

eFLOW

Specifications

Version 5.2

Copyright © TIS, Top Image Systems. All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, translated, transcribed or transmitted in any form, or by any means manual, electric, electronic, electromagnetic, mechanical, chemical, optical, or otherwise, without the prior explicit written permission of TIS.

Contents

About this document	5
eFLOW hardware requirements	5
Network infrastructure	5
Data storage	5
Server hardware requirements	6
SQL server hardware requirements	6
Client hardware requirements	6
Automatic stations	6
Manual stations	7
Standalone system hardware requirements	7
Data storage devices	7
Data storage for eFLOW 5.2.1 and 5.2.2	8
Data storage for eFLOW 5.2	8
Best practice	8
Setup data	9
Dynamic data	10
eFLOW software requirements	12
eFLOW 5.2.1 and 5.2.2	12
eFLOW 5.2	12
Windows Roles and Features	13
Windows Server 2016	13
Windows Server 2012	13
Windows 10	14
Windows 8 / Windows 8.1	14
Supported operating systems	14
eFLOW 5.2.1 and 5.2.2	14
eFLOW 5.2	15
MS SQL Server	16
Supported versions	16
Authentication modes	16
SQL Server Agent	17
MSDTC	17
Verify that MSDTC is installed	17
Install MSDTC	19
Define security properties	19
Start the DTC service	20
Configure MSDTC on other machines	20

AppFabric Server (eFLOW 5.2 and earlier)	20
Overview	21
Install Microsoft Server AppFabric	21
Configure AppFabric to work with eFLOW	22
Step-by-step AppFabric Server configuration	22
AppFabric Cache additional configuration parameters	24
Troubleshooting	24

About this document

This document contains information about eFLOW 5.2 hardware and software requirements.

We strongly recommend checking that all the requirements are met before you start the eFLOW installation.

eFLOW hardware requirements

This section lists the minimum hardware requirements to install and run eFLOW.

Important: All hardware requirements listed here refer to the minimum project configuration and should serve only as a starting point for creating a list of hardware requirements for the specific project.

Network infrastructure

Component	Minimum requirement
Network adapter	Any type of Fast Ethernet network adapter at 1 Gbit and more.
Bandwidth	Between the eFLOW server and the SQL server there should be a minimum of 100 Mbps, depending on the number of users and the volume. The recommended bandwidth is 1 Gbps.
Throughput	Average of 50% of throughput dedicated to eFLOW at any given point of time.

Data storage

Server type	Required storage type
Single eFLOW server	Any hardware storage device that can be accessed by the eFLOW server and has enough free disk space for eFLOW data.
Multiple eFLOW servers (cluster configuration)	See Data storage devices .

Server hardware requirements

Component	Minimum requirement
Processor	Dual processor 2.0 GHz
RAM	2 GB. We recommend increasing the server RAM to 8 GB for the production environment.
Network cards	1GB teamed network card
Hard disk speed	10,000 RPM
Hard disk free space	100 GB. Should be increased for projects processing a large amount of data.

SQL server hardware requirements

The SQL server hardware should provide the best possible performance for data transactions. The rule of thumb is that it should be the most powerful machine in every eFLOW project.

Component	Minimum requirement
Processor	Dual processor 2.0 GHz
RAM	2 GB. We recommend increasing the server RAM to 8 GB for the production environment.
Network cards	1GB teamed network card
Hard disk speed	10,000 RPM
Hard disk free space	100 GB. Should be increased for projects processing a large amount of data.

Client hardware requirements

Automatic stations

Component	Minimum requirement per station
Processor	Pentium 4 2.0 GHz

Component	Minimum requirement per station
RAM	2 GB
Hard disk free space	15 GB for setup and dynamic data 3.5 GB for the OCR installation 0.5 GB for the eFLOW platform installation

Manual stations

Component	Minimum requirement
Processor	Pentium 4 2.0 GHz
RAM	2 GB
Display resolution	640x480, 16 colors. We recommend using displays with a resolution of 1920x1080 and 32-bit true color.
Hard disk free space	15 GB for setup and dynamic data 0.5 GB for the eFLOW platform installation

Standalone system hardware requirements

Component	Minimum requirement
Processor	Pentium 4 2.0 GHz
RAM	2 GB
Hard disk free space	30 GB

Data storage devices

This section describes different eFLOW data storage configurations and explains how to select the best option for your project.

To learn more about the internal eFLOW data storage mechanisms, see the *eFLOW data* section in the *eFLOW Implementation Guide*.

Data storage for eFLOW 5.2.1 and 5.2.2

All eFLOW setup and dynamic data can be stored either on the file system or in the MS SQL server database. The selection of the data storage type depends on the specific project requirements.

Note: In previous eFLOW versions the selection of the storage device was limited, especially for cluster environments. In eFLOW 5.2.1, the file transaction mechanism was replaced by a more effective one, allowing the removal of the existing limitations.

eFLOW application setup data contains all design time definitions, including forms and fields settings, workflow configuration and OCR engines parameters.

Application dynamic data contains information about all collections that are currently found in the system. Collection data includes the recognition results file (.DRD) and all collection attachments, starting with image files.

The eFLOW application setup and dynamic data can be stored either on the file system or in the SQL server database. The `<UseSqlDynamicStorage>` parameter in the `TisConfiguration.config` file indicates where the data will be stored.

Note: The `<UseSqlDynamicStorage>` parameter defines both setup and dynamic data storage types. It is not possible to store the setup data in the file system and dynamic data on the SQL server, or vice versa. This parameter is defined per eFLOW server and not per application. In other words, it defines the storage type for all the eFLOW applications on the current server.

After the storage device has been installed, you must make sure that the **TisAppPool** user has read and write permissions for the storage location. See the *eFLOW users* section in the *eFLOW Implementation Guide* for more information.

Data storage for eFLOW 5.2

eFLOW setup and dynamic data can be stored either in the SQL server database or on any network or local drives.

The `<UseSqlDynamicStorage>` parameter in the `TisConfiguration.config` file indicates whether the data will be saved on the SQL server.

Note: The `<UseSqlDynamicStorage>` parameter is defined per eFLOW server and not per application. In other words, it defines the storage type for all the eFLOW applications on the current server.

Best practice

The table below shows the possible data storage options. It can help you to determine the most effective method to store eFLOW data.

The recommended configurations are indicated by **bold** text.

- Setup data storage: SQL server, local disk or network share.
- Dynamic data storage: SQL server that uses an NAS device.

To learn more about the data storage options, see [Setup data](#) and [Dynamic data](#).

		Local Server Disk	NAS (shared drive)	SAN
Setup	NTFS	Supported	Supported	Supported
	SQL Server	Supported	Supported	Supported
Dynamic	NTFS	Not supported in clustered environments. Single server eFLOW installations can keep the dynamic data on the local server disk.	Not supported	Supported
	SQL Server	Supported	Supported	Supported

Setup data

eFLOW setup data can be saved either on the file system or in the SQL server database.

The table below summarizes the advantages and disadvantages of different options.

Storage type	Description	Pros	Cons
Local drive	eFLOW setup data is stored on the local server drive and is synchronized manually between the servers. This approach is good for projects where setup data is not changed too often.	There is no need to define any shared folders.	In a clustered environment, setup data must be synchronized manually between the servers.

Storage type	Description	Pros	Cons
Shared (NAS) drive	eFLOW setup data is stored on a shared drive (NAS) and can be read by all eFLOW servers.	All eFLOW servers read the same setup data. No need to synchronize the data manually between the servers.	It is necessary to have access to a shared drive (NAS). Setup data on the shared drive cannot be updated directly via the eFLOW Design module. This is because the transactional file storage used by eFLOW does not work with network shared resources. See the <i>Unsupported Scenarios</i> section in When to use transactional NTFS .
Storage area network (SAN) system	eFLOW setup data is stored in a SAN system and is fully accessible to all eFLOW servers.	All eFLOW servers have full access to setup data.	SAN systems are usually expensive and require more professional maintenance than other storage devices.
SQL server	eFLOW setup data is stored in the SQL server database.	All eFLOW setup and runtime data (management, workflow, and collections) is stored on one device.	Requires additional SQL server configuration.

Dynamic data

Application dynamic data contains information about all collections that are currently in the system. Collection data includes the recognition results file (.DRD) and all collection attachments, starting with image files.

eFLOW application dynamic data can be stored either on the file system or in the SQL server database.

The table below summarizes the advantages and disadvantages of different choices.

Note: After the storage device has been installed, you must make sure that the **TisAppPool** user has read and write permissions for the storage location. See the *eFLOW users* section in the *eFLOW Implementation Guide* for more information.

Storage type	Description	Pros	Cons
Local drive	eFLOW dynamic data is saved in the <code>[AppData]\Server\</code> <code>[ApplicationName]\Dynamic</code> folder on the local drive.	Can be easily configured for standalone or single server eFLOW projects. Does not require configuring the SQL server for storing the dynamic data.	Applicable only for standalone or single server eFLOW projects.
Shared (NAS) drive	eFLOW dynamic data cannot be stored on the shared network drive. This is because the transactional file storage used by eFLOW does not work with network shared resources. See the <i>Unsupported Scenarios</i> section in When to use transactional NTFS .		
Storage area network (SAN) system	eFLOW dynamic data is stored in a SAN system and is fully accessible to all eFLOW servers.	All eFLOW servers have the full access to dynamic data. A SAN storage system is the only NTFS storage option for clustered eFLOW environments. This is because the transactional file storage used by eFLOW does not work with network shared resources. See the <i>Unsupported Scenarios</i> section in When to use transactional NTFS .	SAN systems are usually expensive and require more professional maintenance than other storage devices.
SQL server	eFLOW dynamic data is saved in the SQL server database.	All eFLOW runtime data (management, workflow, and collections) is stored on one device. No expensive SAN devices are needed.	Requires additional SQL server configuration.

eFLOW software requirements

This section contains information about installing and configuring eFLOW software prerequisites.

The tables below summarize the eFLOW software requirements.

eFLOW 5.2.1 and 5.2.2

	Server	Client	Web Validate	Web Scan	Web Front Office
.NET 4.5 Installation	√	√	√	√	√
MS SQL Server 2012, 2014, 2016, 2017	√				
Windows Roles and Features	√		√	√	√
MSDTC	√		√	√	√

Note: MSDTC should also be installed and configured on the SQL server machine. In eFLOW 5.2 SP1, MSDTC is required when a transaction uses more than one database. For example, when SLAs are used, a transaction can include both workflow and monitoring databases.

eFLOW 5.2

	Server	Client	Web Validate	Web Scan	Web Front Office
.NET 4 Installation	√	√	√	√	√
MS SQL Server 2008, 2012, 2014	√				
Windows Roles and Features	√		√	√	√
MSDTC	√		√	√	√
Microsoft Server AppFabric	√				

Note: MSDTC should also be installed and configured on the SQL server machine. In eFLOW 5.2, we recommend using MSDTC only if it is explicitly required by the project. To use MSDTC in eFLOW 5.2, the configuration parameter <NoMsDtc> must be set to **false**.

To learn how to install and configure eFLOW software prerequisites, see the following sections:

- [Windows Roles and Features](#)
- [MS SQL Server](#)
- [MSDTC](#)
- [AppFabric Server](#) (Applies to eFLOW 5.2 and earlier. As of eFLOW 5.2.1, AppFabric is no longer required.)

Windows Roles and Features

All required Windows features and Web Server (IIS) options can be enabled using the command line DISM utility.

Note: Command line options differ slightly between operating systems. Make sure you run the relevant script.

Windows Server 2016

```
DISM.EXE /enable-feature /online /quiet /featureName:IIS-WebServerRole /featureName:IIS-WebServer /featureName:IIS-CommonHttpFeatures /featureName:IIS-StaticContent /featureName:IIS-DefaultDocument /featureName:IIS-DirectoryBrowsing /featureName:IIS-HttpErrors /featureName:IIS-ApplicationDevelopment /featureName:IIS-NetFxExtensibility /featureName:IIS-ASPNET /featureName:IIS-NetFxExtensibility45 /featureName:IIS-ASPNET45 /featureName:IIS-ISAPIExtensions /featureName:IIS-ISAPIFilter /featureName:IIS-HealthAndDiagnostics /featureName:IIS-HttpLogging /featureName:IIS-LoggingLibraries /featureName:IIS-RequestMonitor /featureName:IIS-Security /featureName:IIS-WindowsAuthentication /featureName:IIS-RequestFiltering /featureName:IIS-Performance /featureName:IIS-WebServerManagementTools /featureName:NetFx4Extended-ASPNET45 /featureName:IIS-ManagementConsole /FeatureName:NetFx3 /FeatureName:NetFx3ServerFeatures /FeatureName:WAS-WindowsActivationService /FeatureName:WAS-ProcessModel /FeatureName:WAS-NetFxEnvironment /FeatureName:WAS-ConfigurationAPI /FeatureName:WCF-NonHTTP-Activation /FeatureName:WCF-HTTP-Activation /FeatureName:WCF-HTTP-Activation45
```

Windows Server 2012

```
DISM.EXE /enable-feature /online /quiet /featureName:IIS-WebServerRole /featureName:IIS-WebServer /featureName:IIS-CommonHttpFeatures /featureName:IIS-StaticContent /featureName:IIS-DefaultDocument /featureName:IIS-DirectoryBrowsing /featureName:IIS-HttpErrors /featureName:IIS-ApplicationDevelopment /featureName:IIS-NetFxExtensibility /featureName:IIS-ASPNET /featureName:IIS-NetFxExtensibility45 /featureName:IIS-ASPNET45 /featureName:IIS-ISAPIExtensions /featureName:IIS-ISAPIFilter /featureName:IIS-HealthAndDiagnostics /featureName:IIS-HttpLogging /featureName:IIS-LoggingLibraries /featureName:IIS-RequestMonitor /featureName:IIS-Security /featureName:IIS-WindowsAuthentication /featureName:IIS-RequestFiltering /featureName:IIS-Performance /featureName:IIS-WebServerManagementTools /featureName:NetFx4Extended-ASPNET45 /featureName:IIS-ManagementConsole /FeatureName:NetFx3 /FeatureName:NetFx3ServerFeatures /FeatureName:WAS-WindowsActivationService /FeatureName:WAS-ProcessModel /FeatureName:WAS-NetFxEnvironment /FeatureName:WAS-ConfigurationAPI /FeatureName:WCF-NonHTTP-Activation /FeatureName:WCF-HTTP-Activation /FeatureName:WCF-HTTP-Activation45
```

Windows 10

```
DISM /Online /quiet /enable-feature /FeatureName:IIS-WebServerRole /FeatureName:IIS-WebServer
/FeatureName:IIS-WebServerManagementTools /FeatureName:IIS-ManagementConsole /FeatureName:IIS-
ApplicationDevelopment /FeatureName:IIS-ISAPIExtensions /FeatureName:IIS-ISAPIFilter /FeatureName:IIS-Com-
monHttpFeatures /FeatureName:IIS-DefaultDocument /FeatureName:IIS-DirectoryBrowsing /FeatureName:IIS-
HttpErrors /FeatureName:IIS-StaticContent /FeatureName:IIS-HealthAndDiagnostics /FeatureName:IIS-HttpLog-
ging /FeatureName:IIS-RequestMonitor /FeatureName:IIS-Performance /FeatureName:IIS-Security
/FeatureName:IIS-RequestFiltering /FeatureName:IIS-WindowsAuthentication /FeatureName:NetFx3
/FeatureName:IIS-NetFxExtensibility /FeatureName:NetFx4Extended-ASPNET45 /FeatureName:IIS-NetFxEx-
tensibility45 /FeatureName:IIS-ASPNET /FeatureName:IIS-ASPNET45 /FeatureName:WAS-Win-
dowsActivationService /FeatureName:WAS-ProcessModel /FeatureName:WAS-NetFxEnvironment
/FeatureName:WAS-ConfigurationAPI /FeatureName:WCF-NonHTTP-Activation /FeatureName:WCF-HTTP-Activ-
ation /FeatureName:WCF-HTTP-Activation45
```

Windows 8 / Windows 8.1

```
DISM /Online /quiet /enable-feature /FeatureName:IIS-WebServerRole /FeatureName:IIS-WebServer
/FeatureName:IIS-WebServerManagementTools /FeatureName:IIS-ManagementConsole /FeatureName:IIS-
ApplicationDevelopment /FeatureName:IIS-ISAPIExtensions /FeatureName:IIS-ISAPIFilter /FeatureName:IIS-Com-
monHttpFeatures /FeatureName:IIS-DefaultDocument /FeatureName:IIS-DirectoryBrowsing /FeatureName:IIS-
HttpErrors /FeatureName:IIS-StaticContent /FeatureName:IIS-HealthAndDiagnostics /FeatureName:IIS-HttpLog-
ging /FeatureName:IIS-RequestMonitor /FeatureName:IIS-Performance /FeatureName:IIS-Security
/FeatureName:IIS-RequestFiltering /FeatureName:IIS-WindowsAuthentication /FeatureName:NetFx3
/FeatureName:IIS-NetFxExtensibility /FeatureName:NetFx4Extended-ASPNET45 /FeatureName:IIS-NetFxEx-
tensibility45 /FeatureName:IIS-ASPNET /FeatureName:IIS-ASPNET45 /FeatureName:WAS-Win-
dowsActivationService /FeatureName:WAS-ProcessModel /FeatureName:WAS-NetFxEnvironment
/FeatureName:WAS-ConfigurationAPI /FeatureName:WCF-NonHTTP-Activation /FeatureName:WCF-HTTP-Activ-
ation /FeatureName:WCF-HTTP-Activation45
```

Supported operating systems

eFLOW can be installed on both 32-bit and 64-bit machines.

Important: (Applies to eFLOW 5.2 and earlier. As of eFLOW 5.2.1, AppFabric is no longer required.) The eFLOW server should be installed only on machines where AppFabric supports auto-start activation for WCF and WF services. Please check the Microsoft AppFabric Server official website for the most up-to-date information.

eFLOW 5.2.1 and 5.2.2

Operating System	Server	Client	Web Validate*	Web Scan*	Web Front*
Microsoft Windows Server 2016	√	√	√	√	√
Microsoft Windows 10	√	√	√	√	√
Microsoft Windows 8 / 8.1	√	√	√	√	√

Operating System	Server	Client	Web Validate*	Web Scan*	Web Front*
Microsoft Windows Server 2012 R2	√	√	√	√	√
Microsoft Windows Server 2012	√	√	√	√	√

* Web server side

eFLOW 5.2

Operating System	Server	Client	Web Validate*	Web Scan*	Web Front*
Microsoft Windows 10	√	√	√	√	√
Microsoft Windows 8 / 8.1	√	√	√	√	√
Microsoft Windows Server 2012 R2	√	√	√	√	√
Microsoft Windows Server 2012	√	√	√	√	√

* Web server side

MS SQL Server

This section provides information about settings for MS SQL server. It is possible to use the default settings and change them later or select the correct values during the installation.

Supported versions

- MS SQL server 2008 (Applies to eFLOW 5.2 and earlier; as of eFLOW 5.2.1, only MS SQL Server 2012 or later is supported)
- MS SQL server 2012
- MS SQL server 2014
- MS SQL server 2016
- MS SQL server 2017

Tested editions:

- Standard
- Enterprise

Authentication modes

eFLOW can work with either Windows or SQL authentication. After selecting the authentication mode, you must make sure that the corresponding login is defined on the SQL Server.

SQL Server authentication

Make sure that SQL Server authentication is supported and you can log in with the credentials of the planned eFLOW user. If no specific eFLOW user is required, you can start with the user **sa** and password **sa**. The **sa** user settings can be changed either via the SQL Management Studio GUI or using the [SQLCMD](#) utility:

```
SQLCMD -S localhost -Q "ALTER LOGIN sa WITH CHECK_POLICY = OFF" SQLCMD -S localhost -Q "ALTER LOGIN sa WITH PASSWORD = 'sa'"
```

Windows authentication

To use the Windows authentication (integrated security) in eFLOW, you must add the **TisAppPool** user to the SQL server logins. Note that this should be the same user as the IIS Application Pool identity set during installation.

By default, **TisAppPool** runs with the credentials **IIS APPPOOL\TisAppPool**. See [https://technet.microsoft.com/en-us/library/cc771170\(v=ws.10\).aspx](https://technet.microsoft.com/en-us/library/cc771170(v=ws.10).aspx) for more information about the IIS Application Pool identity.

The eFLOW installation will create a new SQL login, provided that the installing user has the permissions to create new SQL server logins. If, for some reason, the required **TisAppPool** identity login was not created automatically during the installation, you can add it manually later.

The login can be added either via the SQL Management Studio GUI or using the [SQLCMD](#) utility.

Local SQL Server login

By default, the IIS Application Pool identity is the local user **[IIS AppPool\TisAppPool]**. You can add the corresponding SQL login as follows:

```
SQLCMD -S localhost -Q "CREATE LOGIN [IIS APPPOOL\TisAppPool] FROM WINDOWS" SQLCMD -S localhost -Q "EXEC sp_addsrvrolemember 'IIS APPPOOL\TisAppPool', 'sysadmin'"
```

Remote SQL Server login

If Windows authentication is requested, you must add the domain user login **[DomainName\UserName]**.

You can add the corresponding SQL login using the following script:

```
SQLCMD -S localhost -Q "CREATE LOGIN [DomainName\UserName] FROM WINDOWS" SQLCMD -S localhost -Q "EXEC sp_addsrvrolemember 'DomainName\UserName', 'sysadmin'"
```

SQL Server Agent

The SQL Agent service should be running to perform database management tasks, such as cleanup jobs for eFLOW databases.

MSDTC

The Distributed Transaction Coordinator service should be running on all machines hosting eFLOW servers and SQL server.

eFLOW uses the Distributed Transaction Coordinator to enable transactions between the file system and database transaction resources.

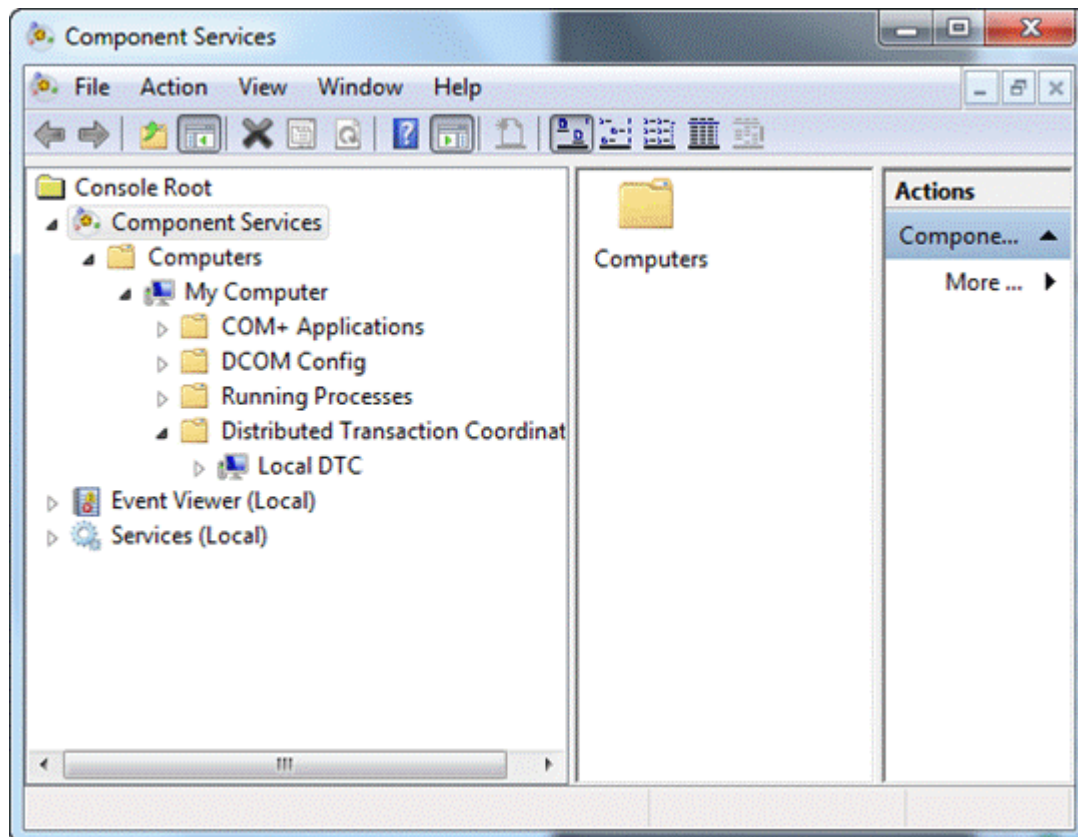
In eFLOW 5.2 SP1, MSDTC is required when a transaction uses more than one database. For example, when SLAs are used, a transaction can include both workflow and monitoring databases.

In eFLOW 5.2, we recommend using MSDTC only if it is explicitly required by the project. To use MSDTC in eFLOW 5.2, the configuration parameter <NoMsDtc> must be set to **false**.

Verify that MSDTC is installed

1. Open **Component Services**: click **Start**, type **dcomcnfg** in the search box, then press ENTER.
2. In the **Component Services > Computers** node, locate the computer for which you want to configure the Distributed Transaction Coordinator.

3. Check that **Local DTC** is in the **Distributed Transaction Coordinator** folder.

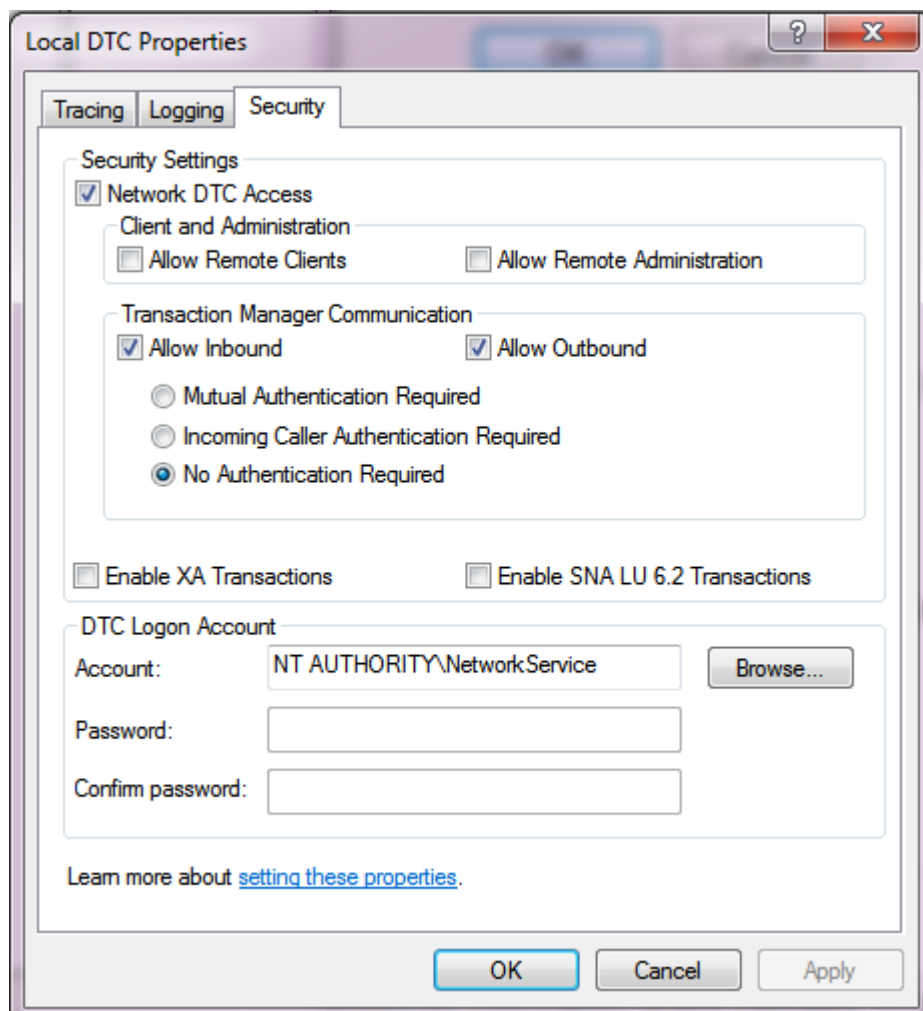


Install MSDTC

If the **Distributed Transaction Coordinator** folder does not exist, you must install MSDTC by running **msdtc -install** from the command line.

Define security properties

1. Open **Component Services**: click **Start**, type **dcomcnfg** in the search box, then press ENTER.
2. In the **Component Services > Computers** node, locate the computer for which you want to define security properties.
3. In the **Distributed Transaction Coordinator** folder, right-click on the **Local DTC** folder and select **Properties**.
4. Select the **Security** tab and modify the DTC settings if required.



The table below shows which DTC settings must be enabled for different eFLOW installations.

	Network DTC Access	Transaction Manager Communication		
		Allow Inbound	Allow Outbound	Mutual Authentication Required*
Standalone	√			
Client				
Server	√	√		√
SQL Server	√		√	√

* This is the recommended setting for Windows 2008 and higher. This is the highest security and the default option.

Start the DTC service

This can be done either using the user interface or from the command line:

- In the Windows Control Panel, go to **Administrative Tools > Services** and start the **Distributed Transaction Coordinator** service if it is not running.
- Run **net start msdtc** from the command line.

Configure MSDTC on other machines

Make sure that MSDTC is configured and running on all the relevant computers (SQL server and eFLOW servers). Repeat the steps above if required.

AppFabric Server (eFLOW 5.2 and earlier)

Note: This section applies only for eFLOW 5.2 and earlier. As of eFLOW 5.2.1, AppFabric is no longer required.

AppFabric is a set of integrated technologies that make it easier to build, scale, and manage Web and composite applications that run on IIS. AppFabric targets applications built using ASP.NET, Windows Communication Foundation (WCF), and Windows Workflow Foundation (WF).

Overview

AppFabric Server consists of two parts:

- AppFabric Caching Services, which can speed up access to frequently required information, such as session data used by an ASP.NET application.

Caching is a key technology for improving application performance and scale. It does so by taking the load off your database, and moving frequently used data into a distributed memory storage that resides close to the application components accessing it.

eFLOW 5 uses AppFabric Caching Services in multi-server projects for obtaining the eFLOW data that needs to be accessed very often or quickly, such as configuration files or chunks of huge collection data.

- AppFabric Hosting Services, which make it easier to run and manage services created with Windows Communication Foundation.

eFLOW 5 uses AppFabric Hosting services for the better management of Web services.

See <http://msdn.microsoft.com/en-us/windowsserver/ee695849.aspx> for more information.

Install Microsoft Server AppFabric

AppFabric can be installed on the following operating systems:

- Windows Server 2012 R2
- Windows Server 2012

Before installing AppFabric, read carefully all the information published on the Microsoft [AppFabric site](#). This site contains a full list of software and hardware requirements, installation instructions, links to related sources, and important troubleshooting tips.

Note: To install AppFabric on 64-bit machines, make sure that the **SetWOW** flag is disabled, that is, the 64-bit computer is not running in 32-bit mode. Run the following command line utility to set the **COMPLUS_ENABLE_64BIT** flag: `C:\Windows\Microsoft.NET\Framework64\v2.0.50727>Ldr64.exe set64`.

The Microsoft AppFabric server installation will not start if **SetWOW** is enabled.

For your convenience, the eFLOW installation includes the AppFabric Server 1.1 installation package. You can install it using the eFLOW Installation Launcher utility.

Of course, you can also [download and install](#) Microsoft Server AppFabric from official Microsoft sites.

Important: The user running the installation should have local Administrator privileges.

Configure AppFabric to work with eFLOW

Single eFLOW server

eFLOW 5 single server installations do not require AppFabric to be configured after installation.

Multiple eFLOW servers

For projects running multiple eFLOW servers, you must configure AppFabric.

If the project needs AppFabric to be fully configured, perform the following actions:

1. Create a database for the AppFabric data: Open SQL Management Studio and create a new database. You may want to select a name that refers to AppFabric in some way (for example, **AppFabricCache**).
2. Configure AppFabric Server: See [Step-by-step AppFabric Server configuration](#) for details.
3. Manage the AppFabric cluster using PowerShell:
 - a. Open **Caching Administration Windows PowerShell**.
 - b. Start the caching cluster: Type **start-cachecluster**.
 - c. Grant a Windows account permissions to connect to the cache cluster as a client: Press ENTER and type **grant-cacheallowedclientaccount**.
4. Enable automatic start of the AppFabric Caching Service: Open the **Services** dialog and make sure that the AppFabric Caching Service **startup type** is set to **automatic**.

Step-by-step AppFabric Server configuration

Open the Windows AppFabric Server Configuration Wizard, then click **Next** to continue.

Configure Hosting Services

Configure monitoring

1. Enable the monitoring configuration by checking the **Set monitoring configuration** check box.

By default, the AppFabric Event Collection service runs under the NT AUTHORITY\LOCAL SERVICE account. Do not change this unless it is required by your project.
2. Select your monitoring provider from the providers list. It should be set to **System.Data.SqlClient**.
3. Configure the monitoring store:
 - a. Click the **Configure** button to open the **Windows Server AppFabric Monitor Store Configuration** dialog box.
 - b. Check **Register AppFabric monitoring store in root web.config**.
 - c. Check **Initialize monitoring store**.
 - d. Configure the connection string by selecting your AppFabric database from the list of available databases.
 - e. Close the dialog box.
4. Click **Next** to continue.

Configure persistence

1. Enable the persistence configuration by checking the **Set persistence configuration** check box.

By default, the AppFabric Workflow Management service runs under the NT AUTHORITY\LOCAL SERVICE account. Do not change this unless it is required by your project.

2. Select your persistence provider from the providers list. It should be set to **sqlStoreProvider**.
3. Configure the persistence store:
 - a. Click the **Configure** button to open the **Windows Server AppFabric Persistence Store Configuration** dialog box.
 - b. Check **Register AppFabric persistence store in root web.config**.
 - c. Check **Initialize persistence store**.
 - d. Configure the connection string by selecting your AppFabric database from the list of available databases.
 - e. Close the dialog box.
4. Click **Next** to continue.

Configure Caching Services

1. Check the **Set Caching Service configuration** check box.

By default, the AppFabric Caching service runs under the NT AUTHORITY\LOCAL SERVICE account. Do not change this unless it is required by your project.

2. Select the **SQL Server AppFabric Caching Service Configuration Store Provider** as your caching service configuration provider.
3. Configure the caching service store:
 - a. Click the **Configure** button to open the **Windows Server AppFabric Persistence Store Configuration** dialog box.
 - b. Check **Register AppFabric Caching Service configuration database**.
 - c. Configure the connection string by selecting your AppFabric database from the list of available databases.
 - d. Close the dialog box.
4. Close the Windows AppFabric Server Configuration Wizard.

Note: SQL Agent should be running in order to perform the scheduled cleanup of AppFabric records from the database.

AppFabric Cache additional configuration parameters

Limit the size of the AppFabric process (per each node)

1. Stop-cachecluster
2. Set-cacheHostConfig {Machine name} {port(22233)} -CacheSize {cache allocation in MB}
3. Start-cachecluster

Control object TTL in the AppFabric cache

Ensure that **Expirable** is set to **True** and the **TTL** value is set to at least 10 min.

Troubleshooting

See the *AppFabric* errors troubleshooting section in the *eFLOW Implementation Guide*, which covers the problems that may arise during the AppFabric installation or at runtime.