box and the "Fabasoft Folio AT Server Properties" dialog box click "OK" and close the component services.

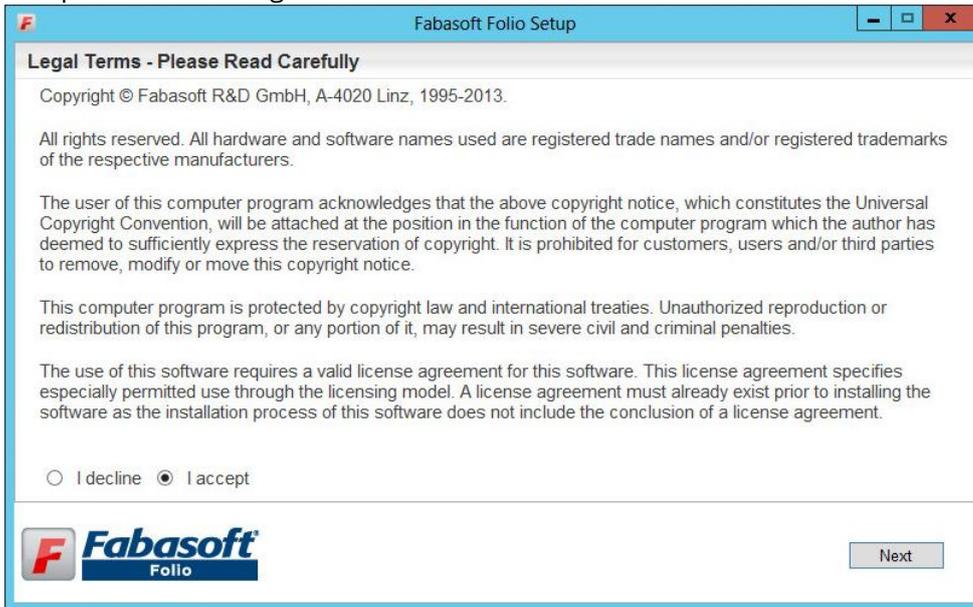
8.2 Installation of Fabasoft Folio AT Services

To install Fabasoft Folio AT Services via the automated setup, perform the following steps:

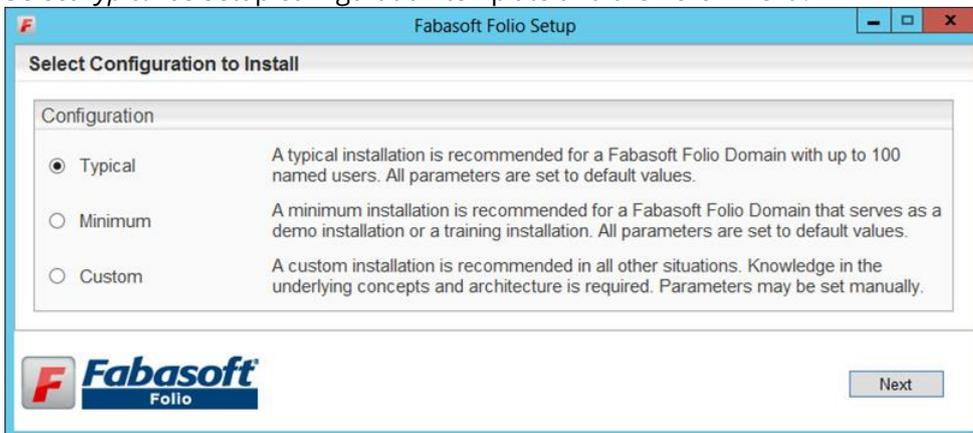
1. Insert the Fabasoft product DVD in the DVD drive of your server. The Fabasoft Server Setup dialog will be displayed automatically.

Note: If this dialog is not displayed automatically, run `setup.exe` in the root directory of the Fabasoft product DVD.

2. Accept the Fabasoft legal terms and click "Next".



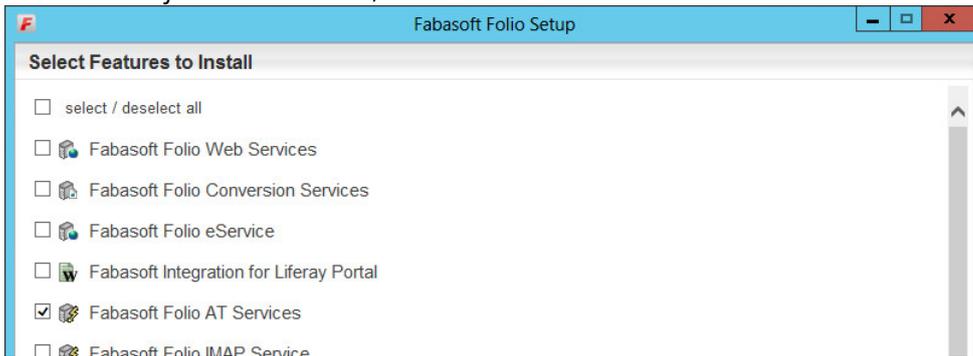
3. Select *Typical* as setup configuration template and then click "Next".



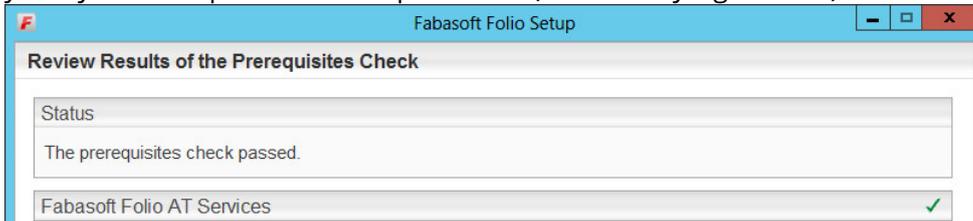
4. In the next dialog two options are available. Select *Use an existing Fabasoft Folio Domain*, in the *Host* field type the hostname of the Fabasoft Folio Backend Server, in the *Port* field type its port and click "Next".

- *Install a new Fabasoft Folio Domain*
Select this option to install Fabasoft Folio Backend Services. In the *License file* field, type the path to the license file or click "Browse" to browse the file system.
- *Use an existing Fabasoft Folio Domain*
Select this option to use an existing Fabasoft Folio backend installation.

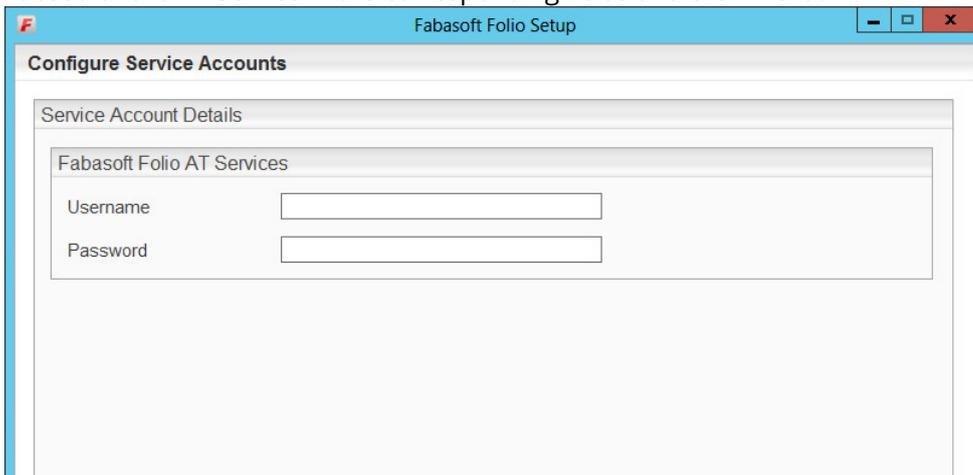
5. Select *Fabasoft Folio AT Services*, clear the checkboxes of all other services and click “Next”.



6. A check on all prerequisites is performed. Its result is presented in an overview. Ensure that your system complies with all requirements (indicated by a green tick) and click “Next”.



7. Type the *Username* and the *Password* of the user account that should be used for running the Fabasoft Folio AT Service in the corresponding fields and click “Next”.



Note: This service user must exist as an operating system user and must have a corresponding user account in the Fabasoft Folio Domain.

8. In the *Review Configuration*, the Fabasoft Folio Services can be configured. If you want to change the settings click “Edit” next to the Fabasoft Folio AT Services. Otherwise click “Install”. In this case the next steps can be omitted.

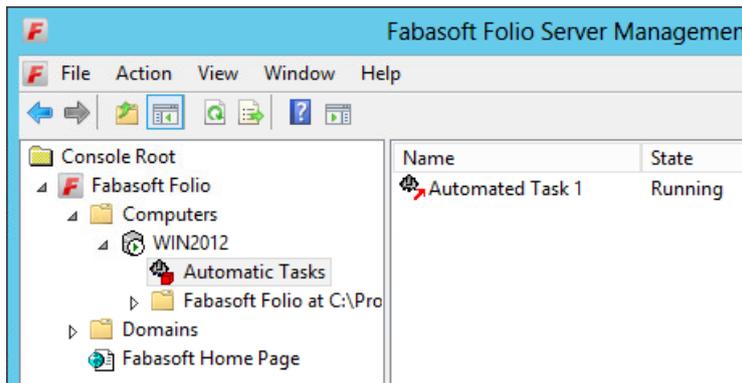
9. The following options are available to configure the Fabasoft Folio AT Services. Create a Fabasoft Folio AT Service for each user who wants to perform automated tasks. The user accounts have to be created before the installation in the operating system environment. In the Fabasoft Folio Domain the user accounts are created automatically.

Note: For all automated tasks in the workflow only one Fabasoft Folio AT Service is needed. Click the name of a Fabasoft Folio AT Service, to edit the settings.

In the *Service name* field the name of the Fabasoft Folio AT Service can be specified. Click “OK” to close this dialog.

- A Fabasoft Folio AT Service can be edited by clicking the name of the service or by clicking “Edit”.
 - A Fabasoft Folio AT Service can be deleted by clicking “Remove”.
 - A Fabasoft Folio AT Service can be created by clicking plus icon.
10. Click “OK”, to close the dialog for configuring the Fabasoft Folio AT Services.
 11. Click “Install” to start the installation.
The Fabasoft Folio AT Services are installed.

In the Fabasoft Folio Server Management navigate to “Fabasoft Folio” > “Computers” > “<computer name>”. The “Automatic Tasks” entry is available.



8.3 Create Further Fabasoft Folio AT Services

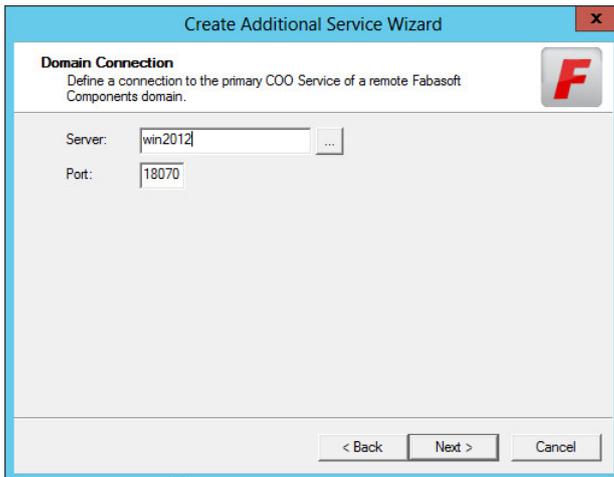
To create a Fabasoft Folio AT Service, perform the following steps:

1. Run the “Fabasoft Folio Server Management”.
2. Navigate to “Fabasoft Folio” > “Computers” > “<computer name>” > “Automatic Tasks”.
3. Right-click “Automatic Tasks”, point to “New” and click “Service”.
The “Create Additional Service Wizard” dialog for creating the Fabasoft Folio AT Service is started.
4. Click “Next”.



5. Define the connection data to the primary Fabasoft Folio COO Store.
In the *Server* field type the hostname of the computer the primary Fabasoft Folio COO Store of the Fabasoft Folio Domain is located on. In the *Port* field type the port number of the primary

Fabasoft Folio COO Store and click "Next".



6. To determine the user account in the context of which the Fabasoft Folio AT Service will be running (see chapter 8.1.1 "User Account") two options are available. Select *This account*, type the user account data in the corresponding fields and click "Next".

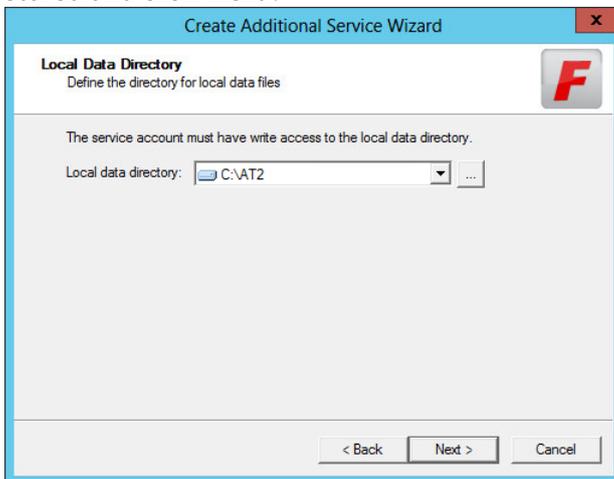


- o *This account*
When selecting this option, in the corresponding boxes the user account data have to be typed.
Note: Click "..." to browse for the user account.
- o *The system account*
When selecting this option, the local system account is used for running the Fabasoft Folio AT Services. The local system account is a predefined local account that can start a service and provide the security context for that service.

7. Type a name for the Fabasoft Folio AT Service and click "Next".



8. In the *Local data directory* field, type the path of the prepared directory where local data can be stored and click "Next".



9. A summary of the settings is displayed. Click "Next".
10. An additional Fabasoft Folio AT Service was installed. Click "Finish".

In the Fabasoft Folio Server Management navigate to "Fabasoft Folio" > "Computers" > "<computer name>" > "Automated Tasks". An additional Fabasoft Folio AT Service is available. It is stopped by default.

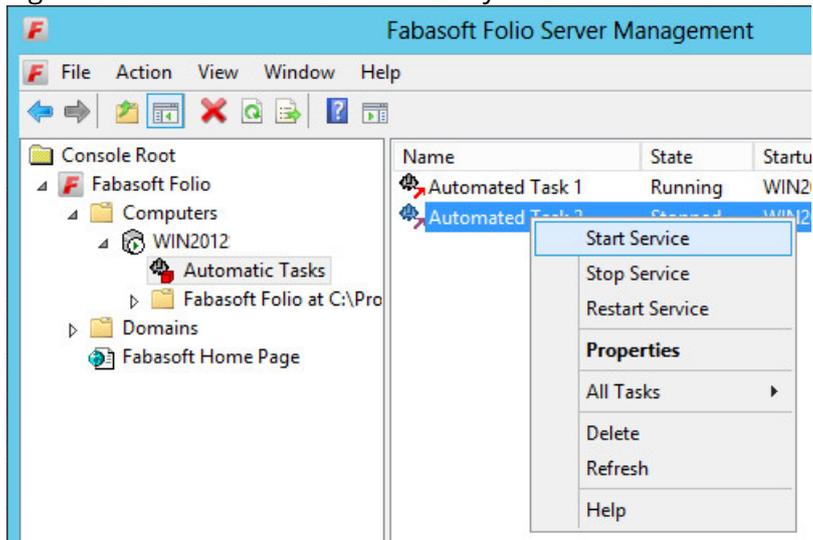
8.4 Start a Fabasoft Folio AT Service

8.4.1 Manual Start

To start a Fabasoft Folio AT Service, perform the following steps:

1. Run the "Fabasoft Folio Server Management".
2. Navigate to "Fabasoft Folio" > "Computers" > "<computer name>" > "Automated Tasks", to display all Fabasoft Folio AT Services.

3. Right-click the Fabasoft Folio AT Service you want to start and click “Start Service”.



The Fabasoft Folio AT Service is running.

8.4.2 Automatic Start

To start a Fabasoft Folio AT Service automatically when starting the operating system, perform the following steps:

1. Run the “Fabasoft Folio Server Management”.
2. Navigate to “Fabasoft Folio” > “Computers” > “<computer name>” > “Automated Tasks”, to display all Fabasoft Folio AT Services.
3. Right-click the Fabasoft Folio AT Service you want to start and click “Properties”.
4. Select *Autostart service when the operating system starts* and click “OK”.

The Fabasoft Folio AT Service is started during each start of the operating system.

8.5 Processing Automated Tasks

8.5.1 Prerequisites this Chapter is Based on

The following prerequisites have to be fulfilled for processing tasks in the background.

8.5.1.1 Prerequisites in the System Environment

- The *Fabasoft Folio/AT* software product has to be installed and a Fabasoft Folio AT Service has to be set up.
- The user for which the Fabasoft Folio AT Service has been set up needs to have launch permissions for the “Fabasoft Folio AT Server” application.

8.5.1.2 Prerequisites in the Fabasoft Folio Domain

- The user for which the Fabasoft Folio AT Service has been set up needs to have a user object in the Fabasoft Folio Domain.

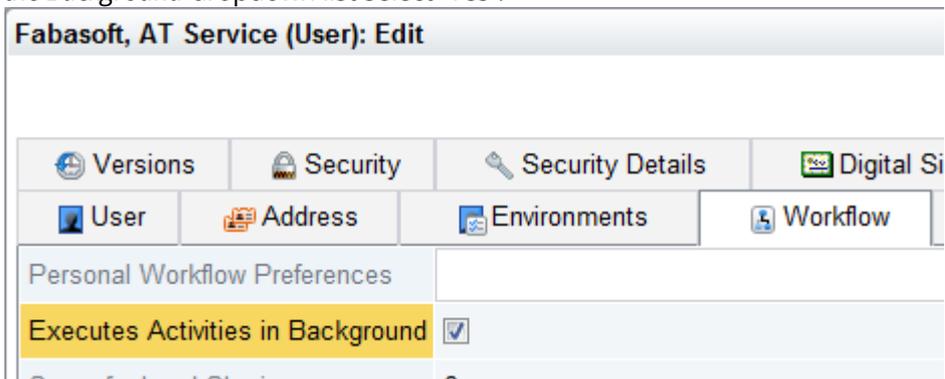
- The user object of the user, for which the Fabasoft Folio AT Service has been set up needs to have a user environment (or a Default Environment Template).

Note: If the Fabasoft Folio AT Services have been installed at the same computer as the Fabasoft Folio Backend Services and the same user has performed the installation, the prerequisites in the Fabasoft Folio Domain are fulfilled automatically.

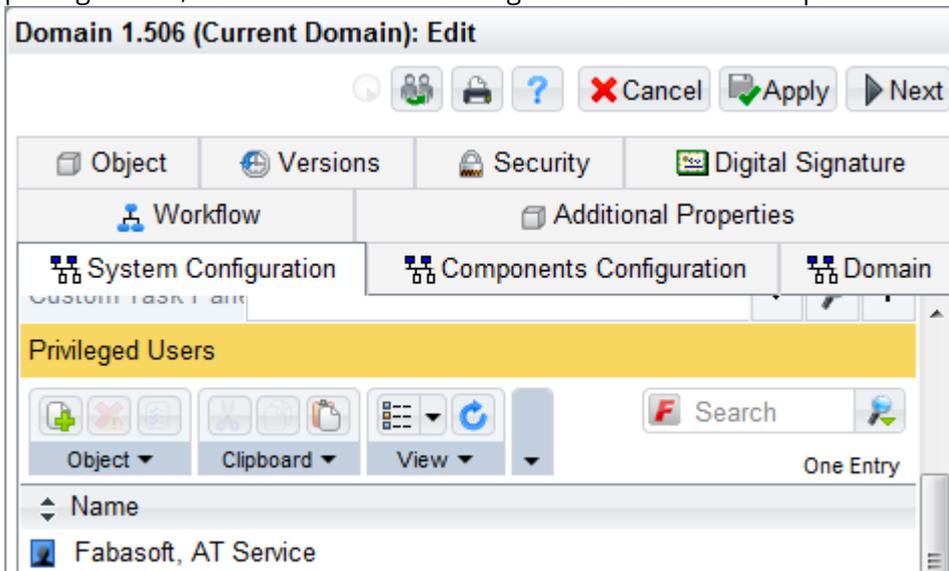
8.5.2 Settings for Processing Automated Tasks

The following settings have to be made to process automated tasks:

- Open the properties of the user object of the user in whose context the automated tasks should be processed in edit mode. On the "Workflow" tab of the *User*, in the *Executes Activities in the Background* dropdown list select "Yes".



- In the *Current Domain*, on the "System Configuration" tab, the user object has to be entered as privileged user, so that an automatic change of the user context is possible for that user.



Note: If the Fabasoft Folio AT Services have been installed at the same computer as the Fabasoft Folio Backend Services and the same user has performed the installation, this setting is done automatically.

8.5.3 List of Automated Tasks and Automated Tasks

8.5.3.1 List of Automated Tasks

When starting a Fabasoft Folio AT Service it is checked whether for the user in whose context the Fabasoft Folio AT Service is running an object of the object class *List of Automated Tasks* is available.

- If in the *User* object on the “Advanced” tab in the *List of Automated Tasks* dropdown list a *List of Automated Tasks* object has been deposited, the contained tasks are processed, even if the user in whose context the Fabasoft Folio AT Service is running is not the owner of the *List of Automated Tasks* object.

Note: If the Fabasoft Folio AT Services have been installed at the same computer as the Fabasoft Folio Backend Services and the same user has performed the installation, a *List of Automated Tasks* object is created and deposited automatically.

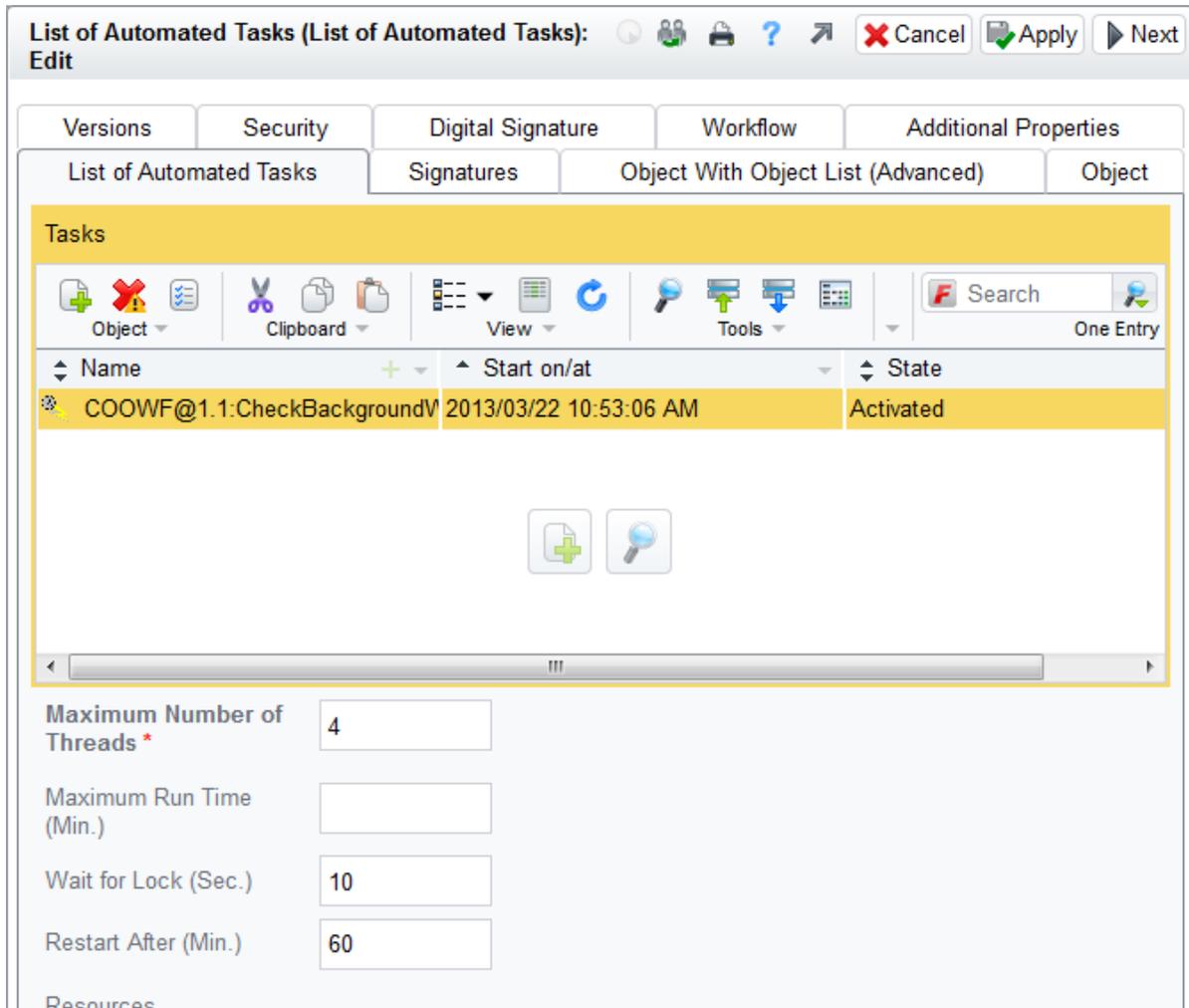
- If in the *User* object no *List of Automated Tasks* object has been deposited, then the Fabasoft Folio Domain is searched for an object of the object class *List of Automated Tasks* with the user in whose context the Fabasoft Folio AT Service is running as *Owner*. In this case the tasks contained in this *List of Automated Tasks* object are processed.
- If in the *User* object no *List of Automated Tasks* object has been deposited and in the Fabasoft Folio Domain no object of the object class *List of Automated Tasks* with the user in whose context the Fabasoft Folio AT Service is running as *Owner* is available, a new *List of Automated Tasks* object is created and processed.

To localize a *List of Automated Tasks* object created automatically, perform the following steps:

1. In the *Domain Administration*, navigate to the *User Objects* and click the “List Automated Tasks” tab.
2. On the “Edit” menu, click “Refresh”.

Now you can edit the properties of the *List of Automated Tasks* object and for example insert an *Automated Task*.

The following properties are available when editing a *List of Automated Tasks* object:



- *Tasks*
In this property, objects of the *Automated Task* object class can be specified. An object of this object class defines a task that is processed by a Fabasoft Folio AT Service. By means of an *Automated Task* it can be specified, at which time which action is processed on which object.
- *Maximum Number of Threads*
This property is used to define, how many threads are reserved in the Fabasoft Folio AT Service for processing tasks. If only one thread is used, the tasks are processed sequentially.
- *Maximum Run Time (min.)*
This property is used to define, how long system resources are allocated to a task. If a task crosses this time limit, the priority of this thread is decreased (to `BELOW_NORMAL`). Thereby for this thread less system resources are needed. If necessary, a new thread is assigned to the thread pool.
- *Wait for Lock (sec.)*
Adding a new task to the task list, the task list has to be locked during this space of time. But if a task wants to create a new task and the task list is locked due to a similar operation, the Fabasoft Folio AT Service tries repeatedly to get the lock for the *Automated Tasks* object. The *Wait for Lock (sec.)* property is used to define, how long a Fabasoft Folio AT Service should try to set a lock before an error message is displayed.

- *Restart after (min.)*
This property is used to define after which time the Fabasoft Folio AT Service tries to start again if it is not available.
- *Resources*
This property can be used to assign tasks to resources. A queue for the Fabasoft Folio AT Service and thus the sequence for processing the tasks can be defined.
By default a resource uses the same values as the standard queue (*Maximum Number of Threads, Maximum Run Time (min.), Restart after (min.)*), but this values can be overwritten per resource.

8.5.3.2 Automated Task

An object of the object class *Automated Task* defines a task which is processed by a Fabasoft Folio AT Service in the background. Each *Automated Task* has to be deposited in the *List of Automated Tasks* object.

When a Fabasoft Folio AT Service is started it is checked automatically, whether the *Check list of background activities* task is available in the *Tasks* field of a *List of Automated Tasks* object. If not, this task is created automatically. It regularly checks the Fabasoft Folio Domain for background activities which are processed.

To create an *Automated Task*, perform the following steps:

1. In the *Domain Administration*, navigate to the *User Objects*. Click *List of Automated Tasks* and open an object of object class *List of Automated Tasks*.
2. Click "Object" > "New" > "Next" and click "Properties".
3. In the *Start on/at* field, type the time when the task should be started. In the *Object* dropdown list select an object, on which the action selected in the *Action* field should be executed.
4. Click "Next".

The following properties are available when editing an *Automated Task* object:

- **State**
In this field, the current state of the task is displayed.
 - "Finished" is used to temporarily deactivate the automated task.
 - "Activated" is used to (re)activate the automated task.
 - "Running" means that the automated task is currently processed.
Note: This value cannot be selected when editing an *Automated Task* object.
 - "Disabled" means, that the automated task has been interrupted, e.g. because the processing time limit has been crossed.
- **Start on/at**
In this field, the time when the task should be processed can be defined.
- **Object**
In this dropdown list, the object is defined, on which the action selected in the *Action* dropdown list should be executed.
Note: The user in whose context the *Automated Task* is processed needs to have read or edit permissions on the object defined in this dropdown list, depending on the action which is executed on the object when the task is processed.
If in the scope of processing the *Automated Task* further objects are read or edited, the user also needs to have the necessary permissions on these objects.
- **Action**
In this dropdown list, the action which should automatically be executed on the object selected

in the *Object* dropdown list is selected.

Note: It is required, that actions for automated tasks can be processed in the background. Thus the appropriate methods must not display dialogue boxes waiting for user input.

- *Resource*

In this dropdown list, a resource can be selected. If you do not select a resource, the task is added to the standard queue of the Fabasoft Folio AT Service.

Note: If a resource is defined in this dropdown list, the queue of the Fabasoft Folio AT Service will be designated as defined in the *Resources* box in the *List of Automated Tasks* object.

- *Repetition*

Automated tasks can be processed periodically. For this purpose, in this field a value can be defined.

- *Repetition Type*

This dropdown list is used to define, how the value of the *Repetition* box will be interpreted. If for example "Minutes Delay" is selected as *Repetition Type* and "2" is typed in the *Repetition* field, the *Automated Task* is processed in a delay of two minutes. The following options are available:

- "Never Repeat": The task is only processed once.
- "Minutes/Hours/Days/Weeks/Months After Ending": The task is processed in an interval of the selected unit. The end time of the last process defines the start time of the next iteration.
- "Minutes/Hours/Days/Weeks/Months After Start": The start time for processing the task again does not depend on the endurance of the last processing, but depends on the point in time defined in the *Start on/at* field.

- *Restrict to weekdays*

This dropdown list allows to define on which weekdays the automated task should be processed.

- *Automatic Delete*

In the *Automatic Delete* dropdown list, the life time of the *Automated Task* object is defined. An *Automated Task* can for example be deleted after finishing. The following options are available:

- "No Automatic Delete": The *Automated Task* is not deleted automatically.
- "Delete Successfully Finished Task": The *Automated Task* is only deleted if processing the action of the task returns an error. If an error occurs, the state of the task is set to "Finished" and the task stays in the task list.
- "Delete Finished Task After Error": The *Automated Task* is only deleted, if processing the action of the task returns an error. If no error occurs, the state of the task is set to "Finished" and the task stays in the task list.
- "Delete Finished Task": The *Automated Task* is deleted, regardless of the result of the action. The task is not deleted immediately after finishing but after a short time removed from the task list. Periodical actions are not deleted automatically.

- *Configuration for History*

In the *Configuration for History* field, logging can be configured for processing the *Automated Task*. The following options are available for logging:

- "No Log"
- "Log Errors"

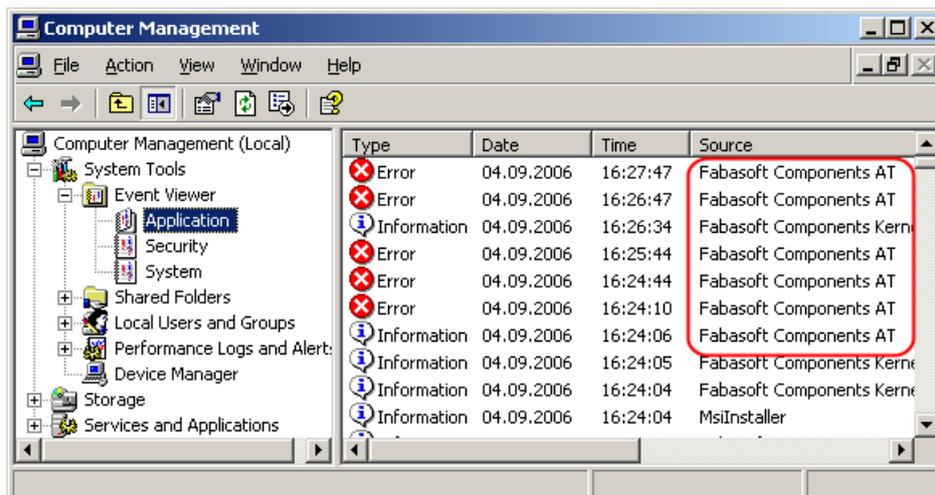
Note: Errors are also logged in the Microsoft Window event viewer. As source the identifier (Fabasoft Folio AT) of the Fabasoft Folio AT Service is displayed.

- "Log Successful Runs"
- "Log Every Run"
- **Maximum Number of History Entries**
In this field, the maximum number of entries displayed in the *History* field is defined.
- **History**
In this field, log data of processing the *Automated Task* is displayed according to the selection in the *Configuration for History* field.
- **Maximum Run Time (min.)**
In this field, the maximum run time for processing a task is defined. If the task is not completed within this time span, the priority of the thread processing the task is decreased (to `BELOW_NORMAL`). In this case, the state of the task is set to "Deactivated". For processing the task again, the state has to be set to "Waiting" manually. In the case of exceeding the specified run time, a new thread is assigned to the resource pool. So the resource again possess the configured number of threads.
Note: The value defined in this field overrides the value in a task list (in the object of the object class *List of Automated Tasks*).

8.5.4 Process

If a Fabasoft Folio AT Service is started, it logs on to the Fabasoft Folio Domain with the configured user account. The Fabasoft Folio AT Service searches for a task list of the user (`CooRuntime::GetCurrentUserRoot()`). The tasks contained in the task list are processed as configured.

The task list is defined via an object of the object class *List of Automated Tasks*. Which *List of Automated Tasks* object is processed is described in chapter 8.5.3.1 "List of Automated Tasks". The *List of Automated Tasks* object is checked for changes every 15 seconds. In case of a modification (e.g. a task has been added), the task list is evaluated and searched for tasks to be processed. Without modification, the task list is evaluated every 60 seconds. If an error occurs during the evaluation of the task list, log data can be found in the Microsoft Windows event viewer.



Note: Processing automated tasks, access rights are checked.

9 Uninstallation of Fabasoft Folio Services

Uninstallation of all installed Fabasoft Folio Services or specific types can be executed using “Uninstall a program” in the Windows Control Panel and requires administrative privileges. The uninstallation process is based on the executable files “uninstall*.exe” in the <Program Files>/Fabasoft/Components folder.

Note: Temporary files (e.g. caches) created by the Fabasoft Folio Kernel are not removed until the Fabasoft Folio Kernel and all depending Fabasoft Folio Services have been uninstalled.

10 Administration Tasks

10.1 Management of Fabasoft Folio Services

To back up or clean up Fabasoft Folio Services the command line tool `fscadmin.exe` can be used. The tool can be found in the <Program Files>/Fabasoft/Components/Management directory.

Example:

- To list all Fabasoft Folio Services use the following command:
`fscadmin.exe --action list`
- To back up the content areas of service four execute the following command:
`fscadmin.exe --action backup --service 4`
- To clean up all content areas of the Fabasoft Folio Domain until yesterday use the following command:
`fscadmin.exe --action cleanup --service all --time 1`
- To clean up all content areas of the Fabasoft Folio Domain until yesterday with an execution time limit of 60 minutes use the following command:
`fscadmin.exe --action cleanup --service all --time 1 --limit 60`
- To check the content consistency of the Fabasoft Folio Domain use the following command:
`fscadmin.exe --action checkconsistency --service all`
- To synchronize the contents of a CAS area by taking the creation date into account use the following command:
`fscadmin.exe --action cleanup --service all --syncbydate 2010-06-06`
- To synchronize the daily backup directory of a CAS area use the following command:
`fscadmin.exe --action cleanup --service all --synctobackup 2010-06-06`
- To restore the daily backup directory of a CAS area use the following command:
`fscadmin.exe --action cleanup --service all --syncfrombackup 2010-06-06`

To clean up a CAS backup directory with the need of a minimum number of passed days since the deletion of a file in the CAS Area the command line tool `cleanbackup` can be used.

Example:

- To remove files from the backup directory which have been deleted for at least 60 days in the CAS Area:
`cleanbackup -root C:/path/to/backupdirectory -days 60`

10.2 Management of the Fabasoft Folio Server Cache

To save or load the cached Fabasoft Folio objects of the Fabasoft Folio COO Service also the command line tool `fscadmin.exe` can be used.

Example:

- To save the COO addresses of all currently cached objects into a file use the following command:

```
fscadmin.exe --action savecache --file <filename> --service <COO service number>
```
- To load objects, of an already saved cache state, into the Fabasoft Folio server cache use the following command:

```
fscadmin.exe --action loadcache --file <filename> --service <COO service number>
```
- To load all objects, and all referenced objects, of a database query result into the Fabasoft Folio server cache use the following command:

```
fscadmin.exe --action loadbyquery --query "Select objname from COOSYSTEM@1.1:UserEnvironment"
```

This query initializes the Fabasoft Folio server cache with all *User Environments* and all referenced objects (e.g. commonly used objects COOSYSTEM@1.1:usrenvlocalobjects).
- To clean up the daily backup directory of a CAS area use the following command:

```
fscadmin.exe --action cleanup --service all --cleanupbackup 2010-06-06
```

10.3 Execution of Fabasoft app.ducx Expressions

To execute Fabasoft DUCX Expressions the command line tool `fsceval.exe` can be used. This tool can be found in the `<Program Files>/Fabasoft/Components/Management` directory also.

Usage: `fsceval { argument }`

```
-verbose          ... verbose
-dumptokens      ... enabledumtokens
-eval<expr>      ... evaluate <expr>
-file <path>     ... evaluate contents of file <path>
-local           ... use previous result as local scope
-global         ... use previous result as global scope
-trace          ... use debugger interfaces for trace
-flags <f>      ... use thread flags <f>
-threads <t>    ... number of threads <f>
-login <logname> ... login for user <logname>, privileged account required
-userfile<path> ... login based on user list in <path>
-statistics      ... add runtime statistics to output
```

Example:

Execution of a simple expression: `fsceval -eval "coort.Trace('hello world')"`

10.4 Additional Administration Tasks

- To clear the generated object images of an object the action `FSCCONV@1.1001:ClearObjectImages` can be used. Doing so the preview images of the object will be regenerated by the next access to the images:
Example: `<object>.FSCCONV@1.1001:ClearObjectImages();`
- To clear the generated object images of a set of objects defined by a Fabasoft Folio Query the action `FSCCONV@1.1001:ClearObjectImages` can be used:
Example: `coouser.FSCCONV@1.1001:ClearObjectImages("SELECT objname FROM`

```
COOSYSTEM@1.1:ContentObject WHERE .FSCCONV@1.1001:objimgnextretry IS NOT NULL",
"c:\\log.txt");
```

- To upgrade all saved object versions and resave the object versions with the current state the action `COOSYSTEM@1.1:UpgradeVersions` can be used. Object versions stored in delta tables (prior to Fabasoft Folio version 7) will be resaved as blob-based version. The first parameter specifies the Fabasoft Folio version up to which object versions should be upgraded. A value of 1407 means that all versions created with Fabasoft Folio versions smaller 14.0.7 will be migrated. The action can be called on a domain object meaning that all object versions in all Fabasoft Folio COO Services will be upgraded. Alternatively, a call on a dedicated COO Service is also possible, which makes parallelization possible. In case of a domain object, a Fabasoft Folio query can be specified to restrict the upgrade process to the query result.

Examples:

- `#TV.TV_LOGFILE = "c:\\upgradeversions.log";`
`coodomain.UpgradeVersions(1407);`
- `#TV.TV_LOGFILE = "c:\\upgradeversions.log";`
`<coosrv>.UpgradeVersions(1407);`
- `#TV.TV_LOGFILE = "c:\\upgradeversions.log";`
`coodomain.UpgradeVersions(1407, "SELECT objname FROM NOTE@1.1:NoteObject");`

- After updating Fabasoft Folio with a version lower than 2020 to version 2020 or higher, the ACLs must be migrated.

To update every old ACL reference to their corresponding new ACL the action `COOSYSTEM@1.1:UpgradeACLs` can be used. This action optionally takes one parameter to specify the amount of threads (default: 1), two parameters to manually specify the old ACL and the new ACL, and a parameter to specify the upgrade step to execute (default: all; `UPGRADEACLs_SERVER` migrates `COOSYSTEM@1.1:objaclobj` using the COO Service, `UPGRADEACLs_KERNEL` migrates all other ACL attributes using Kernel-based object updates). One thread acts on one COO Service at a time. Threads are only used for updating the references in `COOSYSTEM@1.1:objaclobj`. All other attributes are updated after successfully updating all references in `COOSYSTEM@1.1:objaclobj`. If some objects cannot be updated due to a lock, the update on these objects is retried once.

Examples:

- `#TV.TV_LOGFILE = "./upgradeaccls.log";`
`coodomain.UpgradeACLs(4);`
- `#TV.TV_LOGFILE = "./upgradeaccls.log";`
`coodomain.UpgradeACLs(, oldacl, replacementacl);`
- `#TV.TV_LOGFILE = "./upgradeaccls.log";`
`coodomain.UpgradeACLs(, [oldacl1, oldacl2], [replacementacl1, replacementacl2], UPGRADEACLs_SERVER);`

11 Troubleshooting

11.1 Fault Tolerant Heap (FTH)

The Microsoft Windows feature “Fault Tolerant Heap” sometimes interferes with the Fabasoft Folio setup, which results in a suspended `cooprep2.exe` process (and a therefore hanging setup). The process state can be examined using Microsoft Windows Sysinternals Process Explorer (<https://docs.microsoft.com/en-us/sysinternals/>).

The issue is typically caused by an erroneous mitigation that is applied due to a previous application fault. To reset the list of tracked applications, you can run the following command from an elevated command prompt: `rundll32.exe fthsvc.dll,FthSysprepSpecialize`

In case this does not resolve the issue, the FTH can be disabled entirely on the system using the following registry key (system restart required):

```
[HKEY_LOCAL_MACHINE\Software\Microsoft\FTH]
Enabled=0
```

Please consider the Microsoft documentation for further details:

<https://docs.microsoft.com/en-us/windows/win32/win7appqual/fault-tolerant-heap>