Kofax eFlow
Configuration Guide

Version: 6.1.0

Date: 2023-08-29

KOFAX

Table of Contents

Preface

Kofax eFlow is a highly customizable system that can perform a wide variety of tasks. The standard installation configures eFlow so that it is possible to begin working immediately needing no additional configuration effort. However, eFlow configuration often needs to be adjusted to specific project requirements. This guide details information on how to configure and customize different eFlow components.

Product documentation

To access the full Kofax eFlow documentation set online, see the Kofax eFlow Product Documentation page. For a complete set of Kofax eFlow documents, refer to the Kofax eFlow Release Notes.

Chapter 1

eFlow configuration

This chapter provides information about the most common eFlow configurations, such as basic configuration, user permission, and SQL server.

Basic configuration

This section provides information about the eFlow configuration files.

Configuration files

This section describes the structure of the eFlow configuration files.

Basic configuration files

The basic eFlow configuration files (TisConfiguration.config) are created in the `../bin/ConfigSources` folders of the eFlow server and client-side installations.

Every configuration file contains both the values set by the installing user and default parameters that can be changed later if required. The structure of the server and client-side configuration files reflects the functional differences between them.

See Configuration parameters for a detailed explanation of all basic configuration parameters.

Server-side configuration

The server-side TisConfiguration.config file consists of the following configuration sections:

- <eFlowPlatformConfig> contains parameters required by both the clients and the server.
- <eFlowClientProfilesConfig> contains client-side parameters that can be configured from the server. You can create several client profile sections with different names.

  When the configuration file is created, it contains only the Default client profile.
- <eFlowPlatformConfigServer> contains parameters required only by the server.

Client-side configuration

The client-side TisConfiguration.config file consists of the following configuration sections:

- <eFlowPlatformConfig> contains parameters required by both the clients and the server.
- <eFlowClientConfig> contains parameters required only by the client.

The <ProfileName> parameter contains the name of the relevant client profile section configured on the server.

Event log configuration

Separate TisLogger.config files are created for the eFlow server, client, and Web applications.

The files are copied during the installation to the `/ConfigSources` folder in the relevant `/Bin` folders. They contain default values that can be changed later if required.

The following section is included in all eFlow station configuration files and should be added to every custom station configuration:

```
<appSettings> <add key="BasicConfigurationFile"
    value="ConfigSources\TISconfiguration.config"> <add key="LoggerFile"
    value="ConfigSources\TISLogger.config"> <appSettings>.
```

Configuration parameters

The following tables provide detailed information about eFlow basic configuration parameters.

Installation paths

Parameter	Section	Description	Notes
<eFlowInstallPath>	<eFlowPlatformConfig>	eFlow installation folder location	
<eFlowDataPath>	<eFlowPlatformConfig>	Path to the eFlow data folder	
<eFlowOCRsPath>	<eFlowClientConfig>	OCR engines installation location	
<eFlowDBEnginesPath>	<eFlowClientConfig>	DBEngines folder location	Sets automatically under the eFlow client installation path.
<eFlowLangPath>	<eFlowClientConfig>	Localization files folder location	Sets automatically under the eFlow client installation path.

SQL server definitions

Parameter	Section	Default value	Description	Notes
<IntegratedSecurity>	<eFlowPlatformConfigServer>	false	Indicates whether the SQL server will use Windows or user authentication.	Sets the Integrated Security SQL connection string parameter.
<LocalSQLServer>	<eFlowPlatformConfigServer>	(local)	Contains a local or remote SQL server instance name.	

Parameter	Section	Default value	Description	Notes
<DBPassword>	<eFlowPlatformConfigServer>	sa	SQL server user.	Relevant only if SQL Server authentication is used.
<DBUserName>	<eFlowPlatformConfigServer>	sa	SQL server password.	
<DBName>	<eFlowPlatformConfigServer>	eFlow_Management	eFlow management database name.	
<MonitorDBName>	<eFlowPlatformConfigServer>	eFlow_Monitor	eFlow monitoring database name.	
<SqlConnectionTimeoutSec>	<eFlowPlatformConfigServer>	60 sec	SQL connection string parameter.	
<MaxPoolSize>	<eFlowPlatformConfigServer>	1000	SQL connection string parameter.	

Data storage

Parameter	Section	Default value	Description	Notes
<UseSqlDynamicStorage>	<eFlowPlatformConfigServer>	false	Indicates whether eFlow dynamic and setup data will be saved in the SQL database or the file system.	It is not possible to store setup data in the file system and dynamic data in the SQL database or vice versa.

Basic configuration data client profile

Parameter	Section	Default value	Description	Notes
<ProfileName>	<BasicConfigurationDataClientProfile>	Default	The client profile name.	This section is created in the server-side configuration file. Client configuration files contain the relevant profile names. During the installation only one profile called Default is created.

Communication

Parameter	Section	Default value	Description	Notes
<CommunicationProtocol>	<eFlowClientConfig>	https	Selected communication protocol.	The server-side communication parameters are saved on IIS, and they should be the same as those set on the client-side configuration files.

Machine names

Parameter	Section	Default value	Notes
<LocalMachineName>	<eFlowPlatformConfig>	Local machine name. Set automatically during the installation.	The clustered eFlow server local machine name should contain the cluster DNS name.
<ServerMachineName>	<eFlowPlatformConfig>	eFlow server machine name	Server or standalone: Local machine name. Clustered eFlow server: Cluster DNS name. eFlow client: The server name is defined during the installation.

Resource disposal and task management intervals

Parameter	Section	Default value	Description	Notes
<TaskLifeTimeoutSec>	<eFlowPlatformConfigServer>	600 sec	Defines how long the client task notification will be kept on the server.	This time interval should always be larger than the <TaskPollingInterval> set on the client. Otherwise, it can happen that the client never receives its tasks.

Parameter	Section	Default value	Description	Notes
<ResourceDisposerIntervalSec>	<eFlowPlatformConfigServer>	60 sec	Defines how often the resource disposal service checks whether any stations is not working.	This parameter can be increased to reduce the server load. It means the license will become available again a long time after the station stopped working.
<StationLivelinessTimeoutSec>	<eFlowPlatformConfigServer>	600 sec	A timeout that defines if the station is still alive.	If the time that passed since the client last contacted the server exceeds the <StationLivelinessTimeoutSec> value, the resource disposal service releases the station resources (license, management, and workflow databases, and so on).
<TaskPollingIntervalSec>	<BasicConfigurationDataClientProfile>	300 sec	Defines how often the client contacts the server. It checks for tasks even if the station is idle.	It makes sense to increase this time if the server is very busy. In this case, the <StationLivelinessTimeoutSec>and <MonitorAutorunStationsIntervalSec> values should also be increased. This interval should always be smaller than <TaskLifeTimeoutSec>.

See eFlow server mechanisms for more information on resource disposal.

Autorun stations

Parameter	Section	Default value	Description	Notes
<AutorunStarterReconnectionIntervalSec>	<BasicConfigurationDataClientProfile>	300 sec	Defines the time interval for recreating the node if IIS is not working.	Prevents the AutorunStarter service from starting to work before IIS is ready.

Parameter	Section	Default value	Description	Notes
<MonitorAutorunStationsIntervalSec>	<BasicConfiguration DataClientProfile>	600 sec	Defines how often the AutorunStarter service checks the server for tasks to start or stop any stations.	This interval should be always greater than <TaskPollingIntervalSec>.

Collections

Parameter	Section	Default value	Description	Notes
<CollectionsChunkSize>	<BasicConfiguration DataClientProfile>	2,000,000 bytes	Huge collections are transferred between the client and the server in chunks. This parameter defines the size of these chunks.	Should be always smaller than <HugeCollectionsThreshold>.
<CollectionsExportImportChunkSize>	<BasicConfiguration DataClientProfile>	200	The number of collections taken together during the export or import of dynamic data in the Control station.	This parameter can be changed depending on the available amount of memory.
<HugeCollectionsThreshold>	<eFlowPlatformConfigServer>	10,000,000 bytes	If the collection size is bigger than this threshold, all collection data is transferred in chunks according to the <CollectionsChunkSize> parameter.	Should be always bigger than <CollectionsChunkSize>

Miscellaneous parameters

Parameter	Section	Default value	Description	Notes
<DemoMode>	<eFlowPlatformConfig>	false	Indicates whether eFlow is running in Demo mode.	In Demo mode, sometimes intervals are set to their minimal values. Resource disposal, autorun stations start, and so on, are performed immediately, but it can result in a high load on the server. It is not recommended to use this mode in a production environment.

Parameter	Section	Default value	Description	Notes
<PerformanceCounterIntervalSec>	<BasicConfigurationDataClientProfile>	300 sec	Defines how often system performance parameters are checked to be displayed in the Supervise station.	For internal use only.
<AlertsGenerateIntervalSec>	<eFlowPlatform >	60 sec	Defines how often the system checks whether any alerts should be issued.	For internal use only.
<EnableEquationStatistics>	<eFlowPlatformConfig>	false	Indicates whether the eFlow usage statistics are enabled.	
<SaveUserAuthentication>	<BasicConfigurationDataClientProfile>	true	Indicates whether user authentication information is cached on the client machine.	User login information is saved automatically in isolated storage to hide the login dialog from the same user. It is possible to clear the authentication cache manually using the Basic Configuration utility. In certain situations, caching user information may be regarded as a security problem.

Encrypt configuration

This section explains how to encrypt and decrypt the eFlow server configuration section in TisConfiguration.config.

For more information, refer to the "Encrypting and Decrypting Configuration Sections" on the Microsoft website.

Encrypt the server configuration

Encrypting the server configuration is a common project security requirement. It can be easily done using the aspnet_regiis utility found under `C:\Windows\Microsoft.NET\Framework64\v4.0.30319`.

To encrypt the eFlow server configuration section:

1. Go to `C:\inetpub\wwwroot\Tis Web Site\eFlow_6\Bin\ConfigSources`.
2. Comment out the <configSections> section in TisConfiguration.config:

```
 <!--<configSections> <section
name="eFlowPlatformConfig"
```

```
type="TiS.Core.TisCommon.Configuration.CustomSection, TiS.Core.TisCommon.Common,
Version=6.0, Culture=neutral, PublicKeyToken=0752346849557bc9" />
<section name="eFlowPlatformConfigServer"
type="TiS.Core.TisCommon.Configuration.CustomSection, TiS.Core.TisCommon.Common,
Version=6.0, Culture=neutral,
PublicKeyToken=0752346849557bc9" /></configSections>-->
```

3. Temporarily rename TisConfiguration.config to web.config.
4. Encrypt the <eFlowPlatformConfigServer> section:

```
 aspnet_regiis -pef "eFlowPlatformConfigServer" "C:\inetpub\wwwroot\Tis Web Site
\eFlow_6\Bin\ConfigSources"
```

5. Uncomment the <configSections> section.
6. Rename the web.config file back to TisConfiguration.config.
7. Restart IIS.

Restore the original configuration

To restore the original configuration file, perform the same steps as required for encryption, but the aspnet_regiis command should look slightly different:

```
aspnet_regiis -pdf "eFlowPlatformConfigServer" "C:\inetpub\wwwroot\Tis Web
    Site\eFlow_6\Bin\ConfigSources"
```

User permissions

This section describes eFlow users and their security requirements.

eFlow users

This section summarizes what permissions should be granted to different types of eFlow users.

Installing user

The user performing the eFlow installation should have local Administrator rights. See Under the hood: Installation for details about the actions performed during the eFlow installation.

If the installing user has permissions to create SQL server logins, the installation will create a new login based on the SQL server instance name and authentication mode selection. Otherwise, the relevant SQL login should be created manually by the DBA.

TisAppPool user

By default, the TisAppPool user runs under the ApplicationPoolIdentity (for more information on ApplicationPoolIdentity, refer to "Application Pool Identities" on the Microsoft website). This user is generated automatically by IIS when TisAppPool is created during the eFlow installation. In this case, all required permissions need to be given to the local user IIS AppPool/TisAppPool,

It is common practice to run the TisAppPool under a specific user identity.

Normally, the eFlow server installation automatically sets all the necessary access permissions for the TisAppPool user.

TisAppPool user rights and permissions need to be configured only on machines running eFlow Server.

To set the TisAppPool identity permissions manually, follow the instructions below.

File system access

The TisAppPool user should have full access to the following folders on the eFlow server machine:

- eFlow server folder (Tis Web Site physical path)
- eFlow application data folder

Registry access

The TisAppPool user account should be granted read access to the following registry key: `HKEY_LOCAL_MACHINE\SOFTWARE\TopImageSystems\eFlow 6`.

SQL server permissions

If Windows authentication mode is selected, the TisAppPool user should have the following SQL server permissions:

- Create a database

  The eFlow_Management database is created when the TIS application pool starts.

  The application workflow database is created when the application is installed.
- Read/write database access permissions to run stored procedures.

  Required at runtime.

If SQL authentication mode is selected, these permissions should be granted to the relevant SQL login.

If the SQL server is installed on the remote machine, the TisAppPool user should be the domain user.

Debug rights

On standalone eFlow machines, add the TisAppPool user to the user group that has Debug rights (Refer to "Privilege Constants" on the Microsoft website for more information). It will allow the use of a fast resource disposal mechanism instead of the standard client-server mechanism based on the timeouts.

This feature can be very useful for application developers who often need to start and stop the eFlow modules, and need to be sure that all resources are freed immediately.

STS and eFlow server certificates

eFlow installs default certificates (TiS-eFlow-rp.pfx and TiS-eFlow-rpsts.pfx) in the `LocalMachine \Personal certificate store` on eFlow server machines. The TisAppPool user account should be granted full access permissions to these certificates.

Windows Event Log

The Security event log does not inherit permissions from the Event log, therefore, the TisAppPool account must have the following access permissions to be able to write to the Windows Event Log.

- Read - `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\eventlog`
- Full Control - `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\eventlog\Security`

If one of these settings is not present, IIS is not able to write to the TIS log.

Interactive eFlow user

Users running manual eFlow stations should be granted full access permissions to the eFlow application data folder on client machines.

T.i.S. eFlow Autorun Stations Starter user identity

Configure the T.i.S. eFlow Autorun Stations Starter users according to your requirements.

Default configuration

By default, the **T.i.S. eFlow Autorun Stations Starter** service runs under the Local System identity and has all the necessary permissions to start working immediately after installation.

Non-default user identity

For a standalone eFlow machine, the **T.i.S eFlow Autorun Stations Starter** service should run either as Local System (this is the default configuration) or local Administrator.

If the Client and Server are installed on different machines, the domain user must be used as the **T.i.S eFlow Autorun Stations Starter** service identity due to the eFlow security internal mechanism.

User permissions

Configure the **T.i.S. eFlow Autorun Stations Starter** service user as follows:

- Grant Login as service rights.
- Grant full access to the eFlow application data folder.
- Define as one of the application users in the Default STS configuration on the eFlow server.
- Add to the Performance Monitor Users local group. For more information, refer to "Performance and Reliability Monitoring" on the Microsoft website.

Set user permissions

This section describes the user permissions needed by the IT teams responsible for configuring eFlow environments.

See eFlow users for details about the different eFlow user types.

eFlow user types

User type	Description	Default
Installing user	The user performing the eFlow installation.	Local Administrator
TisAppPool user	The eFlow_6 web application runs under TisAppPool. TisAppPool should run under a domain user identity if the SQL server is installed on a remote machine.	ApplicationPoolIdentity (IIS AppPool\TisAppPool)
TisWebAppPool user	Each eFlow web application server runs under its specific application pool: TisWebValidateAppPool, TisWebScanAppPool, and TisWebFrontAppPool.	ApplicationPoolIdentity (for example, IIS AppPool \TisWebValidateAppPool)
AutorunStation user	T.i.S. eFlow Autorun Stations Starter service user.	System user
Interactive user	User working with any eFlow desktop manual station.	Unprivileged local user

File system access permissions

eFlow server machine

Folder	eFlow user	Required permissions	Notes
`[IIS installation folder]\wwwroot\Tis Web Site`	TisAppPool user	Full access	By default, IIS is installed on drive C. If it was moved to another drive, the eFlow installation will follow it automatically.

eFlow web station server machine

Folder	eFlow user	Required permissions	Notes
`[IIS folder]\wwwroot \Web Validate`	TisWebValidateAppPool user	Full access	By default, IIS is installed on drive C. If it was moved to another drive, the eFlow installation will follow it automatically.
`[IIS folder]\wwwroot \Web Scan`	TisWebScanAppPool user		
`[IIS folder]\wwwroot \Web Front`	TisWebFrontAppPool user		

eFlow client machine

Folder	eFlow user	Required permissions	Notes
eFlow [AppData]	Interactive user	Full access	

Folder	eFlow user	Required permissions	Notes
	AutorunStationStarter user		

Registry entries

Registry key	eFlow user	Required permissions	Notes
`HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\eventlog`	TisAppPool user	Read	If one of these values is not set, IIS will not be able to write to the TIS log.
`HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\eventlog\Security`	TisAppPool user	Full control	

Local security policies

User rights	eFlow user	Notes
LogonAsService	AutorunStationStarter service user	
Debug(SeDebugPrivilege Group Policy)	TisAppPool user	Needs to be assigned on standalone machines only.

Local user groups

User group	eFlow user	Notes
Performance Monitor Users	AutorunStationStarter service user	

SQL server

This section describes various configuration and management tasks related to Microsoft SQL Server.

Remote SQL server

It is common practice to install the SQL server separately from the eFlow server. Very often, the project requirements include using company SQL servers across applications.

eFlow is designed to work with a remote SQL server as well as with a local server, but it should be carefully configured to get the best results.

eFlow is optimized to work with the collation SQL_Latin1_General_CP1_CI_AS. Using any other collation may result in errors.

Before installing eFlow

1. Prepare the eFlow server machine according to the standard procedure.
2. Provide proper permissions for the installing user.

   The eFlow installation attempts to create a new SQL login based on the SQL server instance name and authentication mode selection. If you need this to be done, make sure that the user performing the eFlow server installation has permissions to create SQL server logins.
3. Select the domain user identity to be used later by TisAppPool. This domain user should be able to access the remote SQL server machine.
4. Check that the SQL server is configured properly to work with eFlow.

Install the eFlow server

1. Start the eFlow server installation.
2. Set the TisAppPool identity as the domain user identity defined in the previous stage (see Before installing eFlow).
3. Select the SQL server connectivity parameters.
   - SQL server instance name
   - Authentication mode: Defines the user who will create and manage the eFlow databases.
     - Windows: TisAppPool account.
     - SQL: SQL login.
4. Username and password: Relevant only if the SQL authentication mode was selected.

After the eFlow server installation

Verify that eFlow is ready to work with the remote SQL server.

eFlow configuration parameters

If the SQL server was not configured properly during the installation, you can change the configuration later.

All the relevant settings are stored in the following configuration file.

```
C:\inetpub\wwwroot\TiS Web Site\eFlow_6\Bin\ConfigSources
\TisConfiguration.config.
```

Restart the IIS after changing the eFlow configuration.

SQL server logins

By default, the eFlow installation attempts to create a new login with sysadmin permissions. This is fine for testing on the standalone machine; however, for large projects, the DBA must set the account for authentication as follows:

- Windows authentication: Add the TisAppPool domain account to the SQL server logins and grant the relevant permissions.
- SQL authentication: Create a new SQL server login with relevant permissions.

SQL server permissions

The eFlow user (either the TisAppPool account or SQL login) must have permission to perform the following actions on the SQL server:

- Create the databases: eFlow_Management, eFlow_Monitor, and <ApplicationName>_Workflow. The databases are created during the CAB installation.
- Manage databases at runtime:
  - Read
  - Write
  - Add/Remove tables (required only during version upgrade/update)
  - Run stored procedures
- Run stored procedures on the master database (used, for example, when the user deletes applications).

The eFlow user can just have dbcreator server and db_owner database rights if granular permissions are not required.

Troubleshooting

The eFlow server installation includes all the files needed to work with remote SQL. If problems still occur at runtime, it is recommended to download and install the SQL server management objects together with other SQL packages that they depend on.

- Microsoft SQL Server 2014 Feature Pack.
- Microsoft SQL Server 2016 Feature Pack.
- Microsoft SQL Server 2017 Feature Pack.
- Microsoft SQL Server 2019 Feature Pack.
- Microsoft SQL Server 2022 Feature Pack.

For more information on Microsoft SQL Server 2014 to 2022 feature packs, refer to the corresponding site on the Microsoft website.

Create eFlow databases

Usually, eFlow databases are created automatically during the platform and application installations. In this case, the following conditions should be met:

Requirement	Usage
The user installing eFlow should have permissions to create new logins.	The installation will run the script creating a new login for the IIS\TisAppPool user (Windows authentication) or a new SQL user (SQL authentication). By default, this new login will get administrator privileges.

Communication ports

This section describes the different aspects of eFlow communication ports configuration.

Configure HTTPS

This section describes how eFlow works with HTTPS, what configuration changes are required, and how to analyze possible problems.

Hypertext Transfer Protocol Secure (HTTPS) is the secure version of HTTP, the protocol over which data is sent between your browser and the website that you are connected to. It means all communications between your browser and the website are encrypted.

The main motivation for HTTPS is authentication of the visited website and protection of the privacy and integrity of the exchanged data.

The security of HTTPS is that of the underlying TLS, which uses long-term public and secret keys to exchange a short-term session key to encrypt the data flow between client and server.

X.509 certificates are used to guarantee that you are talking to the partner with whom you want to talk. Consequently, certificate authorities and public key infrastructure are necessary to verify the relation between the owner of a certificate and the certificate, as well as to generate, sign, and administer the validity of certificates.

HTTPS in eFlow installation

HTTPS is selected by default as the eFlow communication protocol that works out of the box after the installation.

The setup process modifies the eFlow server and client configuration to enable the HTTPS protocol.

eFlow server

The following is the configuration of the eFlow server.

Select the SSL certificate

The secure communication protocol requires an SSL certificate to check the trusted client-server connection.

The eFlow installation allows the user to select the certificate that will be attached to the HTTPS binding definition. It can be either a new self-signed SSL certificate or one of the certificates that are already installed on the server machine.

The tables below compares the advantages and disadvantages of different certificate types that can be used by eFlow.

Type	Pros	Cons	Notes
Self-Signed Certificate	Created during the installation. Free of charge. Can be managed by the local machine administrator.	Potential security risk on the client machines. It needs to be installed as a trusted certificate on client machines to avoid security warnings. Cannot be automatically revoked if compromised.	We recommend using self-signed certificates only on standalone machines.
Domain Certificate	Free of charge. Provides good domain-level security. Does not require any additional actions on the client machines.	Cannot be used in environments that include several domains.	
CA Certificate	High level of security. CA usually provides easy-to-use management tools. Revoked automatically if needed.	Can be expensive. Requires professional IT management.	

HTTPS binding

HTTPS binding is added to TiS Web Site bindings. By default, the installation selects the "55444" port for the HTTPS communication protocol. If required, any other port number can be chosen by the user.

The user performing the installation must select the SSL certificate that will be attached to the HTTPS binding definition.

Web Services configuration

The installation configures eFlow web services to work with HTTPS. The relevant changes are reflected in the eFlow 6 and TisDefaultSTS web configuration files.

Files with the extensions .config_http and .config_https are examples of web.config files that contain HTTPS and HTTP settings. These files are available in the eFlow 6 and TisDefaultSTS local folders.

eFlow basic configuration

If the HTTPS communication protocol is selected, the <CommunicationProtocol> parameter value in TisConfiguration.config should be https.

Both the eFlow 6 and TisDefaultSTS basic configuration files should contain the same communication protocol type.

eFlow client

The following is the configuration of the eFlow client.

SSL certificates

The eFlow client installation does not perform any validation of the SSL certificates. If a trusted domain or CA certificates are used, the clients work properly immediately after the installation.

If self-signed certificates are used, authentication warnings are issued on the client side. Use the following procedure to prevent these warnings:

1. Export the self-signed certificate installed on the eFlow server. For more information, refer to "Export a Server Certificate (IIS 7)" on the Microsoft website.
2. Import the certificate to the Trusted Root certificate repository on every client machine.

Web Services configuration

The eFlow client installation performs the relevant changes in the Binding.config and Client.config files.

Files with .config_http and .config_https extensions are the examples of configuration files that contain HTTP and HTTPS settings. These files are available in the `<eFLOWInstallPath>/Bin/ConfigFiles` folder.

Switch between HTTP and HTTPS

The eFlow installation includes examples of configuration files containing HTTP and HTTPS settings.

To switch between these communication protocols, you need to replace the existing configuration files with the samples brought in by the installation and change the default server name.

- HTTP: The default server name is QA-SERVER-01.
- HTTPS: The default server name is QA-SERVER-01.TisTesting.local.

Use the following procedure to switch between HTTP and HTTPS.

Server-side configuration

1. Open the `C:\inetpub\wwwroot\Tis Web Site\eFlow_6` folder.
   - In the Web.config file:
     a. Rename the existing web.config file to web.config.temp.
     b. Change the name of the relevant configuration file (either web.config_http or web.config_https) to web.config.
     c. Replace all occurrences of the default server name in the new web.config file.
2. Open the `C:\inetpub\wwwroot\Tis Web Site\TisDefaultSTS`folder.
   - In the Web.config file:
     a. Rename the existing web.config file to web.config.temp.

   b. Change the name of the relevant configuration file (either web.config_http or web.config_https) to web.config.
   c. Replace all occurrences of the default server name in the new web.config file.
- In the TisConfiguration.config file:
   a. Open the `..\Bin\ConfigSources` folder.
   b. Rename the existing TisConfiguration.config file to TisConfiguration.config.temp.
   c. Change the name of the relevant configuration file (either TisConfiguration.config_http or TisConfiguration.config_https) to TisConfiguration.config.
   d. Replace all occurrences of the default server name in the new TisConfiguration.config file.
   e. In the TisConfiguration.config file, ensure that the server name is pointing to the correct server at the <LocalSQLServer> and username/password <DBPassword>, <DBUserName> are correct.

3. If you need to switch from HTTP to a secure communication protocol, you must define an SSL certificate.

Either create and install a new self-signed certificate or use the existing one.

Client-side configuration

1. Open the `...\Program Files (x86)\TIS\eFlow 6\Bin\ConfigSources` folder.
2. Rename the existing BindingSection.config and ClientSection.config files to *.config.temp.
3. Change the extensions of the relevant sample configuration files (either *.config_http or *.config_https) to .config.
4. Replace all occurrences of the default server name with a fully qualified server name in the new configuration files.
5. Change the value at the <communicationprotocol> in the TisConfiguration.config file to point to the correct protocol (http\s).

Troubleshooting

This section describes some common issues and HTTPS validations.

Common symptoms

The following symptoms can indicate that the communication protocol is not configured correctly:

- The eFlow server-side is installed properly, but clients are not working.
- The station splash screen is displayed, but the station is not loaded.
- The Windows Event Log contains errors with only "invalid parameter" messages that do not provide any detailed information.

Sometimes it is not easy to understand exactly what went wrong during the installation. We recommend beginning with the standard procedure, as described in the Troubleshooting checklist.

HTTPS validations

Verify the HTTPS configuration.

SSL certificate validity

1. Select the relevant SSL certificate in the server certificates view in IIS manager.
2. Open the **General** tab and check that this certificate has the corresponding private key.

HTTPS binding configuration for Tis Web Site

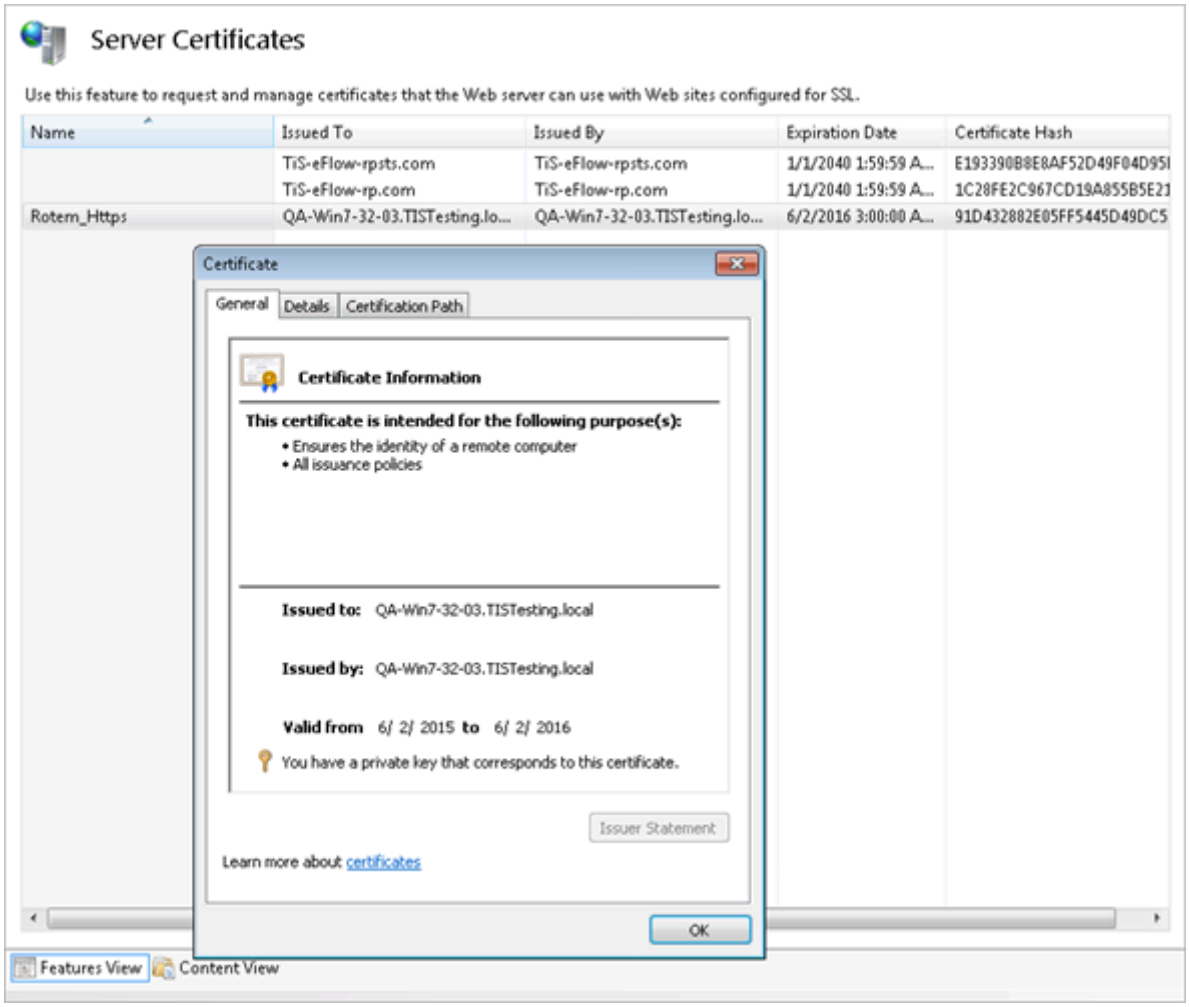
1. Open the **Tis Web Site** bindings dialog in the IIS manager.
2. Open the HTTPS binding and try to reselect the SSL certificate defined for this binding.

An error message appears if the certificate is not valid or not installed correctly. In this case, you must create and install a new self-signed certificate. See Create a new self-signed certificate manually.

HTTPS configuration server name

The HTTP configuration accepts <ServerName>. But HTTPS configuration requires <FullServerName> instead of <ServerName> where the <FullServerName> is the full name of the server machine for which the certificate was issued.

T o obtain the full name of the server, open the HTTPS certificate on IIS as shown below.



Useful utilities and methods

Fiddler

We recommend using the Fiddler utility tool for troubleshooting eFlow communication problems. For more information about the Fiddler tool, refer to "Fiddler" on the Telerik website.

<ServiceSecurityAudit> configuration element

This service behavior specifies settings that enable the auditing of security events during service operations. When auditing is enabled, successful, failed, or both authentication attempts are audited. The events are written to one of three event logs: application, security, or the default log for the operating system version. You can view the event logs using Windows Event Viewer.

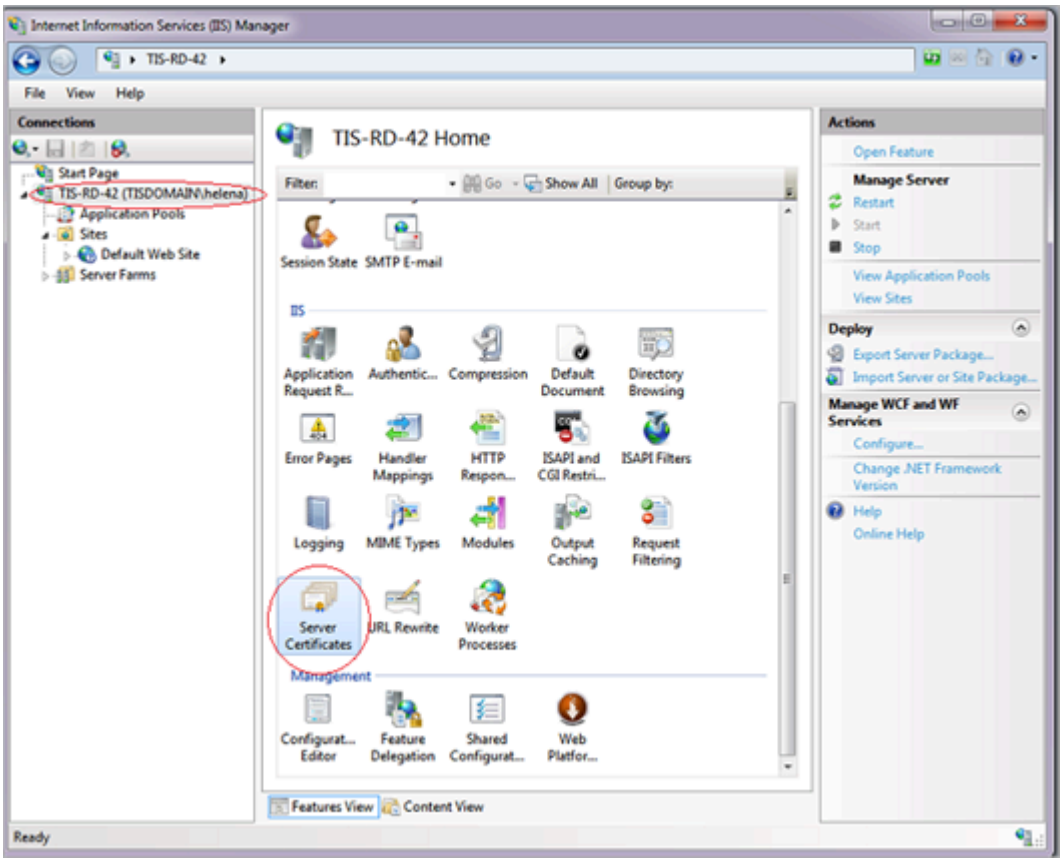
For more information, refer to "<ServiceSecurityAudit>" on the Microsoft website.

Create a new self-signed certificate manually

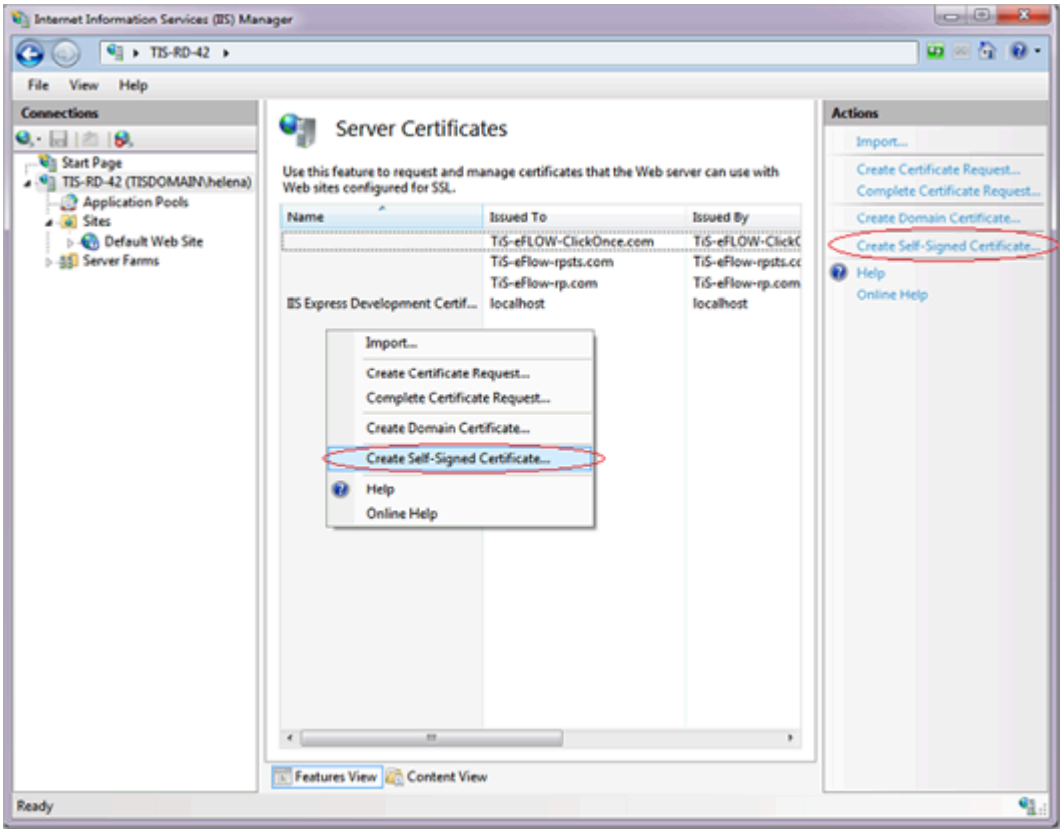
Sometimes the self-signed certificate must be replaced by a new one. For example, the self-signed certificate created automatically by the installation appears to be corrupted.

To create a new self-signed certificate manually:

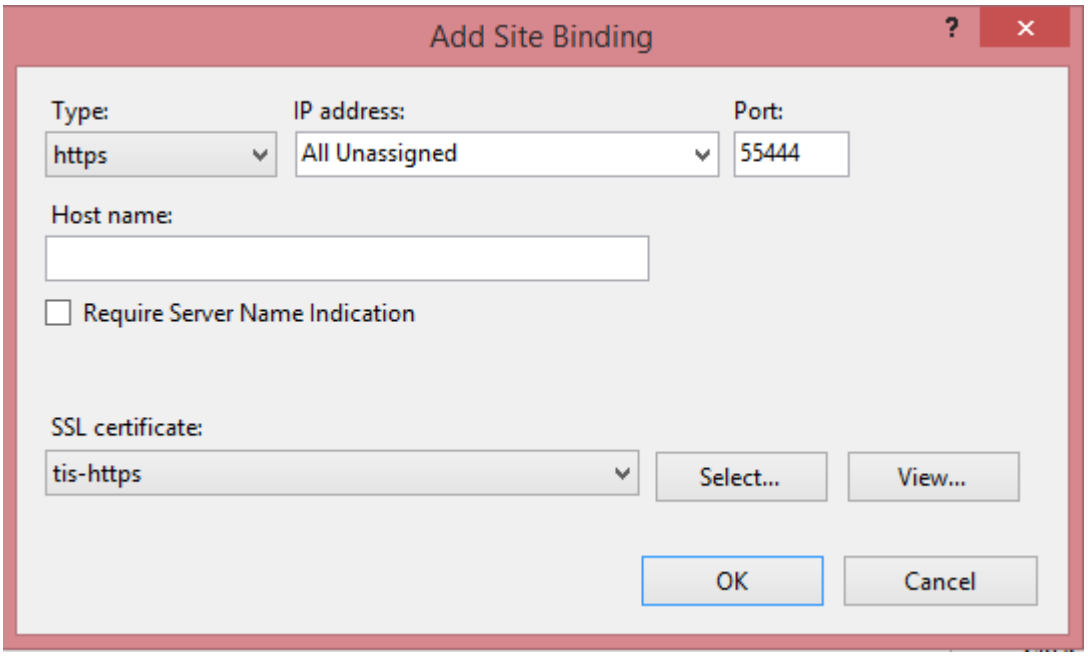
1. Open the Internet Information Services (IIS) Manager.
2. Select the **Server Certificates** feature on the local host.



3. Create a new self-signed certificate.



4. Add a new HTTPS binding to **TIS Web Site**.



Configure a firewall

This section describes the communication ports that need to be configured for proper functioning of eFlow and related third-party components.

eFlow HTTPS port

By default, you need to open port 55444, unless the port number was changed during the eFlow server installation.

eFlow HTTP port

By default, you need to open port 55222, unless the port number was changed during the eFlow server installation.

SQL server connection

By default, SQL clients use port 1433.

For more information, refer to "Configure Windows Firewall to Allow SQL Server Access" on the Microsoft website.

MSDTC

Open port 135 (the RPC Endpoint Mapper port) on both SQL and eFlow server machines.

For more information, refer to "Configuring Microsoft Distributed Transaction Coordinator (DTC) to work through a firewall" on the Microsoft website.

Change the port number

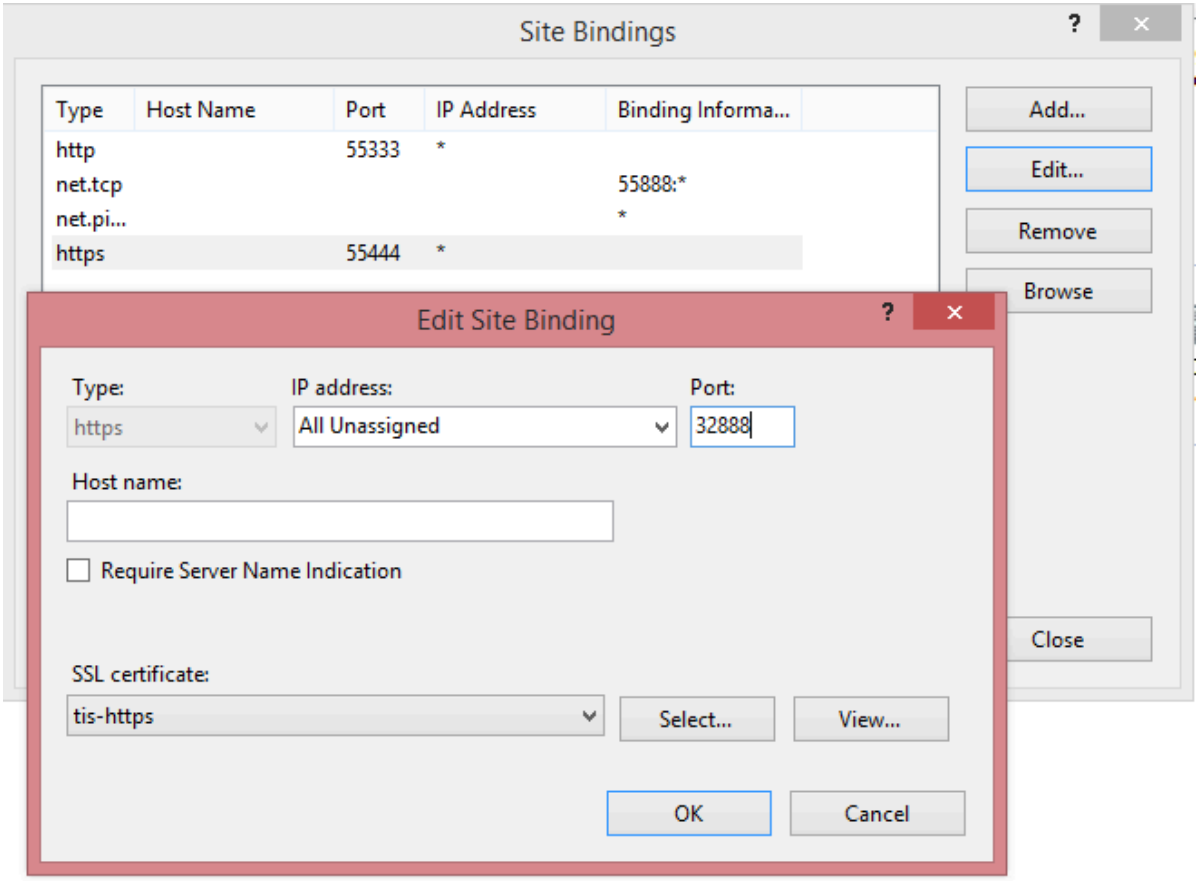
This section describes the system-wide changes required if the default eFlow communication port number needs to be changed after installation.

No additional actions are needed if the required communication port was selected during the installation. The installer will configure everything automatically.

Server-side configuration

To change the TiS Web Site HTTPS port number selected during the eFlow installation:

1. Open the IIS Manager on the eFlow server machine.
2. Go to **Sites** > **TiS Web Site**.
3. Select **Edit Bindings** from the right-click menu.
4. Select **Edit** to change the HTTPS port number.



5. Restart IIS.

Client-side configuration

Modify the client configuration as per the server settings.

The client configuration must be changed in accordance with the server settings.

1. Back up `C:\Program Files (x86)\TIS\eFlow 6\Bin\ConfigSources\TisConfiguration.config` and then open it for editing.

   In the <CommunicationPort> property, replace 55444 (the default eFlow HTTPS port number) with the required port number.

2. Back up `C:\Program Files (x86)\TIS\eFlow 6\Bin\ConfigSources\BindingSection.config` and then open it for editing.

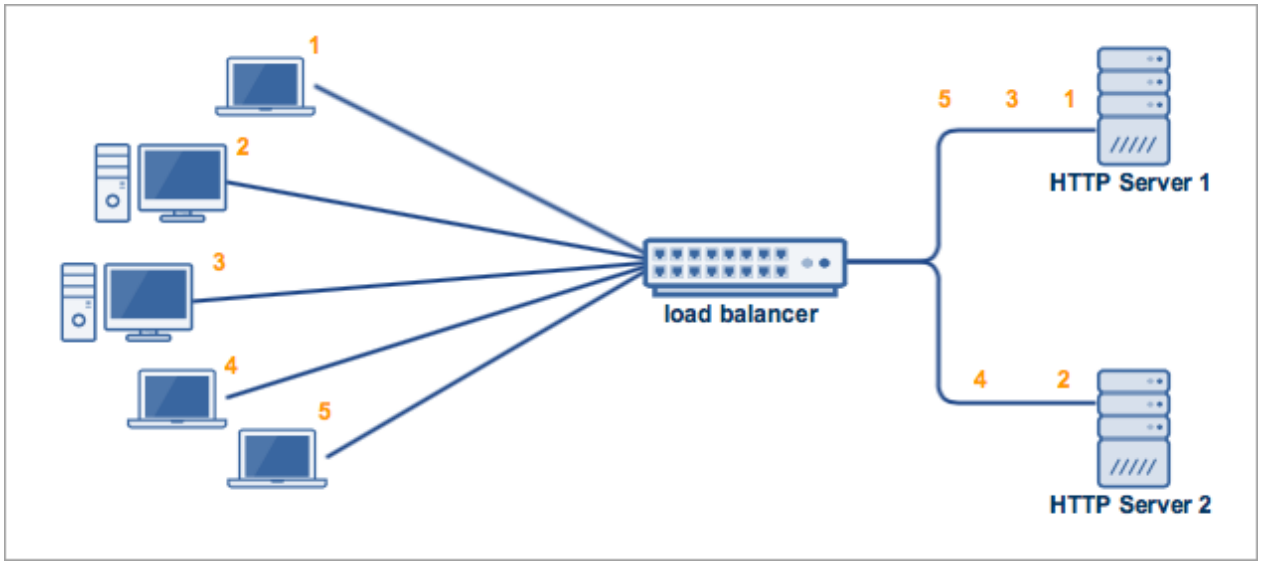
   Replace all instances of 55444 (the default eFlow HTTPS port number) with the required port number.

3. Back up `C:\Program Files (x86)\TIS\eFlow 6\Bin\ConfigSources\ClientSection.config` and then open it for editing.

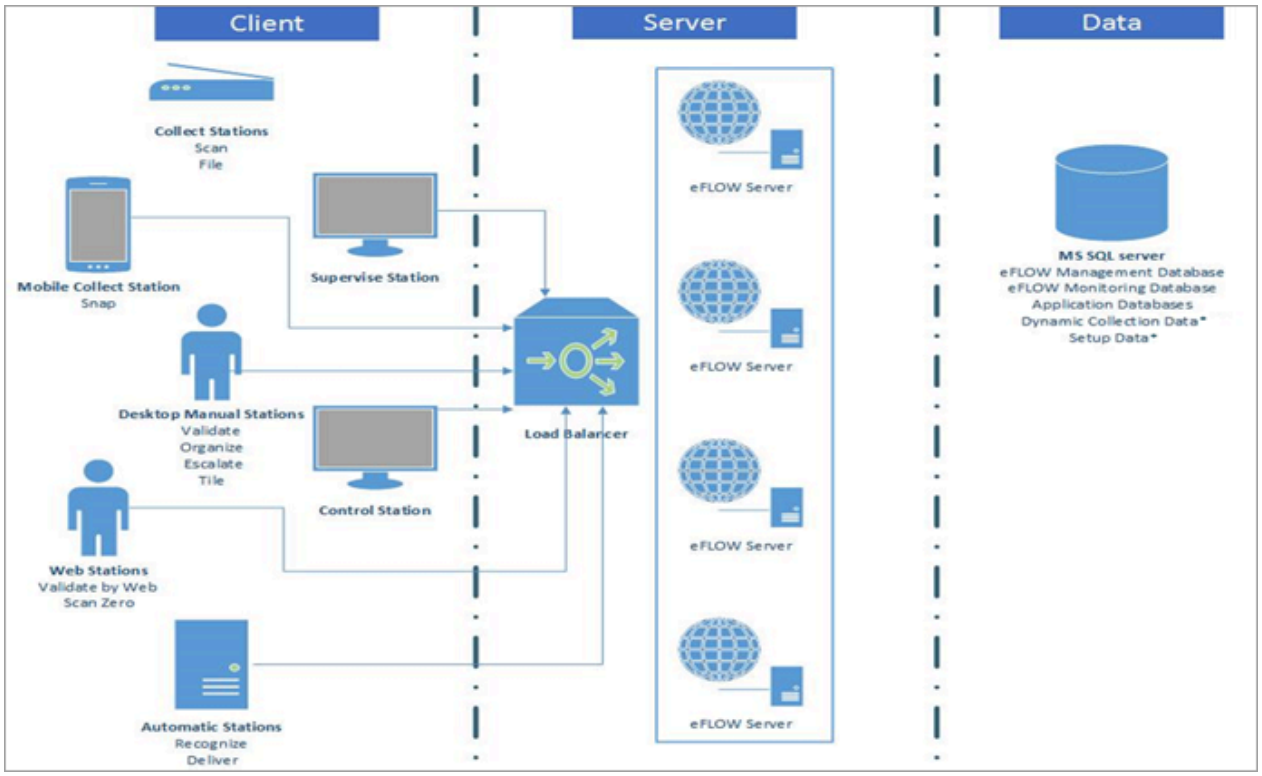
   Replace all instances of 55444 (the default eFlow HTTPS port number) with the required port number.

Cluster environment setup

This section provides step-by-step instructions on how to configure eFlow servers in an active-active cluster environment.



If you need to balance the load of the traffic between the eFlow server and multiple clients due to many clients, heavy data transfer load, and a large number of applications, or to improve performance, or any other such reasons, you may require more than one eFlow server. Then the following configuration is necessary.



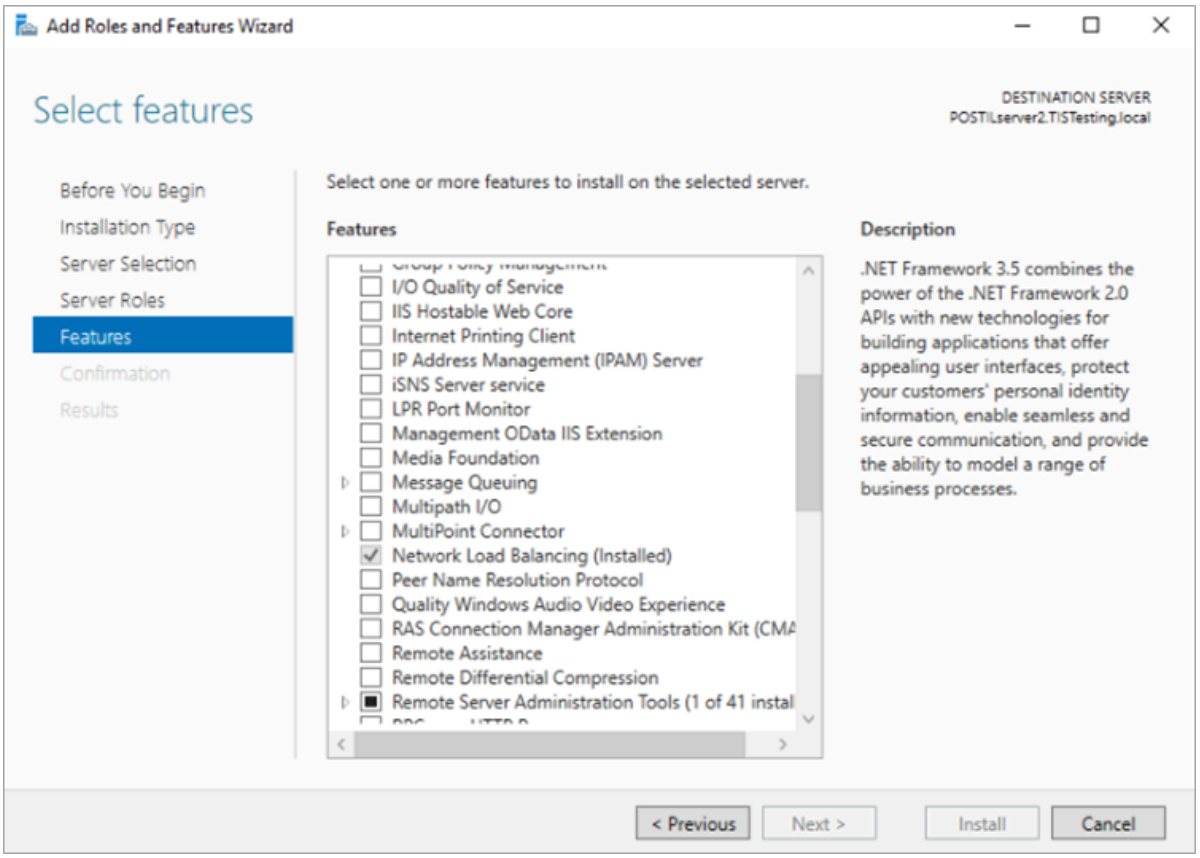
In this type of environment, multiple eFlow servers manage communication with the clients and the data layers. To work in such an environment, apart from setting up the eFlow server, and clients and data storage, you must set up a network load balancer on each server.

A load balancer diverts the communication between the clients to the vacant eFlow server at any moment.

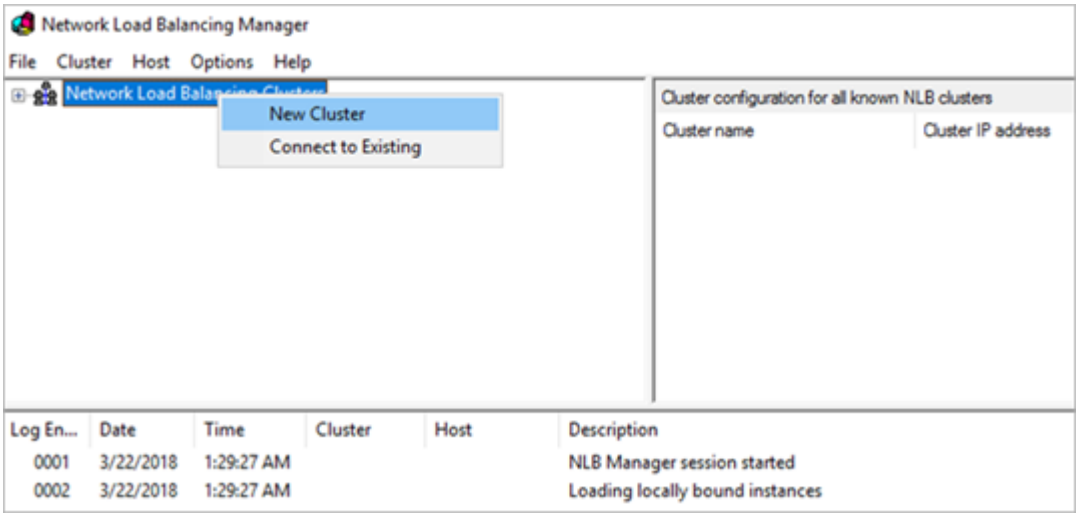
eFlow supports any type of load balancer: software or hardware.

Configure the first eFlow server

1. If the Network Load Balancing Manager is not installed, in the **Add Roles and Features Wizard**, on the **Features** list, select **Network Load Balancing**.

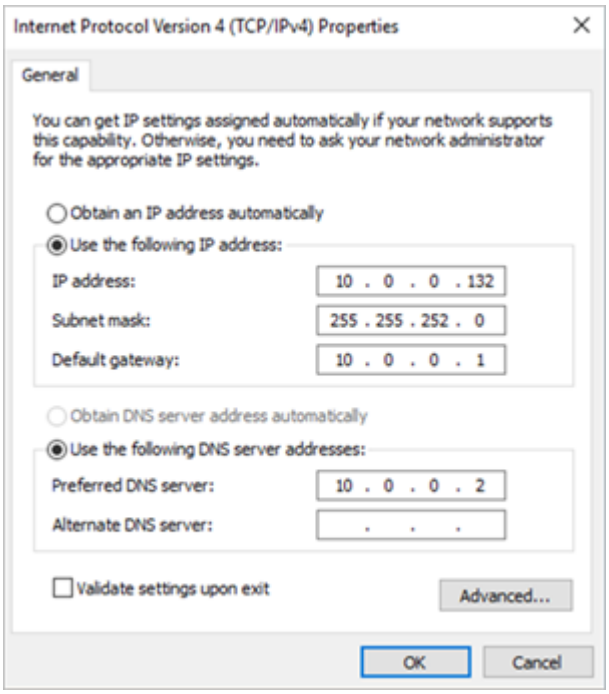


2. Open the Network Load Balancing Manager, right-click on **Network Load Balancing Cluster** and select **New Cluster**.

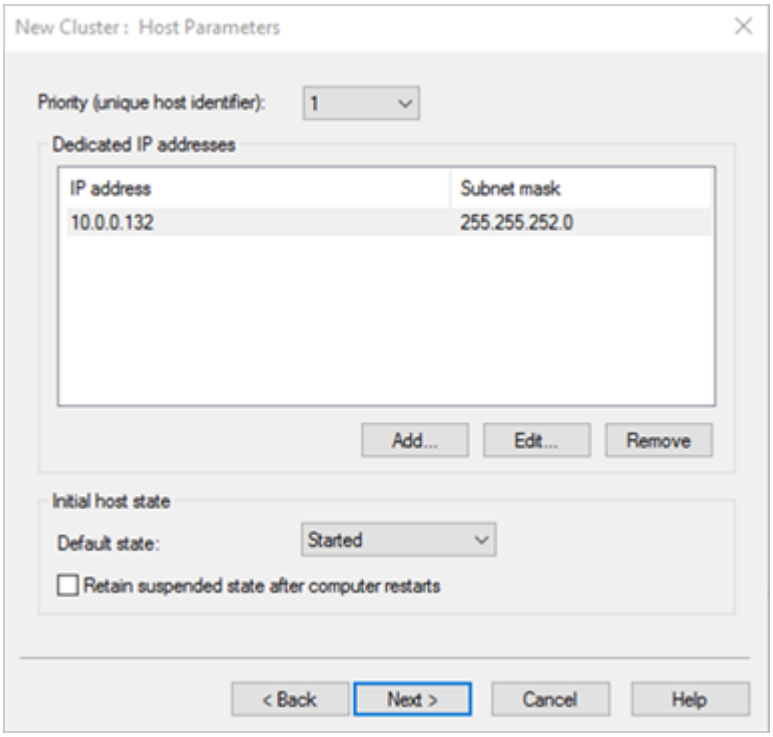


3. On the **New Cluster: Connect** screen, in the **Host** field, enter the IP address of the server itself. This IP address must be static.

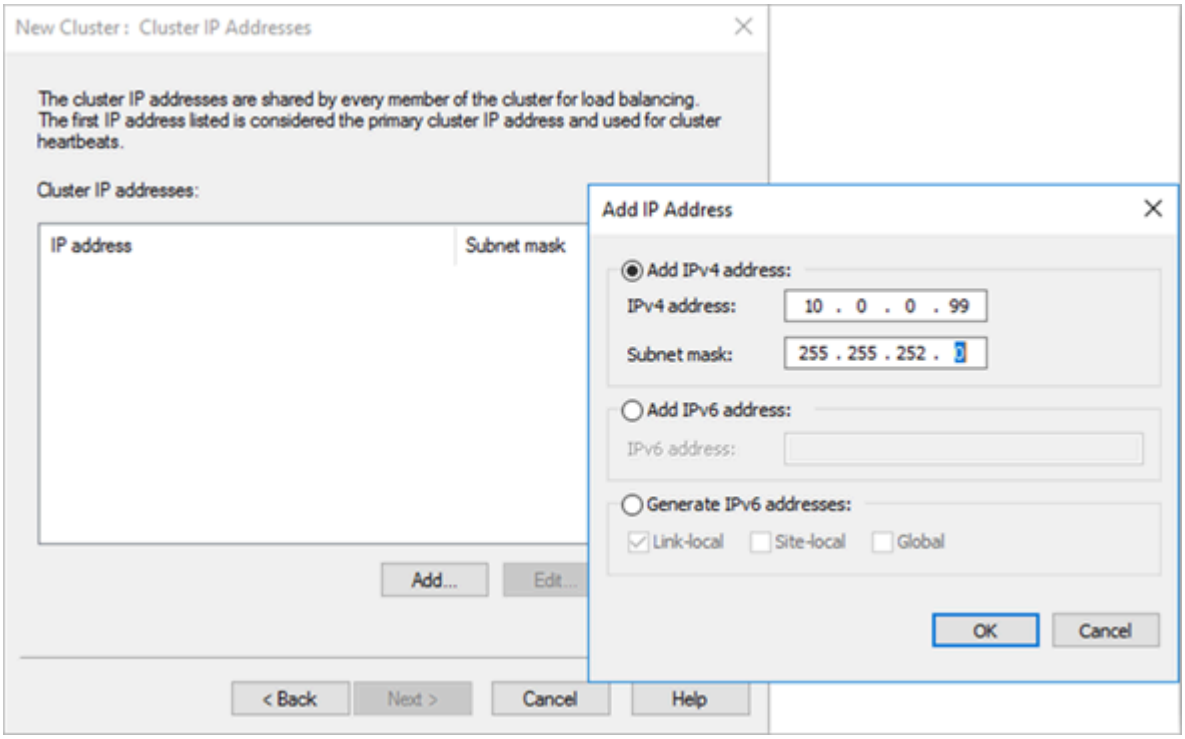
4. Select the network adapter from the list that was defined with the spoofing configuration, then click **Next**.
5. On the **Host parameters** screen, click **Add**.
   The **Internet Protocol Version 4 (TCP/IPv4) Properties** screen appears.



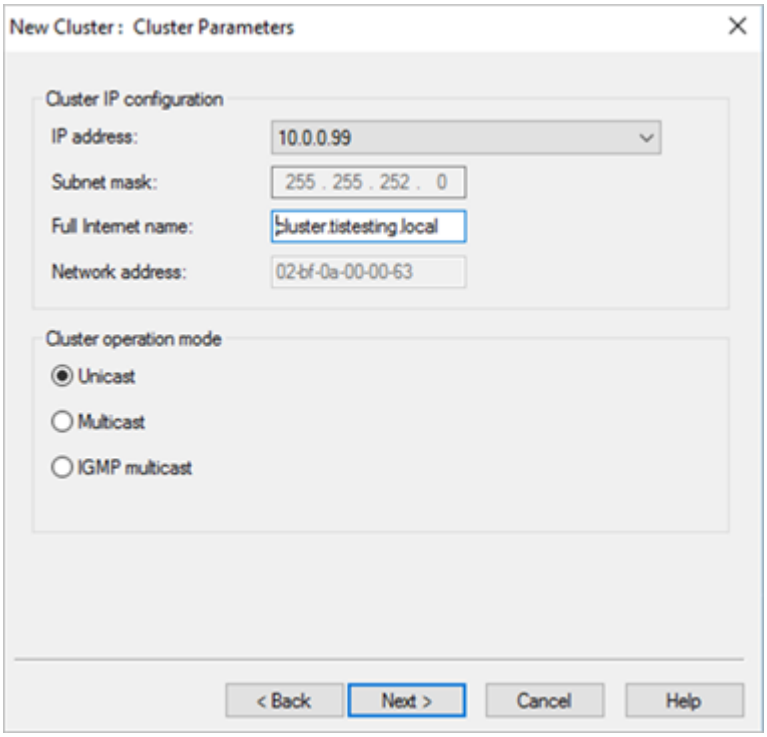
6. Enter the **IP address** and the **Subnet** mask, and then click **OK**.
   On the **Host Parameter** screen the added IP address is displayed.



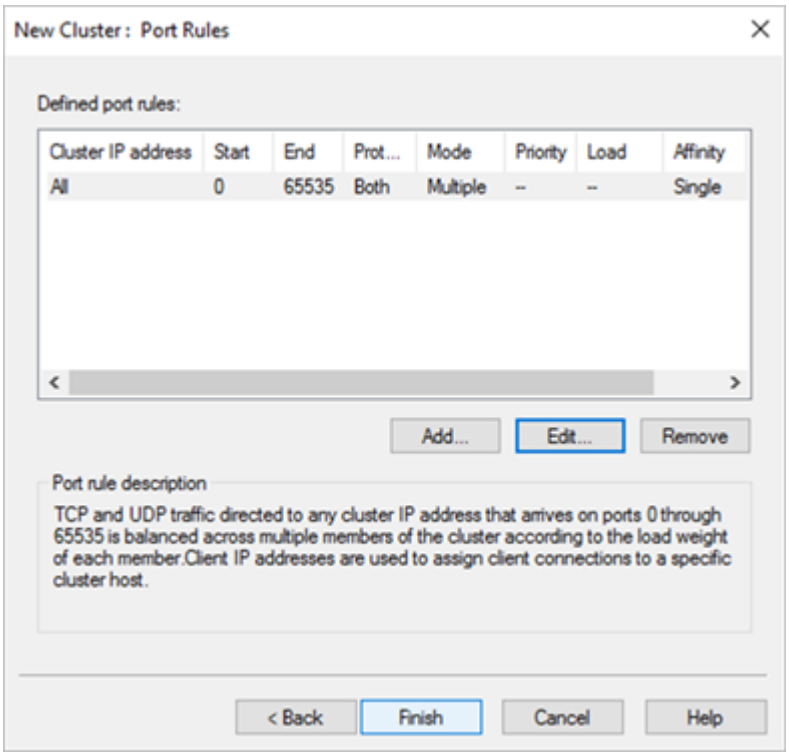
7. Click **Next**.
8. On the **Cluster IP Addresses** screen, click **Add** and enter the static IP address of the cluster server.



9. In the **Cluster Parameters** screen, enter the **Full Internet name** and click **Next**.



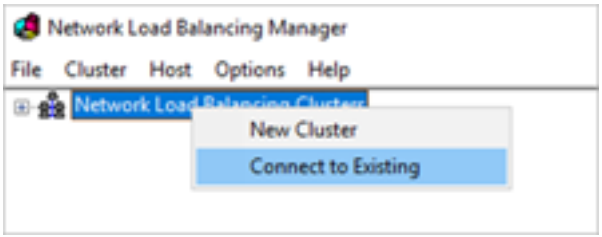
10. In the **Port Rules** screen, click **Add** and configure the properties as needed.



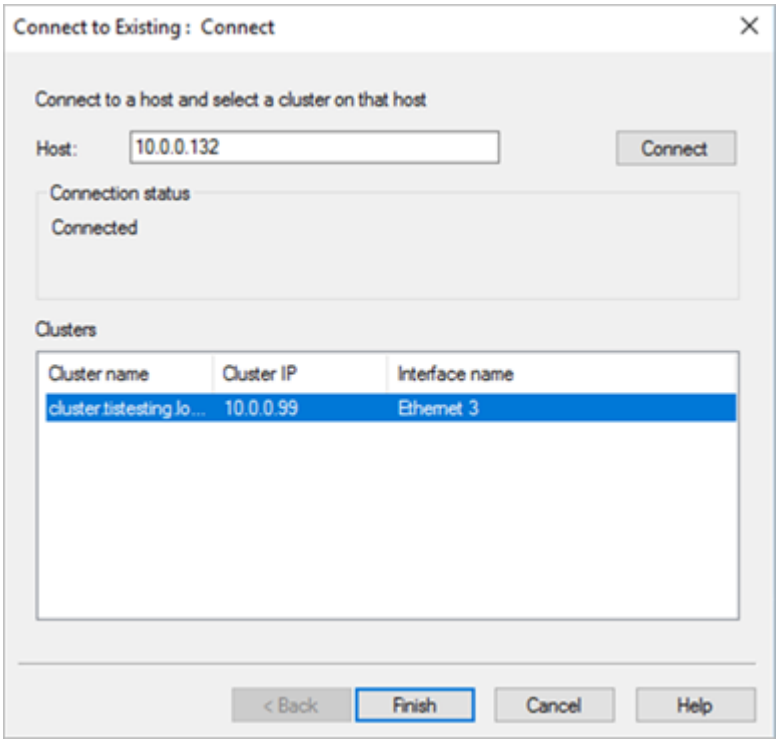
11. Click **Finish** to complete the configuration of the first host and cluster server.

Configure the second eFlow server

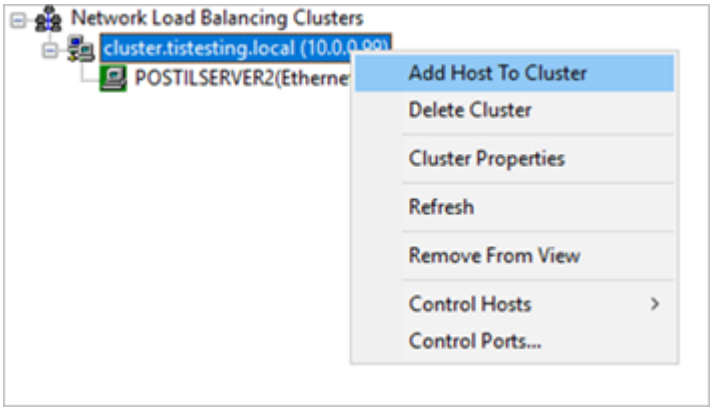
1. To connect to the cluster that you just defined, right-click on **Network Load Balancing Cluster** and select **Connect to Existing**.

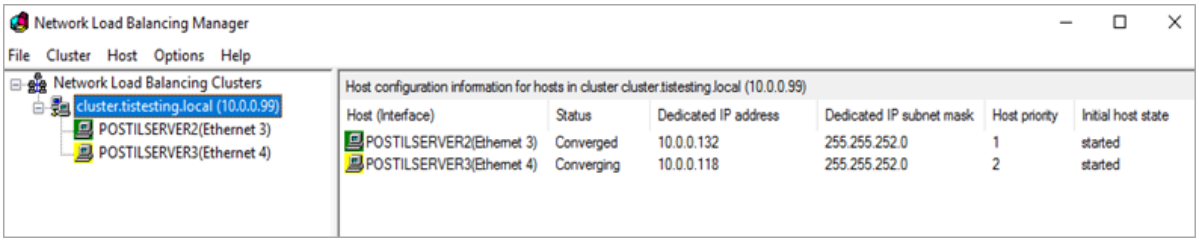


2. On the **Connect to Existing: Connect** screen, in the **Host** field, enter the IP address of the first eFlow server that you have defined in the preceding section.



3. On the **Clusters** list, select the cluster you just defined, then click **Finish**.
4. You must now link the second server to the cluster environment as an active-active configuration. Right-click on the cluster server name and select **Add Host To Cluster**.





5. Wait for the status to change to **Converged**.

Storage

Use an SQL server for storage.

1. For the eFlow server and STS server, in the TISConfiguration.config file, set the UseSqlDynamicStorage parameter to **True**. The TISConfiguration.config file is available at the following locations.
   - `C:\inetpub\wwwroot\Tis Web Site\eFlow_6\Bin\ConfigSources`
   - `C:\inetpub\wwwroot\Tis Web Site\TisDefaultSTS\Bin\ConfigSources`
2. Ensure that all eFlow servers are configured to point to the same SQL server by configuring the following parameters in the TISConfiguration.config file.
   - <DBPassword></DBPassword>
   - <DBUserName></DBUserName>
   - <IntegratedSecurity></IntegratedSecurity>
   - <LocalSQLServer></LocalSQLServer>

Software

You need an active eFlow license certificate for eFlow servers and load balancer (if you are using it).

License

eFlow license is required with the same number of activations as the number of eFlow servers in the environment. Each eFlow server must be activated.

Certificates

Export the certificate from the load balancer and the eFlow servers and import it on each client.

1. Open the Windows Manage Computer Certificates tool.
2. Browse to **Trusted Root Certification Authorities > Certificates**, find the certificate with "Issued to" as the local machine name, and export it.
3. Import the certificate to each client.
4. Repeat steps 1 and 3 for the certificates of each eFlow server.

Additional Windows configuration setup is not required for the eFlow cluster environment.

Configuration

Clients

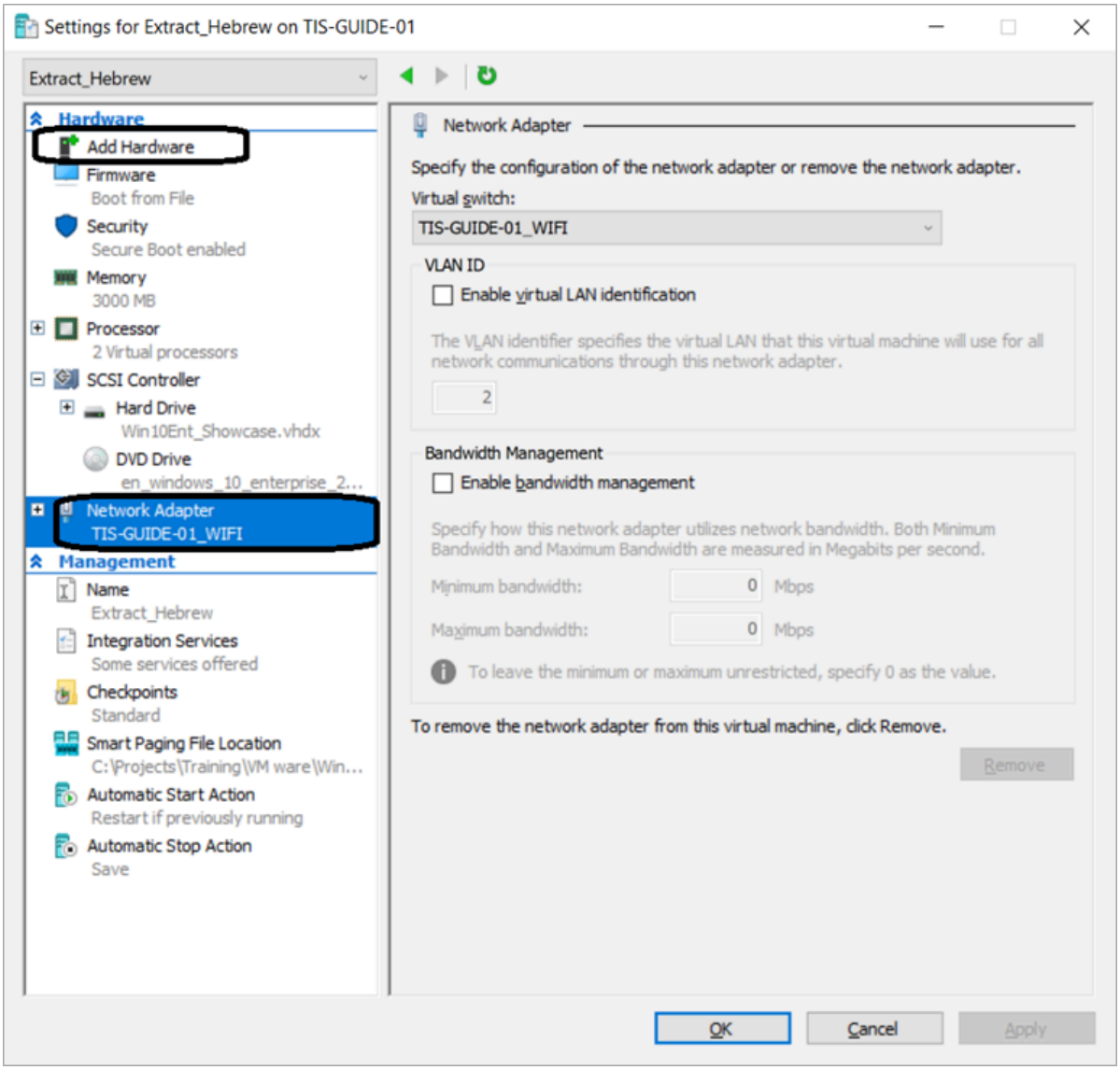
Configure all clients to point to the load balancer as the eFlow server.

1. Open the client configuration file (TISconfiguration.config) available in the following folder:
   `…\TIS\eFlow 6\Bin\ConfigSources`

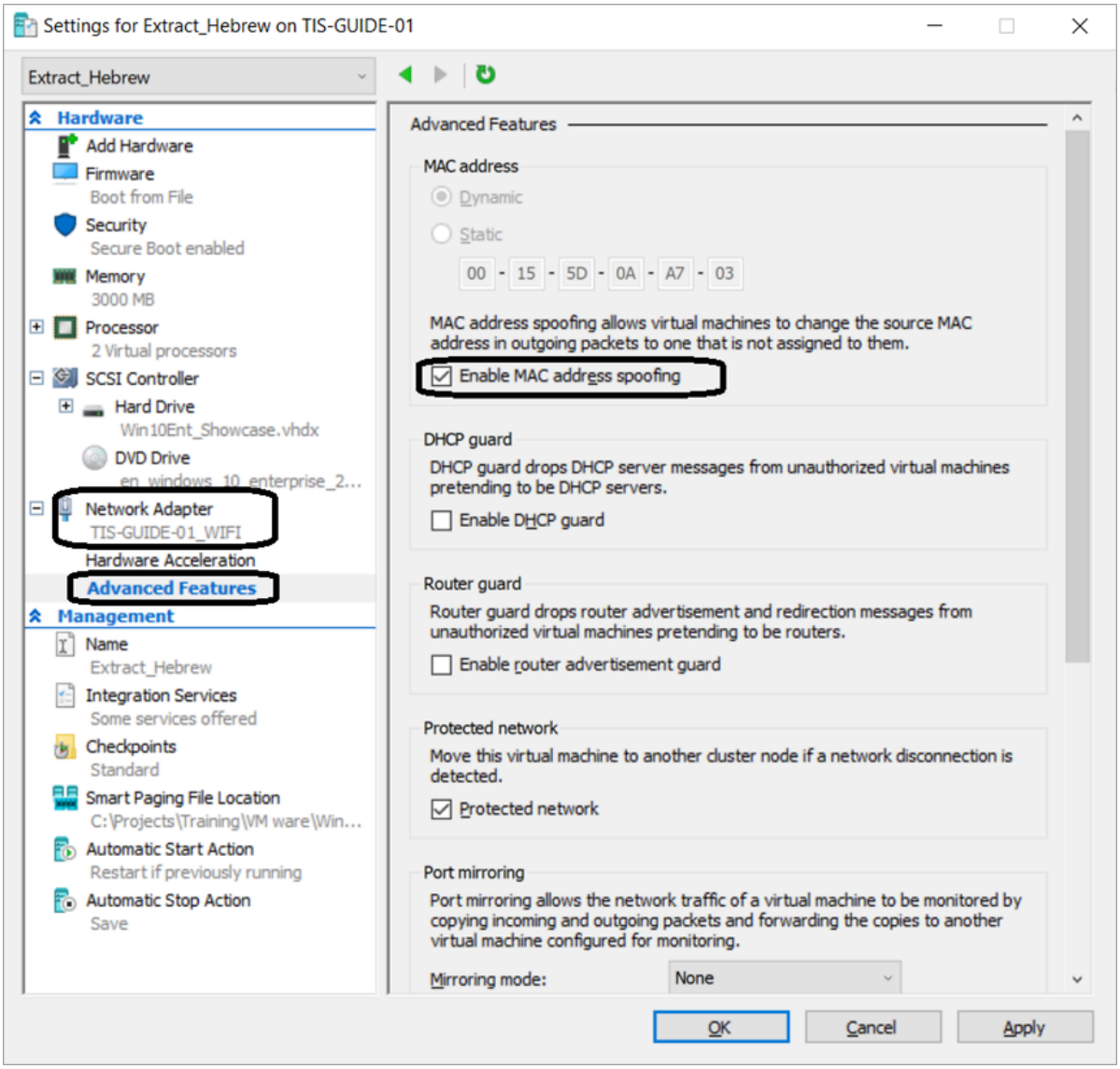
2. Add the load balancer machine name to the ServerMachineName property:
<ServerMachineName>Load balancer machine name</ServerMachineName>

Server

1. On each server, add a second adapter of the **Network Adapter** type.



2. On each server, in the first **Network Adapter** instance, go to **Advanced Features** and select **Enable MAC address spoofing**.



Change the eFlow location

This section explains how to change the eFlow location after installation.

Move the eFlow server to a non-default location

By default, the eFlow server (eFlow_6 web application) is installed in the `c:\inetpub` folder.

Use the following procedure to move the eFlow server to another location after installation.

1. Move the contents of the TIS Web Site to another location.

2. Change the physical path of the TIS Web Site using the IIS Manager.
3. If you have moved your inetpub folder to another drive, you also need to change the following registry value:
   `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\InetStp\PathWWWRoot`

Move the application data to another location

The path to the <AppData> folder is usually set during the installation.

Use the following procedure to change the application data location after the installation.

eFlow server machine

1. Make sure no eFlow stations are running.
2. Stop the **T.i.S. eFlow Autorun Starter Service** if it is installed and enabled on the server machine.
3. Open the IIS Manager and stop **TisAppPool**.
4. Copy the application data to a new location.
5. Change the <eFlowDataPath> value in `C:\inetpub\wwwroot\Tis Web Site\eFlow_6\Bin\ConfigSources\TisConfiguration.xml`.
   This will be the new eFlow server dynamic data path.
6. Make sure that the following users have read and write access permissions for the new location:
   - **TisAppPool** user. By default, this is **IIS AppPool\TisAppPool**, but it can be any other user-defined identity.
   - **T.i.S. eFlow Autorun Starter Service** user if you are running the autorun stations on the eFlow server machine.
   - Windows users if you run some clients manually on the eFlow server machine.

eFlow client machine

1. Make sure no eFlow stations are running.
2. Stop the **T.i.S. eFlow Autorun Starter** service if it is installed and enabled on this machine.
3. Copy the application data to a new location.
4. Change the <eFlowDataPath> value in the client basic configuration file available by default at the following location:
   `C:\Program Files (x86)\TIS\eFlow 6\Bin\ConfigSources\TisConfiguration.xml`.
   This will be the new eFlow client dynamic data path.
5. Ensure that the following users have read and write access permissions for the new location:
   - **T.i.S. eFlow Autorun Starter** service user if any autorun stations are running on this machine.
   - Windows users if you run some clients manually on this machine.

Do not remove the original <AppData> folder. It will not be used anymore at runtime but should remain as it was created by the installation. For more information, refer to "Understanding Windows Installer (WSI) self-repair" on the Microsoft website.

Chapter 2

eFlow licensing

This chapter describes the eFlow licensing mechanism, the set of licensed features, and working procedures for license installation and activation.

Licensing mechanism

This section describes the most important concepts related to the eFlow licensing mechanism. To learn about the license installation and management procedures, see License setup and management.

eFlow servers and licenses

An eFlow license is issued for a certain amount of eFlow servers and should be activated by the eFlow License Activation server.

The number of licensed features available in a multi-server environment depends on its architecture.

Every eFlow management database is connected to one license instance. As a result, in cluster environments, all eFlow servers share the same licensed features, while each standalone installation can use the full feature set. See Licensing models for common eFlow server configurations for details.

License activation

You can register an eFlow license with Kofax either online or via an email service. Each fully functional eFlow license is available for seven days after its installation. To continue working beyond this initial period, you must activate the license.

Each eFlow license contains a value called Number of Concurrent Activations that defines the number of eFlow servers that can concurrently use the license.

License lifecycle

1. Submit the eFlow license request via the Web License Request System.
2. Receive an automatically generated license file via email.
3. Install the license file using the Administrate module.

4. Activate the license separately for each server. The total number of allowed concurrent activations is set according to the license parameters.

Validate the license at runtime

Each eFlow station performs the runtime license check when it logs in. The license service checks that the active license contains the required station licenses of this type are still available.

For example, every Recognize station requires an AutoStations license. If the active license allows 10 AutoStations and 9 stations are already running, then only one more station of this type can be started.

Many stations use some service licenses in addition to the station license. If the required service license is not found, the station continues to run, but the service is not be loaded, and the functionality provided by this service is not available.

Licensed features

Stations

All eFlow stations are divided into several groups, as shown in the following table.

One license type can serve different stations. For example, you can run many Validate stations using the same Manual Stations license.

Type	Stations	Notes
Input Stations	Collect	PixTools_NET is added automatically.
Scan Stations	Scan	PixTools_NET is added automatically.
Auto Stations	Run	
	Recognize	The Recognize station replaces the FreeMatch, Processing, Form ID, PageOCR, and Smart Categorizer stations. It is necessary to add one of the service licenses to provide the functionality needed.
	Collect Free	
	Build Free	
Manual Stations	Validate	
	Web Validate	

Type	Stations	Notes
	Organize	
	Escalate	
	Tile	
Export Stations	Deliver	It is necessary to add one of the service licenses to provide the ERP export functionality if needed.
	Deliver2ERP	
Control Stations	Control	
Supervise Stations	Supervise	
Develop Stations	Design	
	Design Smart	
	Design Free	
	Analyze Free	
	Analyze Recognition	
	Test OCR Page	
	View Learning	
Custom Stations	Custom	

Services

Services provide specialized functionality and are licensed as separate features.

Typically, services are used together with station licenses. For example, to obtain a license for the Recognize station with Freedom functionality, you should request the AutoStations license combined with the efFreedomService license.

Services	Notes
Check Reader	
Integra Service	
Freedom Service	OP_JUSTICR added automatically.
Smart Service	
ERP Export Oracle	
ERP Export SAP	

Page counter

Page Counter is a special license type based on the number of physical pages that are allowed to be acquired by the system.

License	Notes
Page Counter	The total page count increases when a new collection is created.

Third-party licenses

These licenses are issued by third-party software providers and should be used together with one of the proprietary eFlow licenses.

License	Notes
Exorbyte	Used in lookup tables based on the Exorbyte search engine. Can be used in manual or automatic mode.
PixTools_NET	Used when additional image enhancements are required by the custom code. It is not necessary to request this license separately for Input and Scan stations, as it is included automatically.

Field OCR/ICR engines

The Abbyy engine will be deprecated in a future Kofax eFlow release.

Company	Engine	Additional attributes	Notes
AToIA	AToIA		
Abbyy	ABBYY	None	
		CJK	Chinese, Japanese, and Korean languages.
		HEB	Hebrew language.
		CJK|HEB	
JustICR	CharacTell	25	Additional attributes refer to different OCR engine speed values.
		250	
		400	
Expervision	Expervision		
IRIS	IRIS		
Kadmos	Kadmos		
Oce	Oce	25	Additional attributes refer to different OCR engine speed values.
		50	
		100	
		150	
		250	
ScanSoft	OmniPage	None	

Company	Engine	Additional attributes	Notes
PenPower	PenPower		
Ricoh	Ricoh Machine		
Ricoh Hand	Ricoh (Hand)		

Barcode recognition engine

Company	Engine	Additional attributes
ClearImage	Inlite	
QualitySoft	QualitySoft	None
		2DBarcode
ScanSoft	OmniPage	1DBarcode
EMC Captiva	PixTools_1D_Barcode	
	PixTools_DataMatrix_Barcode	
	PixTools_PDF417_Barcode	

Page recognition engines

The Abbyy engine will be deprecated in a future Kofax eFlow release.

Company	Engine
Abbyy	Abbyy
	Abby Japanese Page
	Abby Hebrew Page
Expervision	Expervision
Oce	OCE Page
ScanSoft	ScanSoft Page
	ScanSoft ICR Page
PenPower	PenPower Page
Ricoh	Ricoh Page

Examples of typical license requests

Required functionality	Licenses to be requested
Integra recognition station	Auto Stations Integra Service

Required functionality	Licenses to be requested
Freedom recognition station	Auto Stations Freedom Service
Smart	Auto Stations Smart Service The JustICR license is added automatically.
ERP Export with SAP	Export Stations Export Destination SAP

License setup and management

You can install and manage licenses in the eFlow Administrate module.

1. In the **eFlow LaunchPro**, select **Administrate** from the **Tools** list.
2. In **Administrate**, right-click on **Domain Configuration** and select **License**.
   The **License Management** dialog box appears.

Read eFlow licensing mechanism before beginning the license installation.

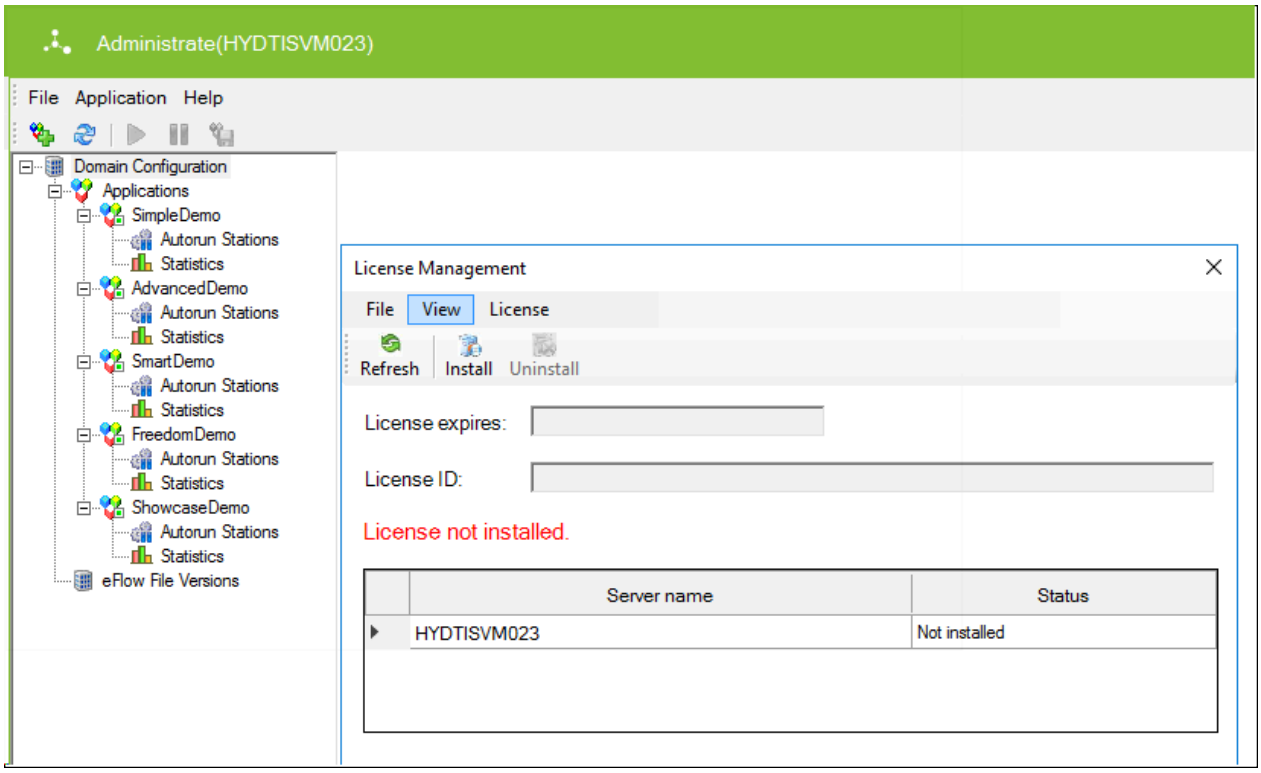
Create a license request

Customers and partners can contact Kofax eFlow sales representatives to request a license.

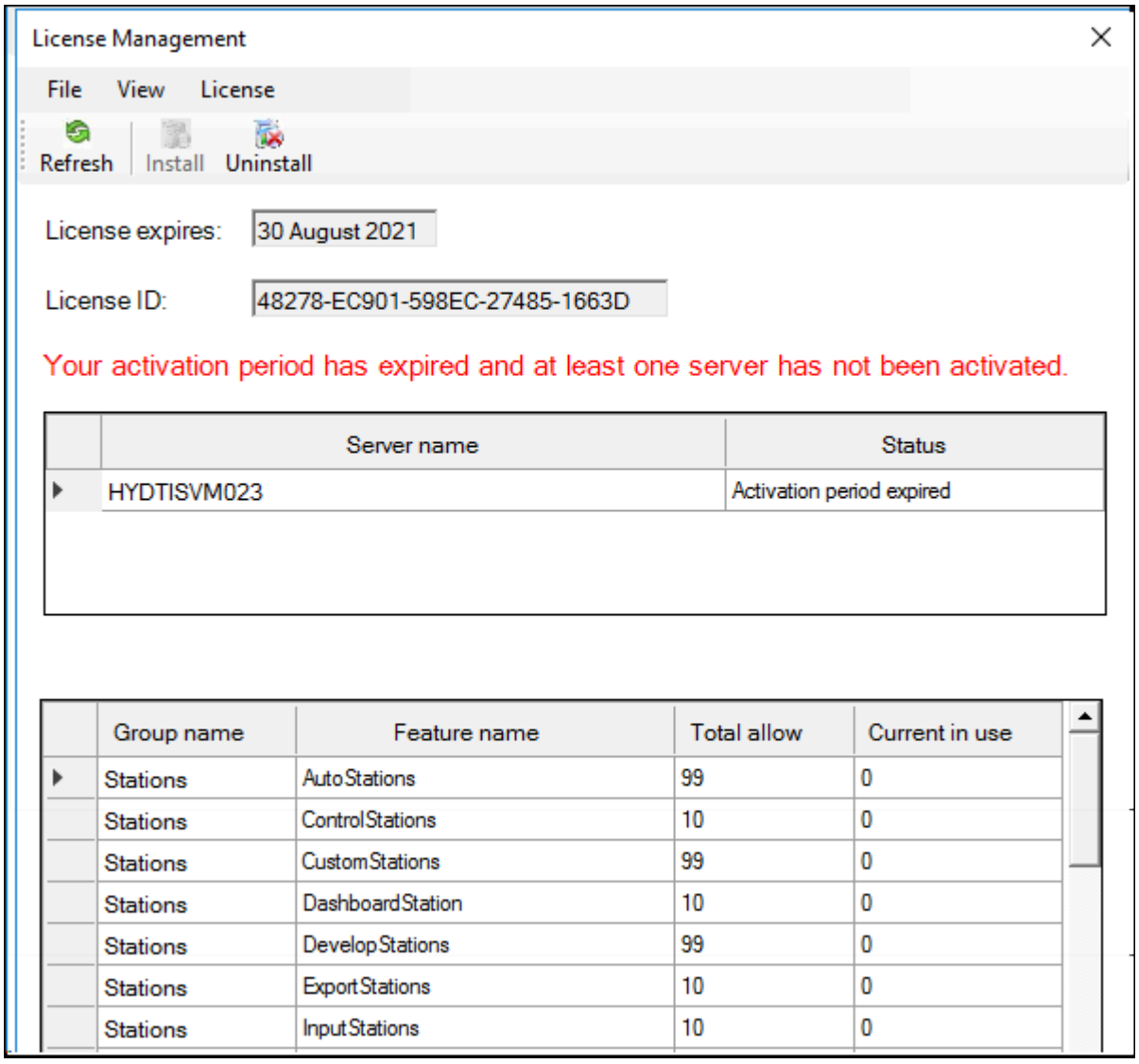
Once the license is successfully issued, you will receive the license by email. You must install and activate the new license.

Install a license

1. In the **License Management** dialog box, click **Install**.
2. Enter the license expiration date and the license ID.



After the license installation, the **License Management** dialog box displays all available eFlow servers with the status **Not activated**. Every eFlow license is fully functional for seven days after its installation. To continue working beyond the trial period, you must activate the license.



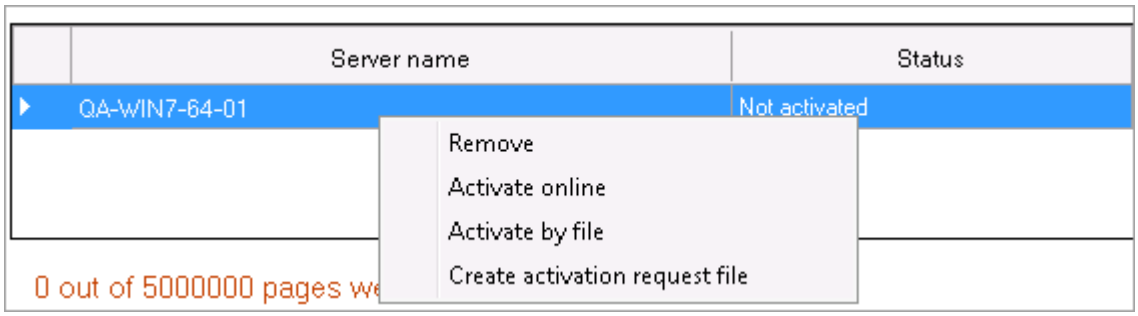
Activate a license

A new license is fully functional for seven days following its creation beyond which you must activate it. During the activation, the eFlow servers that belong to the current license will be registered on the eFlow License server.

After successful activation, the **License Management** dialog box displays information about the installed license and active eFlow servers. The expiration date is set automatically as defined in the license.

In a multi-server configuration, repeat the activation procedure for each eFlow server.

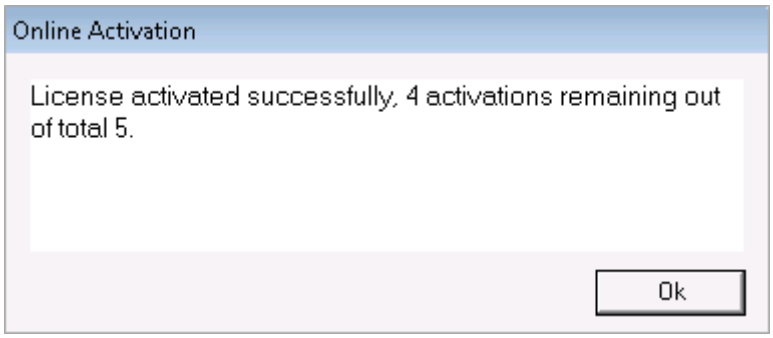
To activate a license, in the **License Management** dialog box, select one of the activation methods from the **License** menu, or right-click on the server name and select one of the activation methods.



Activate online

We recommend using online activation if an Internet connection is available. This is the simplest method.

Allow about half a minute for the procedure to complete. A dialog box displays the results of the activation attempt.



Activate by file

Offline activation is achieved via email communication, as shown in the following figure.

To activate the license by file perform the following steps:

1. Select the menu item **Create activation request file** to create an activation request file (.ACT_REQ).

2. Send the newly created activation request file to the automatic license activation email address: eFlowActivation@kofax.com.

   You will receive a reply that contains the license activation file (.ACT). Save this file to use later during the activation.

3. From the **License** menu, select **Activate by file**.
4. Open the saved activation file.

Deactivate a license

License deactivation is required under the following circumstances:

- One of the eFlow server machines will be replaced.

  An active eFlow server must be deactivated before removing it.

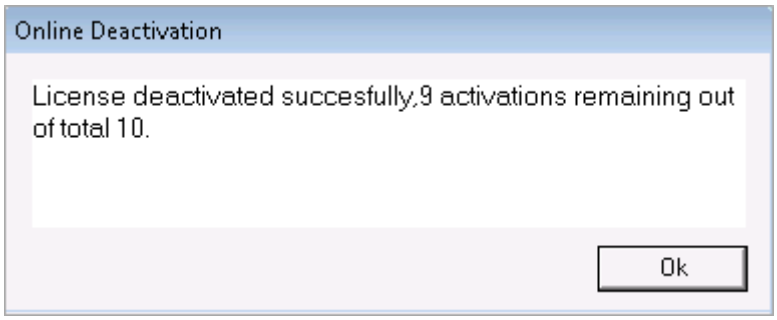
  As the activation number is limited per license, failure to deactivate a license will not free the activation in use. If the license has reached its limit, further activations will be refused.
- The eFlow license must be uninstalled.

  Once all the servers are deactivated, the license can be uninstalled and reused if necessary.

Deactivate online

If an Internet connection is available, it is recommended to perform the online deactivation by selecting Deactivate Online from the License menu.

The process should take several seconds, at the end of which a message shows the remaining activation for the license.



Deactivate by file

1. In the absence of an Internet connection, you must select **Create deactivation request file** from the **License** menu. This command will deactivate the server locally and create the deactivation request file.
2. To finish the deactivation procedure, send the deactivation request by email to the automatic license address: eFlowActivation@kofax.com

   You will receive a reply from the above email address containing confirmation that the deactivation has been registered successfully. No further action is required on your part.

Uninstall a license

In the **License Management** dialog box, click **Uninstall**.

The Uninstall button is not available for active licenses. You must deactivate a license before you can uninstall it.

Reinstall a license

1. Deactivate and uninstall the existing license.
2. Install a new license or reinstall the old one.

Servers Configuration

This section describes the most common license management procedures that involve eFlow servers.

Read eFlow licensing mechanism before applying any changes to the server configuration.

Licensing models for common eFlow server configurations

Clustered eFlow servers

In the "clustered eFlow servers" model, all servers share the same eFlow management database.

You can install just one license file on one of the servers and then activate all the clustered servers.

The total amount of licensed features (stations, services, OCRs, and more) used by all the servers may not exceed the number defined in the license. For example:

- The customer ordered one license with two activations: One Input station and one Data Entry station.
- The eFlow servers are installed on two machines with NLB, and the licenses on both servers are activated.
- You can only run one Input station on one machine, and one Data Entry on another machine.

Multiple standalone eFlow servers

In the "multiple standalone eFlow servers" model, each server has its eFlow management database.

You must install and activate the license on each server.

You can use the same license for several eFlow standalone installations if the number of servers does not exceed the total allowed number of concurrent activations

Each environment can use the full set of features defined in the license. For example:

- The customer ordered one license with tow activations: One Input station and one Data Entry station.
- The eFlow servers are installed on two standalone machines, and licenses on both servers are activated.
- You can run one Input station and one Data Entry on each machine.

Add a new eFlow server

A new eFlow server installation is recognized by the license service and the server name is added automatically to the server list in the Administrate module.

If the total number of activated servers is less than stated in the license, you can simply activate the newly installed server.

Once all the available license activations are already used, you can switch to a new eFlow server after removing one of the existing servers using the following procedure.

1. Deactivate one of the existing servers. The active server cannot be removed.
2. Remove the deactivated server.
3. Activate the new server.

Remove an eFlow server

You can remove an eFlow server from the servers list.

1. Open the **License Management** dialog box in the **Administrate** module.
2. Right-click the server you wish to remove.
3. Select **Remove** from the context menu.

Be careful not to remove the single installed server in the standalone eFlow configuration.

eFlow server hardware or software changes

Some common management tasks require special attention. For example:

- Renaming the eFlow server machine.
- Applying hardware changes on the eFlow server.
- Moving the virtual machine to another host.

Before performing any of these tasks, deactivate and remove the server that is subject to change. This will ensure that the license activation runs smoothly afterward.

If for some reason, the customer did not deactivate the eFlow server before performing the management tasks, invalid server code error messages are issued.

Chapter 3

eFlow internals

This chapter provides insight into various aspects of how eFlow functions.

eFlow data

This section describes how eFlow stores its data. To learn about the data storage hardware options, refer to the "Data storage devices" section in *Kofax eFlow Technical Specifications*.

eFlow databases are for internal purposes only and should not be changed.

Application setup data

Core objects containing the application setup data are serialized to XML files when the user saves the changes from one of the eFlow design modules, such as Design, Design Free, or Design Smart.

Storage

The application setup data can be stored on the file system or in the SQL database, depending on the <UseSqlDynamicStorage> basic configuration parameter value.

The relevant TisConfiguration.config file belongs to the eFlow_6 web application and is installed in the `C:\inetpub\wwwroot\Tis Web Site\eFlow_6\Bin\ConfigSources` folder.

By default, <UseSqlDynamicStorage> is False, that is, the setup data is saved on the file system.

- File system

  eFlow application setup data is stored under [AppData]\Server\[ApplicationName]\Setup.
- SQL server

  Setup data for all eFlow applications is stored in the <E_Setup> and <E_Configuration> tables in the eFlow_Management database.

Add setup attachments

When setup is stored on the NTFS, it is possible to copy the additional custom files to the setup folder, and the eFlow server will read them together with other setup files.

This method cannot be used when the setup data is saved in the database, because the eFlow server does not consider the local changes.

In this case, it is necessary to use the **Attachments Manager** utility in the Design module to add custom attachment files to the application setup.

Application dynamic data

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, with the image files being the most important.

Storage

Application dynamic data can be stored on the file system or in the SQL database, depending on the <UseSqlDynamicStorage> basic configuration parameter value.

The relevant TisConfiguration.config file belongs to the eFlow_6 web application and is installed in the `C:\inetpub\wwwroot\Tis Web Site\eFlow_6\Bin\ConfigSources` folder.

By default, <UseSqlDynamicStorage> is False; that is, the dynamic data is saved on the file system.

- File system

  Application dynamic data is stored in the `[AppData]\Server\[ApplicationName]\Dynamic` folder.

  The data of each collection is stored in a separate folder.

  You should never try to delete or move collections directly from the dynamic data folder. You can do this either in the Control module or using custom code.

  You must ensure that the TisAppPool user has read and write permissions for the storage location.

- SQL server

  Application dynamic data is stored in the <E_Attachment> table in the application workflow database.

STS configuration

The eFlow default security configuration files DomainSecurity.xml and UsersId.xml contain a description of the user roles and permissions applied in the eFlow modules.

Storage

The security configuration can be stored on the file system or in the SQL database, depending on the <UseSqlDynamicStorage> basic configuration parameter value.

The relevant TisConfiguration.config file belongs to the TisDefaultSTS web application and is installed in the `C:\inetpub\wwwroot\Tis Web Site\TisDefaultSTS\Bin\ConfigSources` folder.

- File system

  Security configuration files are stored under `[AppData]\Server\System\Setup`.

- SQL server storage

  The eFlow security configuration data is saved in the <E_Setup> table in the eFlow_Management database because it is the setup data of the eFlow system application.

Create eFlow internal databases

The eFlow_Management and eFlow_Monitor databases are created automatically during the installation when TisAppPool starts working.

The application workflow database <ApplicationName>_Workflow is created for each application during the application installation.

Switch between the file system and database storage

By default, the eFlow installation configures the setup and dynamic data to be stored on the file system.

To store the data in the database:

1. Open IIS and stop TisAppPool.
2. Change the <UseSqlDynamicStorage> value to True in the server-side TisConfiguration.config files under the eFlow_6 and TisDefaultsSTS web applications (`C:\inetpub\wwwroot\Tis Web Site\eFlow_6\Bin\ConfigSources` folder and `C:\inetpub\wwwroot\Tis Web Site\TisDefaultSTS\Bin\ConfigSources` folder).

   The <UseSqlDynamicStorage> parameter must have the same value in eFlow_6 and TisDefaultSTS. Saving the security setting on the file system storage and the app data on dynamic storage can cause the system to malfunction.

3. Restart IIS to apply the changes.
4. The data will be uploaded to the database, but not removed from the file system.

   We recommend keeping the data for some time as a backup, until you are sure that everything is working properly.

   Once you are sure that setup data was uploaded properly, you can remove everything from the local folders.

eFlow Management database

The eFlow runtime management data is stored in the eFlow_Management database on the SQL server. This database is created automatically when TisAppPool starts working.

To create the management database at runtime, the TisAppPool user should have dbcreator permissions that will not be required later. You can revoke this privilege after database creation.

As the eFlow server is designed to be stateless, every change in the runtime system data is accompanied by the relevant database record change. The eFlow_Management database contains the following information:

- Applications

- Runtime stations
- License
- User tags
- Alerts
- Autorun stations configuration

eFlow Monitoring database

The eFlow Monitoring database contains eFlow internal statistics data used by the Supervise module. The database is created when TisAppPool starts working, together with the management database.

Application Workflow database

The application workflow database named [ApplicationName]_Workflow is created automatically during the application installation. This database contains all the application-related information:

- Collection metadata
- Workflow status: collections, queues, sessions
- SLA data: definitions, status

Statistics database

The application statistics database named [ApplicationName]_Statistics is created automatically when you enable statistics for the application in the eFlow Administrate module. Refer to the "Generate statistics" section in the *Kofax eFlow Administrate User Guide* for information on how to enable statistics.

The following statistics settings in Administrate affect the following tables of the Statistics database.

Table name	Administrate setting / Use condition	Filled by *
eFlow_AuditCollectionsQueue_ eFlow_AuditCollections	CollectionAudit is activated	eFlow server
eFlow_SessionAudit	SessionsAudit is activated	eFlow server
eFlow_CompareFields	Compare Fields is activated	Export station
eFlow_ComparePages	Compare Pages is activated	Export station
eFlow_ManualFormIdCounters	ManualFormId Operator Counters is activated	Organize station
eFlow_DataEntryTypistCounters	DataEntry Operator Counters is activated	Validate / Escalate stations
eFlow_DataEntryFieldAggrCounters	DataEntry Fields is activated	Validate / Escalate stations
eFlow_DataEntryFormAggrCounters	DataEntry Forms is activated	Validate / Escalate stations
eFlow_MaintenanceParams	Set during statistics table configuration	

Table name	Administrate setting / Use condition	Filled by *
eFlow_Stat_Sessions	Whenever any table requires session ID lookup	eFlow server

* Actually, it is always the eFlow server that fills the data (eFlow runtime clients never access the Statistics database directly); however, this column describes the "original" source of the recording request.

eFlow auditcollectionsqueue

The audit information about a specific queue in the eFlow system is stored in this table.

Column name	Description	Data type and flags
SessionID	Session information about the initiator of the operation.	Bigint NOT NULL
CollectionAction Name	A string describing the collection operation related to this record: • Unspecified • Lock • Create • Reject • Remove • Discard • UnlockFinal • UnlockNotFinal • Save • Prioritize • RestorePriority • MoveToQueue • Hold • Release	Char(20) NOT NULL
SourceStation	Source station for the collection action.	Nvarchar(50) NOT NULL
TargetStation	Target station for the collection action.	Nvarchar(50)
TimeTag	The time at which the operation occurred. This time is stamped by the database client (eFlow server) and denotes the time at which the eFlow server marked the operation.	Datetime NOT NULL

Column name	Description	Data type and flags
StoreTime	The time at which the record was inserted into the database. This time is stamped by the database server and denotes the time the record was inserted. This time may be different from TimeTag if the eFlow server machine time is not synchronized with the database server time. Also, because statistics operations are asynchronous, there might be a delay between this time and the insertion time.	Datetime NOT NULL

eFlow_AuditCollectionsQueue_eFlow_AuditCollections

This table stores audit information about collections in the eFlow system. Each lock, unlock, split, merge, and stream of data from the server to the client and from the client to the server is logged for auditing.

Column name	Description	Data type and flags
CollectionsOperationId	A GUID is unique to a single logical operation. There might be several records with the same value, which means that all those records represent a single logical operation. For example, Merge collection involves locking all the form collections, so all these audits will have the same operation ID.	Uniqueidentifier NOT NULL
CollectionName	The name of the collection on which the operation took place.	Nvarchar (128) NOT NULL
FlowName	The name of the flow to which the collection belongs.	Nvarchar(50) NOT NULL
FormsInCollection	The number of forms in the current collection.	Int NOT NULL

eFLOW_CompareFields

This table stores field-level "OCR statistics" information. For each dynamic field, the table stores a record containing the original value (usually this is the value just after OCR) and the final value (the value during export). Using predefined comparison algorithms, these results can be compared to produce statistics on recognized, rejected, bad, and more.

Column name	Description	Data type and flags
SessionId	Current session ID.	Bigint NOT NULL
SetupFlowName	The setup flow name of the collection that contains the form with the current dynamic field. This value is equal to the FlowType property of collection objects.	Varchar(128) NOT NULL
SetupFormName	The setup form name of the dynamic form that contains the current dynamic field. The setup form name is equal to the FormType property of dynamic forms.	Varchar(128) NOT NULL

Column name	Description	Data type and flags
SetupFieldName	The setup field name of the current dynamic field. This may be different from the dynamic field name if the field is part of a field array.	Varchar(128) NOT NULL
CollectionName	The name of the collection that contains the current dynamic field.	Varchar(128) NOT NULL
DynamicFormName	The name of the current dynamic form.	Varchar(128) NOT NULL
DynamicFieldName	The name of the current dynamic field.	Varchar(128) NOT NULL
OriginalValue	The original value of the field content if the field content was ever modified. This value remains NULL if the field content is never modified. Which means that the actual OriginalValue is identical to the FinalValue. If the field content is modified (even if its original value was restored eventually), the value does not remain NULL; it represents the field content that was filled by the Recognize station.	Varchar(256)
FinalValue	The final value of the field content, as found by the Export station.	Varchar(256)
OrigRectangle	The original rectangle of the field content if the field content was ever modified. This value is NULL if the field region is never modified. If the field content was modified, but the field region was not modified, the value will be NULL. If this value is not NULL, it usually represents the field rectangle that was filled by the Recognize station.	Varchar(128)
FinalRectangle	The final rectangle of the field content as found by the Export station.	Varchar(128)
TimeTag	The time at which the operation occurred. This time is stamped by the eFlow client and denotes the time at which eFlow marked the operation.	Datetime

Column name	Description	Data type and flags
StoreTime	The time at which the record was inserted into the database. This time is stamped by the database server and denotes the time the record was inserted. This time may be different from TimeTag if the eFlow client machine time is not synchronized with the database server time. Also, because statistics operations are asynchronous, there might be a delay between this time and the insertion time, even if all machines have the same time.	Datetime

eFLOW_ComparePages

This table stores the page-level "FormID statistics" information. For each dynamic page, the table stores a record containing information about the originally recognized EFI (usually FormID recognition) and the finally chosen EFI (usually after manual FormID). Using comparison algorithms, the original and final EFIs can be compared to determine the percentage of good, bad, and rejected EFIs.

Column name	Description	Data type and flags
SessionId	Relevant session ID.	Bigint NOT NULL
SetupFlowName	The setup flow name of the collection that contains the form with the current dynamic page. This value is equal to the FlowType property of collection objects.	Varchar(128) NOT NULL
CollectionName	The name of the collection that contains the current dynamic page.	Varchar(128) NOT NULL
OriginalEFIName	The original EFI that was recognized (usually by FormID) if the page ID was modified. This value will be NULL if the initial EFI was never modified. That is, if this value is NULL, it means that the actual OriginalEFIName is identical to FinalEFIName and that the page ID was never modified. If the page ID was modified (even if its original value was restored eventually), this value will not be NULL. If this value is not NULL, it usually represents the EFI assigned by the Recognize station.	Varchar(128)
OriginalIdMethod	Enumeration of the original EFI identification method. • 0 - Page not identified • 10 - Page identified automatically • 20 - Page identified manually • 30 - Default page ID chosen This will be NULL when OriginalEFIName is NULL.	Int

Column name	Description	Data type and flags
FinalEFIName	The final EFI name picked up by the Export station. Usually, this is the EFI name detected by the Recognize station if OriginalEFIName is NULL; otherwise, it is the EFI name inserted by Organize.	Varchar(512)
FinalIdMethod	Enumeration of the final EFI Identification method. • 0 - Page not identified • 10 - Page identified automatically • 20 - Page identified manually • 30 - Default page ID chosen	Int
OriginalIdInfo	The original template property of the dynamic form that contains the current dynamic page. This value will be NULL if the page template was never modified. If the page EFI was modified, but the template was not modified, the value will be NULL. If this value is not NULL, it usually represents the page template that was filled by the Recognize station.	Varchar(128)
FinalIdInfo	The final template of the page, as found by the Export station.	Varchar(128)
TimeTag	The time at which the operation occurred. This time is stamped by the database client and denotes the time at which eFlow marked the operation. This time may be different from the StoreTime if the eFlow machine time is not synchronized with the database server time. As statistics operations are asynchronous, there might be a delay between this time and the insertion time, even if all machines have the same time.	Datetime
StoreTime	The time at which the record was inserted into the database. This time is stamped by the database server and denotes the time the record was inserted. This time may be different from the TimeTag if the eFlow client machine time is not synchronized with the database server time. As statistics operations are asynchronous, there might be a delay between this time and the insertion time, even if all machines have the same time.	Datetime

eFlow_DataEntryFieldAggrCounters

This table extends the information stored in the eFlow_DataEntryFormAggrCounters table with information about the keys typed per setup field.

Column name	Description	Data type and flags
SetupFlowName	The setup flow name of the current field.	Varchar(128) NOT NULL
SetupFieldName	The setup field name of the current field.	Varchar(128) NOT NULL
TotalTypedCount	The total number of dynamic fields that belong to a specified setup field.	Bigint
TotalKeysHit	The total number of keystrokes for this field type (within the context of the collection and form type), between StartTime and EndTime of the main type entry.	Bigint NOT NULL
TotalFocusTimeSecs	The amount of time this setup field was in focus, between StartTime and EndTime of the main type entry, when the focus is on fields of this setup field type (within the context of the "parent" collection and setup form).	Float
TotalZoomTimeSecs	The amount of time spent with the image zoom open, between StartTime and EndTime of the main type entry, when the focus is on fields of this setup field type (within the context of the "parent" collection and setup form).	Float
TotalExceptionTimeSecs	The amount of time spent with the Exception dialog open, between StartTime and EndTime of the main type entry, when the focus is on fields of this setup field type (within the context of the "parent" collection and setup form).	Float
TotalLearnTimeSecs	The amount of time spent on learning and setting regions, between StartTime and EndTime of the main type entry, when the focus is on fields of this setup field type (within the context of the "parent" collection and setup form).	Float
StoreTime	The time at which the record was inserted into the database.	Datetime

eFlow_DataEntryFormAggrCounters

This table extends the information stored in the eFlow_DataEntryTypistCounters table with information about the keys typed per form.

Column name	Description	Data type and flags
SetupFlowName	The setup flow name of the collection containing the form.	Varchar(128) NOT NULL
SetupFormName	The setup form name of the current form.	Varchar(128) NOT NULL
TotalLockedCount	The total number of dynamic forms that belong to the specific setup form type.	Bigint
TotalKeysHit	Total number of keystrokes for this form type (within the context of the collection), between StartTime and EndTime of the main type entry.	Bigint NOT NULL
TotalWorkTimeSecs	The total amount of time spent working in the station, between StartTime and EndTime of the main type entry.	Float

Column name	Description	Data type and flags
TotalZoomTimeSecs	The amount of time spent with the image zoom open, between StartTime and EndTime of the main type entry, when the focus is on any fields that belong to this setup form (within the context of the specified collection).	Float
TotalExceptionTimeSecs	The amount of time spent with the **Exception** dialog open, between StartTime and EndTime of the main type entry, when the focus is on any fields that belong to this setup form (within the context of the specified collection).	Float
TotalLearnTimeSecs	The amount of time spent on learning and setting regions, between StartTime and EndTime of the main type entry, when the focus is on any fields that belong to this setup form (within the context of the specified collection).	Float
StoreTime	The time at which the record was inserted into the database.	Datetime

eFlow_DataEntryTypistCounters

This table stores statistics on Validate operators. Using this table, it is possible to get typist statistics.

Column name	Description	Data type and flags
ID	Primary key.	Bigint NOT NULL
SessionID	Represents typist session information.	Bigint NOT NULL
StartTime	Start time for which this record is relevant (based on the eFlow client's machine time).	Datetime
EndTime	The end time for which this record is relevant (based on the eFlow client's machine time).	Datetime
KeysHit	The total number of keys hit during the time between StartTime and EndTime. This includes non-visible keys, such as Enter.	Int
FieldsLocked	The number of fields locked by the last GetCollections operation.	Int
FieldsTyped	The number of fields typed during the time between StartTime and EndTime. This includes fields that were not modified, that is, fields that were only viewed and skipped over, such as Enter or Tab.	Int
FieldsModified	The number of fields modified during the time between StartTime and EndTime. Enter and Tab in a field are not considered field modification.	Int
FieldsLearned	The number of fields learned by the last GetCollections operation.	Int

Column name	Description	Data type and flags
FormsLocked	The number of forms locked by the last GetCollections operation.	Int
FormsTyped	The number of forms typed (that is, the number of forms in which any field was typed) during the time between StartTime and EndTime.	Int
FormsLearned	The number of forms learned (that is, the number of forms in which any field was learned) during the time between StartTime and EndTime.	Int
CollectionsLocked	The number of collections locked by the last GetCollections operation.	Int
CollectionsTyped	The number of collections typed (that is, the number of collections in which any field was typed) during the time between StartTime and EndTime.	Int
CollectionsFinalized	The number of collections finalized (StampAsFinal set to True) when PutCollections was performed. This will usually be zero when a "Quick save" is made.	Int
CollectionsLearned	The number of collections learned (that is, the number of collections in which any field was learned) during the time between StartTime and EndTime.	Int
ZoomTimeCountSecs	The amount of time spent with the image zoom open between StartTime and EndTime.	Real
ExceptionTimeCountSecs	The amount of time spent with the Exceptions dialog open between StartTime and EndTime.	Real
BreakTimeCountSecs	The amount of time spent without any activity ("Idle time"). Only idle periods greater than a predefined, hard-coded threshold (5 seconds) are accumulated.	Real
LearnTimeCountSecs	The amount of time spent on learning between StartTime and EndTime.	Real
StoreTime	The time at which the record was inserted into the database. This time is stamped by the database server and denotes the time the record was inserted.	Datetime

eFlow_MaintenanceParams

The information used by background database server operations is stored in this table.

Column name	Description	Data type and flags
PurgeRecordsOlderThan_x_Days	Specifies the maximum number of days for which data will be kept in the database. Any statistical information older than this number of days will be purged (deleted) by a background database task. Negative or zero values will be treated as Never purge.	Int NOT NULL
DatabaseVersion	Specifies the statistics database version.	Int NOT NULL

eFlow ManualFormIdCounters

This table stores statistics on Organize operators. Using this table, it is possible to get statistics on the number of pages, forms, and collections per given period that a specific operator worked on.

Column name	Description	Data type and flags
ID	Primary key. For future use.	Bigint NOT NULL
SessionId	GUID that represents session information on the Organize operator. This column is a foreign key to the eFlow_Sessions table, from which additional information can be accessed, such as the station name, username, and computer name represented by this column.	Bigint NOT NULL
StartTime	Start time for which this record is relevant (based on the eFlow client's machine time).	Datetime
EndTime	The end time for which this record is relevant (based on the eFlow client's machine time).	Datetime
CollectionsLocked	The number of collections locked by the last GetCollections operation.	Int
CollectionsFixed	The number of collections from which at least one page/form was fixed during the time between StartTime and EndTime.	Int
CollectionsFinalized	The number of collections finalized (StampAsFinal set to True) when PutCollections was performed. This will usually be zero when a "Quick save" is made.	Int
PagesLocked	The number of pages locked by the last GetCollections operation.	Int
PagesFixed	The number of pages fixed during the time between StartTime and EndTime.	Int
FormsLocked	The number of forms locked the by last GetCollections operation.	Int
FormsFixed	The number of forms completely or partially fixed during the time between StartTime and EndTime.	Int
BreakTimeSecs	The amount of time spent without any activity ("Idle time"). Only idle periods greater than a predefined, hard-coded threshold (30 seconds) are accumulated.	Float

Column name	Description	Data type and flags
StoreTime	The time at which the record was inserted into the database. This time is stamped by the database server and denotes the time the record was inserted. This time may be different from TimeTag if the eFlow client machine time is not synchronized with the database server time. Also, because statistics operations are asynchronous, there might be a delay between this time and the insertion time, even if all machines have the same time.	Datetime

eFlow_SessionAudit

This table stores audit information on sessions in the eFlow system. This table audits session operations, such as logins and logout.

Column name	Description	Data type and flags
SessionID	Represents session information for which this record is relevant.	Bigint NOT NULL
EventType	Describes the session operation. Can be Login or Logout.	Char(20) NOT NULL
LogoutType	Describes the logout operation. Can be Disconnected or Exited.	Char(20)
AdditionalInfo	Additional logout information.	Nvarchar(100)
EventTime	The time at which the operation occurred. This time is stamped by the database client and denotes the time at which eFlow marked the operation.	Datetime
StoreTime	The time at which the record was inserted into the database. This time is stamped by the database server and denotes the time the record was inserted. This time may be different from the EventTime if the eFlow machine time is not synchronized with the database server time. Also, because statistics operations are asynchronous, there might be a delay between this time and the insertion time, even if all machines have the same time.	Datetime

eFLOW_Stat_Sessions

This table can be used to convert a session ID to the session information: Station name, username, machine name, process name, process ID.

Many sessions with the same station, user and computer name can exist during the same period.

Column name	Description	Data type and flags
ID	Primary key for accessing the statistics session information.	Bigint NOT NULL
StationID	Station ID matching the session ID.	Varchar(50)
MachineName	Computer name matching the session ID.	Varchar(50)
UserName	User name matching the session ID.	Varchar(50)
ProcessId	Process ID matching the session ID.	Int
ProcessName	Process name matching the session ID.	Varchar(50)
StationName	Station name matching the session ID.	Varchar(50)
StoreTime	The time at which the record was inserted into the database.	Datetime

eFlow server mechanisms

Distributed tasks

eFlow client-server communication is one-directional for security reasons. Normally, firewalls are configured to prevent servers from contacting the clients directly.

Sometimes the server needs to transfer information to clients, for example, to send notifications about a new application. This is done with the help of a distributed tasks mechanism that does not use the eFlow client-server communication infrastructure.

The following internally distributed task types are supported:

- Specific client
- Group of clients
- Broadcasting

One at a time, every client checks if any task notifications waiting. The <TaskPollingInterval> configuration parameter defines how often the client contacts the server. Every time the client contacts the server, the time stamp of this contact is saved. This is done even if the station is idle, thus providing the data for the station liveliness check.

The client can ignore a notification if it is not relevant. A notification that was not delivered is removed automatically. The <TaskLifeTimeout> configuration parameter defines how long the notification will be kept on the server.

Resource disposal

The eFlow resource disposal service uses the time stamp of the last client contact to decide if the station is still alive. When active eFlow clients contact the server, the time stamp of the contact is saved. Idle clients' time stamps are saved when they check for tasks (see Distributed tasks).

The station is considered not working if the time that passed since the last time it contacted the server exceeds the time interval defined by the <StationLivelinessTimeout> configuration parameter.

Once the resource disposer finds that the station is not working, it releases the station license and makes the relevant changes in the management and application workflow databases. The <ResourceDisposerInterval> configuration parameter defines how often the resource disposer checks for non-working stations.

To learn more about the relevant configuration parameters, see Resource disposal and task management intervals.

eFlow customizations

This section describes the internal eFlow customization mechanism.

eFlow is a highly customizable system. In addition to powerful design modules, it supports a variety of customization mechanisms. Customizations make feasible the implementation of practically any project requirements.

You can write code that is applied when certain events are fired or create custom recognition activities. If nothing else provides an acceptable solution, you can always create a new custom station.

Customization DLLs

Dynamic library (*.dll) is the simplest and most used eFlow customization choice.

Basic customization code samples are included in the standard eFlow installation and can be a good starting point for beginners.

Configure eFlow events

Once the customization code is created, follow these steps to attach the DLL to the relevant eFlow event:

1. Select your application in the **LaunchPro** module.
2. Open Design and switch to the **Workflow** tab.

3. Select the required station and open its properties.
4. Attach your DLL to the relevant event.
5. Save your changes.

How eFlow stores customization data

When you save the latest Design changes, eFlow Core performs the following actions:

- Customization DLLs are copied to the Customization folder under the application setup.
- Customization methods are connected to relevant events in the application setup configuration files.

Thus, the customization DLL becomes part of the application CAB. If needed, it will be exported together with other elements of the application setup.

If your customization code is compiled as EXE, it will not be copied automatically to the application setup folder. We do not recommend using EXE files in eFlow event customizations.

Runtime actions

At runtime, customization DLLs are downloaded to the Customization folder in the client <AppData> folder.

- eFlow looks for the DLLs in the Customization folder. If found, it will run the customizations only from there.
- If the required DLL is not found in the Customization folder, the eFlow Bin folder is searched.

The recognition workflow always looks for TIS and Customized Activity DLLs in the eFlow Bin folder only.

Update customization

Once the customization DLL is downloaded to the client, it will not be synchronized automatically with the server.

To update the customization DLL:

1. Remove the relevant event customization from the Design.
2. Reconnect the DLL by selecting it and attaching it to the event.
3. Save your changes.

Chapter 4

eFlow security mechanism

This chapter explains the eFlow security model and its default implementation and describes the possible ways of customizing the security service.

eFlow security mechanism overview

Security model

The eFlow security model is based on the Microsoft claims-based identity architecture and is implemented with the Windows Identify Foundation (WIF) API. Refer to "WIF" on the Microsoft website for more information.

WIF externalizes authentication and thus application designers can focus on implementing business logic. Instead of implementing authentication in the application, they use an external system to provide authentication. This system is a web service, or a product, known as Security Token Service (STS). Refer to "STS" on the Microsoft website for more information.

STS manages authentication by issuing security tokens to those whom the business trusts to gain access to its private network. Without a security token, you cannot pass through the boundaries of a firewall, and therefore cannot access a business's private network.

The eFlow installation includes a sample Security Token Service implemented in the DefaultSTS web application that is installed under the TiS Web Site.

The source code of DefaultSTS can be found in the `...\eFLOW 6\Customization Samples\STS Sample` folder. It is recommended to use it for customizing STS.

 The default eFlow security service is not sufficient for real project requirements and should be replaced by other solutions.

Configure eFlow security

Authentication types

eFlow supports both Windows and User authentication types. Windows authentication is selected by default during the installation. See Runtime security checks for more information.

Authentication type settings are saved in the basic configuration file in the eFlow website installation folder: `C:\inetpub\wwwroot\Tis Web Site\eFlow_6\Bin\ConfigSources\TisConfiguration.config`.

To change the authentication type, edit this configuration file.

```
<eFlowPlatformConfigServer>  <AuthenticationType>Windows</AuthenticationType>  </eFlowPlatformConfigServer>
```

Define roles per application

Role definitions are a part of the application setup data and are set in the Design module. By default, any new application contains only the Administrator role, which grants full access to all eFlow modules.

The following predefined permissions are supported by eFlow:

- Administrator

  This user can perform all application-level actions.

  A user who is an Administrator in the System application has permissions to perform system-level operations, such as adding or deleting applications.
- EXECUTE

  This user can open this station or module.
- READ

  This user can read application data.
- WRITE

  This user can write application data.
- CONFIGURE

  This user can change the station configuration.

To create a new role:

1. In eFlow **LaunchPro**, select the application and open the **Design** tool.
2. In the Application Explorer tree, expand the **Security** node.
3. Right-click **Roles** and select **New Role**.
4. Enter a name for the role (for example, Typists) and press Enter.
5. Set the required role permissions.

For example, the Typist role would require **EXECUTE**, **READ**, and **WRITE** permissions for Validate stations.

When the **EXECUTE** permission is defined in the role but other permissions are disabled, the station can be launched, but it will be impossible to modify anything (such as dragging a collection to a different station in Control).

User permissions and roles

You must assign at least one application role to every eFlow user.

As long as the default STS is used, user permissions and roles are saved in configuration files in the `ProgramData\TIS\eFlow 6\AppData\Server\System\Setup` folder. See Default STS configuration for detailed information about these configuration files.

The user performing the installation is always granted administrator permissions for all eFlow applications.

The default security configuration is good enough to start exploring eFlow, but often it is not good enough for real-world projects. The security objects can be extended or replaced by more sophisticated ones and can be saved in many ways, as described in Customize STS.

Runtime security checks

Authentication

The authentication process depends on the user authentication type defined in the TisConfiguration.config basic configuration file.

- **Windows authentication**: The station tries to log in with the current user credentials. If it fails, it displays a login dialog box.
- **User authentication**: The login dialog box opens when the user logs in for the first time.

By default, if the login succeeds, the user credentials are saved per user in the isolated local storage on the client. Thereafter, any station on this client will log in automatically.

In certain situations, this default behavior does not suit the project requirements, for example:

- Users with different roles (for example, Typist and Scanner) need to log in to different stations on the same machine. The user information saved in the authentication cache will prevent switching between users.
- Due to project security requirements, it is necessary to enter a username and password every time the station logs in. In this case, login information should never be saved locally.

It is possible to control how the authentication information is saved.

- The user can clear the authentication cache using the Basic Configuration utility.
- The <SaveUserAuthentication> basic configuration client-side parameter indicates whether the authentication cache should be enabled.

Authorization

Once the authentication succeeds, the system needs to validate user permissions for the station the user is trying to log in.

eFlow authorization includes the following steps.

1. Find the roles defined for the current user in the specific application.
   By default, role definitions per user are stored in the `ProgramData\TIS\eFlow 6\AppData\Server\System\Setup\DomainSecurity.xml` file.
2. Verify that at least one role allows this user to run the station.

Some roles may not allow full station control. For example, changing the station configuration often requires special permissions. For such roles, only the relevant permissions are checked on login.

Certificates

eFlow uses TIS certificates to validate incoming tokens. eFlow installs default TIS certificates on all server and client machines.

The default eFlow certificates are not issued by an official Certificate Authority. Therefore, replace certificates with more secure certificates suited to your organization's needs.

Acquire new certificates

Consider the following options before acquiring new eFlow certificates.

Option	Pros	Cons
Use the existing company software certificates.	Secure. No need to create or purchase certificates.	You may not able to install certificates as you wish due to the company's IT policy.
Create certificates with a free certificate creation utility.	Installation is fully controlled. Free of charge.	It may not fit the requirements of the company security policy and may require some knowledge of certificates creation and management tools.
Buy certificates from one of the well-known certificate authorities (CA).	The highest level of security. Installation can be fully controlled.	It may add costs to the project.

Replace the default certificates with the new ones

The default certificates must be replaced by installing the new certificates on all on all eFlow server and client machines. For more information, see Install eFlow certificates.

If you rename the new certificates with the default eFlow names: TiS-eFlow-rp and TiS-eFlow-rpsts, then you do not need to change any code or configuration file after installing the new certificates.

If new certificate names differ from the default TIS names, make the following changes.

- **Changes in DefaultSTS**
  1. Open `<eFLOWInstallPath>\Customization Samples\STS Sample\DefaultSTS.sln` in Visual Studio.
  2. Replace all instances of TiS-eFlow-rpsts.com and TiS-eFlow-rp.com with the new names.
  3. Compile the solution.
  4. Copy the output TiS.Security.TisDefaultSts.dll to the `<IISROOTFOLDER>\TiS Web Site \TisDefaultSTS\Bin` folder and overwrite the existing DLL.
- **Changes in the Configuration files**

Replace all instances of TiS-eFlow-rpsts.com and TiS-eFlow-rp.com with the new names in the server and client configuration files.

- Server:
  `<IISROOTFOLDER>\TiS Web Site\eFlow_6\web.config`
  `<IISROOTFOLDER>\TiS Web Site\TisDefaultSTS\web.config`
- Client:
  `<eFLOWInstallPath>\Bin\ConfigSources\BehaviorSection.config`
  `<eFLOWInstallPath>\Bin\ConfigSources\BindingSection.config`
  `<eFLOWInstallPath>\Bin\ConfigSources\ClientSection.config`

Default STS configuration

This section describes the structure of the default STS configuration files and explains how to handle them.

Edit STS configuration files

The eFlow installation automatically gives the installing user eFlow Administrator permissions for all applications. Of course, this basic configuration cannot satisfy the requirements of real-life projects.

To edit the STS configuration, use the STS Management utility. This is a server-side utility that enables you to configure user roles and permissions.

This utility is essential when the <UseDynamicSQL> property is True because it provides access to the security configuration information saved in the database.

Before editing the security configuration, stop TisAppPool and restart it after you have finished.

To configure user roles and permissions:

1. Open the STSManagement.exe file from `c:\inetpub\wwwroot\Tis Web Site \TisDefaultSTS\Bin\`.
2. Select the eFlow application to be configured.
3. Select the user roles you want to apply.
4. Select a user or a group from the active directory list.
5. Define the user roles.

If you change the security configuration files manually, you must save them in the database; otherwise, the changes will not be reflected in eFlow.

DomainSecurity.xml

The DomainSecurity.xml configuration file contains user security definitions per application.

```
<DomainSecurity> <Applications>
<AppAndUsersAppName="System">
<Users>
<UserAndRoles Name="NTAUTHORITY\SYSTEM" IsGroupName="false">
<Roles>
<Role Name="Administrator">
</Roles>
</UserAndRoles>
</Users>
</AppAndUsers>
<AppAndUsers AppName="">
<Users>
<UserAndRoles Name="NTAUTHORITY\SYSTEM" IsGroupName="false">
<Roles>
<Role Name="Administrator">
</Roles>
</UserAndRoles>
</Users>
</AppAndUsers>
</Applications>
<UserGroups>
</UserGroups>
</DomainSecurity>
```

<Applications>

This container element includes a list of application security definitions. The AppName attribute specifies the name of the application.

- AppName=System

  There should be at least one application node named System.

  Ensure that all users defined under AppName="" (or AppName=NameOfApplication) are also defined under AppName=System so that the system knows the authorization level of the user trying to perform system-level actions (such as writing to the log).

  All users must have READ and WRITE permissions under the System application node. We recommend that you be careful when adding more permissions so that no unauthorized user can perform actions that may change system behavior.
- AppName=" "

  An empty application name means that the default settings are applied to all applications in the absence of application-specific ones.
- AppName=NameOfApplication

  If users and roles are defined explicitly for a specific application, the default settings are ignored.

<users>

Contains the list of users and their roles per application. Define at least one interactive user for every application.

- Windows authentication

  Make sure the username is written in the following format:

  "DOMAIN\User" (for domain users) or "MACHINE\User" (for local users)

  You can use the Windows Active Directory group name in the following format "DOMAIN \GroupName".
- User authentication

  Usernames are taken from the UsersID.xml file.

IsGroupName attribute

By default, the IsGroupName attribute is False, which indicates that login is performed as defined in the <UserAndRoles> section of the current application.

IsGroupName="true" indicates that the Name attribute is the name of the group. It can be a Windows group or one of the internal STS groups defined in the <UserGroups> section.

For example, <UserAndRoles Name="Typists" IsGroupName=true> defines the Typists user group.

On login, the system searches for the username in the <UserGroups> section of DomainSecurity.xml.

A login error occurs if the group name is not found in the <UserGroups> section or Windows Active Directory.

<UserGroups>

Contains the list of internal STS user groups. Every group contains the names of users that have the same roles. It can include both Windows AD users and eFlow users defined in UserId.xml.

Use STS groups when the same user definitions are required by several eFlow applications or if many users share the same role definitions. Instead of copying the whole list several times, you can create an internal STS group that includes all these users.

The STS group name can be included in any <Users> section of any application security definitions.

Do not confuse STS user groups with Active Directory groups. The only purpose of internal STS groups is to make the editing of the default DomainSecurity.xml easier.

In the following example, the <UserGroups> section contains Typists as a group definition.

```
<UserGroups>
<Users>
<UserGroup Name="Typist">
<User Name="DOMAIN\Anna"/>
<User Name="DOMAIN\Maria"/>
```

```
<User Name="DOMAIN\Joe"/>
</UserGroup>
</Users>
</UserGroups>
```

<Roles>

Contains the list of roles defined per user in the current application. Every user should have at least one role defined.

You can define more than one role per user.

Permissions are created as the superposition of all role permissions.

The role name must be the same as defined in the Design module.

If the role you defined in DomainSecurity.xml does not exist in the application, the station returns an error on login and the relevant error message appears in the TIS log.

The role defined under the default (empty) application node (AppName=" ") will be applied in all applications.

If you create a custom role in the Design module (such as Operators or Typists), this role will only be applied if it is entered under the specific application (AppName=NameOfApplication). If you use the AppName=" " setting for all applications, the custom role will not be applied.

System application users can have only the following predefined roles:

- Administrator
- READ
- WRITE
- EXECUTE

Users defined in other applications can have the Administrator role or any other role defined in the Design module.

The following example shows how to add READ and EXECUTE roles to user definition:

```
<UserAndRoles Name="TISDOMAIN\User" IsGroupName="false">
<Roles>
<Role Name="READ">
<Role Name="EXECUTE">
</Roles>
</UserAndRoles>
```

userId.xml

The UserId.xml configuration file contains user definitions that are applied when the User authentication method is selected.

```
<UserIds>
<Users>
```

```
<UserId User="Admin" Password="password" />
</Users>
</UserIds>
```

<Users>

Contains the list of <UserId> elements with names and passwords to be used by eFlow.

Install eFlow certificates

You can install the eFlow certificates manually under certain circumstances. For example:

- You need to replace the default eFlow certificates with those issued for your project.
- Error messages after a new installation indicate certificate problems.

Install certificates from the Certificates Management Console

You must be logged on as an administrator to perform these steps.

1. Click the Windows Start button and type mmc, and then double-click mmc.exe.
   The Microsoft Management Console tool opens.
2. From the main menu, select **File** > **Add/Remove Snap-in**.
3. Click **Certificates** > **Add**.
4. Click **Computer account** > **Local computer** > **Finish** > **Ok**.

Delete invalid certificates

Server

1. Expand the **Certificates - Local Computer** tree and go to the **Personal** > **Certificate**.
2. Delete the two eFlow certificates available found in that store.

Client

1. Expand the **Certificates - Local Computer** tree and go to the **Trusted People** > **Certificate** store.
2. Delete the two eFlow certificates available found in that store.

Reinstall certificates

Reinstall the server or client certificates if required.

Server

1. Right-click **Certificates** under **Personal** store and select **All Tasks** > **Import**.

2. Click **Browse** and navigate to the `c:\inetpub\wwwroot\Tis Web Site\TisDefaultSTS\Certificates` folder.
3. Change the file type to Personal Information Exchange (*.pfx) and select the Tis-Eflow-rp.pfx file.
4. Type the password for the private key as eFlow.
5. Select **Mark this key as exportable**.
6. Repeat these steps for the Tis-Eflow-rpsts.pfx file.

Set access permissions for private keys:

1. Select the installed certificate in the **Personal** store and right-click **All tasks** > **Manage private keys**.
2. Click **Add** > **Locations** and change the location to your local machine.
3. In the **Enter the object names to select** field, add: IIS AppPool\TisAppPool.
4. Click **Check Names** and ensure that TisAppPool is a valid user name.

Client

1. Right-click **Certificates** under the **Trusted People** store and select **All Tasks** > **Import**.
2. Click **Browse** and go to `c:\Program Files (x86)\TiS\eFlow 6\Certificates`.
3. Change the file type to Personal Information Exchange (*.cer) and select the Tis-Eflow-rp.cer file.
4. Place this certificate in the **Trusted People** store.
5. Repeat these steps for the Tis-Eflow-rpsts.cer file.

The Web Completion and Web Scanning servers are eFlow clients and need client-side certificates to function properly.

Install certificates using command line utilities

Server

Install private certificates

```
C:\Program Files (x86)\TIS\eFlow 6\Bin\eFlowCmd.exe InstallCertificate MY "c:\inetpub\wwwroot\TiS Web
    Site\TisDefaultSTS\Certificates\TiS-eFlow-rp.pfx" "eFlow"
```

```
C:\Program Files (x86)\TIS\eFlow 6\Bin\eFlowCmd.exe InstallCertificate MY "c:\inetpub\wwwroot\TiS Web
    Site\TisDefaultSTS\Certificates\TiS-eFlow-rpsts.pfx" "eFlow"
```

Install public certificates

```
c:\windows\system32\certUtil.exe -addstore "TrustedPeople" "c:\inetpub\wwwroot\TiS Web
    Site\TisDefaultSTS\Certificates\TiS-eFlow-rp.cer"
```

```
c:\windows\system32\certUtil.exe -addstore "TrustedPeople" "c:\inetpub\wwwroot\TiS Web
```

```
    Site\TisDefaultSTS\Certificates\TiS-eFlow-rpsts.cer"
```

Set access permissions for the TisAppPool user

```
 "c:\inetpub\wwwroot\TiS Web Site\TisDefaultSTS\Certificates\winhttpcertcfg.exe" -g -c
 LOCAL_MACHINE\My -s
    "TiS-eFlow-rp.com" -a "TiSAppPool"
```

```
"c:\inetpub\wwwroot\TiS Web Site\TisDefaultSTS\Certificates\winhttpcertcfg.exe" -g -c
 LOCAL_MACHINE\My -s
    "TiS-eFlow-rpsts.com" -a "TiSAppPool"
```

Client

```
c:\windows\system32\certUtil.exe -addstore "TrustedPeople" "c:\Program Files (x86)\TiS
\eFlow
    6\Certificates\TiS-eFlow-rp.cer"
```

```
c:\windows\system32\certUtil.exe -addstore "TrustedPeople" "c:\Program Files (x86)\TiS
\eFlow
    6\Certificates\TiS-eFlow-rpsts.cer"
```

Troubleshooting

This section describes the issues that you may encounter and their resolution.

eFlow client

Once the eFlow server-side is installed correctly, you must check the eFlow client-side installation.

Review the eFlow client-side basic configuration

1. Open `C:\Program Files(x86)\TiS\eFlow 6\Bin\ConfigSources\TisConfiguration.config`.
2. Check the validity of the parameters configured by the installation:
   - <LocalMachineName>
   - <ServerMachineName>
   - <eFlowDataPath>
   - <eFlowInstallPath>
   - <Communication protocol>

     The default value is https. Ensure that the protocol name is written correctly and is identical to the protocol selected on the server.

Verify the validity of the client-side configuration of WCF services

1. Open `C:\Program Files(x86)\TiS\eFlow 6\Bin\ConfigSources\BindingSection.config`.
2. Open `C:\Program Files(x86)\TiS\eFlow 6\Bin\ConfigSources\ClientSection.config`.

3. Make sure that the configuration is correct. For example, the server name is valid.

eFlow installs the default client configuration files together with those changed during installation. They can be a useful reference for troubleshooting.

eFlow certificate problems

The Event log may contain messages indicating eFlow certificate problems.

In this case, you may need to re-install the certificates.

eFlow licensing problems

eFlow only runs on a machine that has a serial number. eFlow uses the mainboard serial number for license encryption. On virtual machines, if the serial number is blank, you must check whether the machine has a serial number. To do so, enter the following command at the Windows Command prompt.

```
gwmi win32_baseboard
```

If no serial number is returned, contact the IT team and get the problem fixed.

SQL server

Open SQL Server Management Studio and log in with the SQL server user credentials defined in your project. By default, eFlow works with "sa" as the username and password.

1. If login fails, check your SQL server settings.
2. If the login succeeds, verify if the **eFlow_Management** database is created.

Testing the validity of this database is out of the scope of this checklist.

eFlow communication ports

Make sure that the ports used by eFlow are open for communication. See Communication ports.

User permissions

Verify that the eFlow services and interactive users have all the necessary permissions. See Set user permissions for details.

Customize STS

This section describes the ways of customizing the default STS service provided with the eFlow installation.

Replace the default STS

Incorporate an existing company STS into eFlow

Many organizations have their own STS systems and are willing to use them in eFlow as well.

It is possible to use a cloud STS such as a LiveID STS, a pre-built STS such as Active Directory Federation Services (AD FS) 2.0, and many others.

Create a custom STS

Sometimes no 3rd party STS is available or specific project requirements impede using the existing one.

If you want to issue custom tokens or provide custom authentication or authorization, you can build a custom STS using WIF. WIF helps you build your STS and provides extensibility points to implement authentication logic based on your business requirements.

Customize the default STS

Sample STS project

You can customize the TIS default STS implementation in various ways according to project requirements.

The first and the most important step in STS customization is the requirements analysis.

The eFlow installation includes a basic STS project that can serve as a starting point for the customization. The DefaultSTS solution is available in the <eFlowInstallPath>\Customization Samples folder.

Change the following files in the DefaultSTS solution if security objects do not suit the project requirements:

- DomainSecurity.cs
- UsersId.cs
- TisDefaultSts.cs
- Helper.cs

We do not recommend changing the default service name or URL. If the service name or URL is changed, you will need to make multiple changes in the server and client configuration files.

Installation

The compiled customized TiS.Security.TisDefaultSts.dll should be copied to the `<IISRootFolder>\Tis Web Site\TisDefaultSTS\Bin` folder.

Chapter 5

Recognition

This chapter provides information about different aspects of image recognition in eFlow.

Recognition workflow overview

This section provides an overview of the recognition workflow in eFlow.

From individual recognition stations to recognition workflow

Recognition is the main part of the workflow in any eFlow project. In the first eFlow versions only one station was responsible for all form recognition and OCR or ICR actions. As the functionality requirements grew, it was necessary to split this station into several more specialized ones that could be added to the workflow as separate units, such as Form ID, Freedom, Recognition, Smart, and other stations. Each of them was used in specific workflow scenarios only and customization options were rather limited.

Recognition Services introduced in eFlow 5 replaced the specialized recognition stations. They are implemented as a workflow based on the Windows Workflow Foundation Framework (refer to "Windows Workflow Foundation" on Microsoft website for more information). The recognition workflow provides flexibility and a structured approach to the station architecture at the same time.

Recognition workflow advantages

The most significant advantages of the recognition services architecture compared to the individual station approach are as follows:

- One starting point for all recognition modules

  One Recognize station replaces multiple stations performing recognition (Form ID, Recognition, Free Match, Processing, Smart, and so on). The Recognize station is based on the generic eFlow processing station module (efProcessShell.exe) that runs TiS.Recognition.dll capable of loading requested recognition activities.

  In most cases, the required functionality can be achieved by applying different activities to one station. As a result, the number of get/put collection operations decreased, and the system performance is improved.
- Workflow versatility

  Predefined eFlow recognition activities can be used in various combinations to provide the required functionality. The Lego-style architecture allows the creation of complex recognition flows within minutes.

Recognition services are fully backward compatible. You can reproduce the existing functionality simply by changing the workflow at design time.

- Ease of customization

  If it is not possible to achieve the required result by combining the predefined activities: you can customize the existing recognition activities or write new ones.

  Visual workflow programming allows you to create loops, assignments, and so on, without writing a line of code. Thus, developers can concentrate on recognition tasks only.

  Custom recognition activities can be reused in different projects. These small building blocks can be easily moved and incorporated into different recognition workflows.

Configure the Recognize Workflow

1. Open the Design module **Workflow** tab.
2. Add a new Recognize station to the application workflow.
3. Open the **Station configuration** dialog box.
4. Select the XAML that will define the station behavior. This can be done in two ways:
   - Select the required file from the existing XAML files list.

     This list contains all XAMLs that have been imported previously into the current CAB.
   - Import a new XAML.

     Basic eFlow recognition configuration files are in the `\Customization Samples \Recognition Workflow\XAMLSamples` folder. Each of these XAMLs provides functionality that reproduces one of the old recognition stations. They are good enough for initial acquaintance with the Recognition workflow and can be easily replaced later with the custom XAML files.

Once you have selected XAML, you can set some additional station configuration parameters:

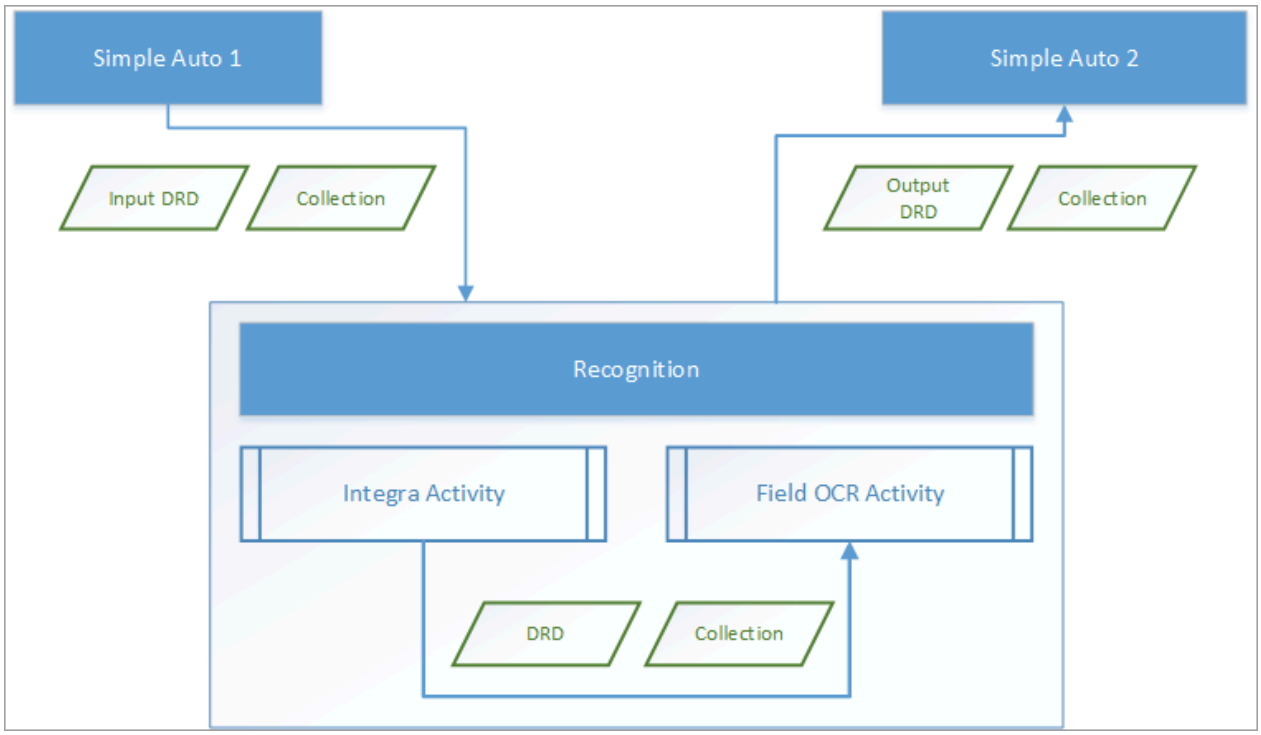
- Default Page OCR Engine.
- Default Smart model.
- Station Tags.

At runtime, the Recognize station loads the recognition workflow definitions stored in the Design module and runs the activities.

How the Recognize station works

The image below shows the Recognize station as a part of the workflow. The Simple Auto station preceding the Recognize station provides the input for the Recognize station in the form of collection data and a DRD object. If the DRD is absent, the Recognize station creates a new one.

The Recognize station can run several activities one after another. For example, it is often required to run the Integra activity followed by the Field OCR activity. Each activity updates the input DRD object; the output of the previous activity becomes the input of the next one. After all the activities are completed, this DRD is converted to collection data to become the Recognize station output.



Design flow

The eFlow Design Flow module helps you to modify the existing recognition workflow XAML or create a new and simple XAML easily.

To change the eFlow recognition workflow, you should have some working knowledge of Windows Workflow Foundation (Refer to "Windows Workflow Foundation" on the Microsoft website for more information) programming.

Use this module only for editing the existing XAML files or for demo or training sessions. For project development, we recommend using Microsoft Visual Studio 2010 or 2012.

To start editing the recognition workflow:

1. In **eFlow LaunchPro**, open the **Design Flow** tool.
2. Select one of the following options to load the XAML you want to work with:
   - **New file**
   - **Load from application**
   - **Load from file**
3. Make the necessary changes and save the result.
   The toolbox view of Design Flow contains a list of predefined eFlow recognition activities and some basic Microsoft workflow activities.

If you choose to create a new XAML, some predefined variables, arguments, and imports that are required to define the recognition workflow properly are available within this file.

Built-in recognition activities

See Built-In Recognition activities for a description of the predefined eFlow activities.

Recognition workflow API

The Recognition Services API Reference contains a detailed description of public interfaces, methods, and properties used by the eFlow recognition services. This information will help you to better understand the recognition activities parameters.

The public members can be used for any recognition services customization.

Built-in recognition activities

Built-in recognition activities are the main building blocks of any eFlow recognition workflow.

These activities can be used in various combinations to provide the required functionality.

This section describes the eFlow built-in recognition activities.

Integra activity

The Integra recognition activity performs structured forms recognition.

Parameter	Description	Type	Notes
Common			
EFIs	Gets or sets the list of EFIs to be used in recognition.	InArgument<IList <String>>	To check all the EFIs of the current flow, use the default parameter value. Otherwise, open the EFIs dialog box and select the EFIs from the list.
ImageEnhancement	Gets or sets an image enhancement string to be applied before form recognition.	InArgument<String>	To apply the enhancement, create the enhancement string using the Design module and copy it to the activity definitions.
InputDRD	Contains the initial document recognition data based on the current collection data.	InArgument<IDocument Recognition>	If no DRD data is provided, this activity will create a new DRD object.

Parameter	Description	Type	Notes
Pages	Gets or sets an array of pages that should be recognized.	InArgument<Int32[]>	To apply form recognition to all pages in a collection, use the default empty array. If you need to run the activity on specific pages only, you can use a VB or LINQ expression that returns the array of integers that contains the selected page numbers. See Custom Recognition activities for several examples of page definitions.
StopOnMatch	Gets or sets a value that indicates whether form recognition will stop after finding the first matching EFI.	InArgument<Boolean>	The default value is True. If set to False, the activity checks all the available EFIs.
Events			
PostMatchEFI	The event is fired after every EFI check, whether or not an EFI match was found.	InArgument<String>	
PostPage	The event is fired after page recognition.	InArgument<String>	
PreMatchEFI	The event is fired before evaluating every EFI in the selected EFIs list.	InArgument<String>	These events replace the IPE hooks used in previous eFlow versions.
PrePage	The event is fired before running recognition on every page.	InArgument<String>	
Misc			
DisplayName	Gets or sets the activity name.	InArgument<String>	The default name is IntegraActivity.
Result	Output DRD.	OutArgument<IDocumentRecognition>	Contains the final document recognition data.
OCR			
RunFieldOCR*	Gets or sets the value that indicates whether OCR will be applied on every recognized EFI.	InArgument<Boolean>	If no OCR engine is defined on the ROI level, the activity will run the field OCR engine. Results will be available on the PostEFIHook event in the DRD file.

Parameter	Description	Type	Notes
RunFieldOCR OnFinalMatch*	Gets or sets the value that indicates whether OCR will be applied on the ROI level after the final EFI recognition.	InArgument<Boolean>	Set this property to True if an ROI recognition engine is defined; otherwise, ROI recognition will not be performed. If no OCR engine is defined on the ROI level, the activity runs the field OCR engine.

* To decide which OCR activity to use, follow these guidelines:

- To allow eFlow to perform FormOut on the best-matched EFI, do not enable RunFieldOCR. eFlow will perform all the necessary actions automatically.
- To implement your logic based on recognition results (using the PostEFIHook event), apply RunFieldOCR.

  You can always use RunFieldOCROnFinalMatch if you want to perform recognition on the EFI that was selected as the best matching one.

  Do not use RunFieldOCR if "Stop On Match" is set to True (this is the default value).

Freedom activity

The Freedom recognition activity performs unstructured form recognition.

Parameter	Description	Type	Notes
Common			
FDDBFolder	Gets or sets the path to the FDDB folder.	InArgument	In most cases, leave this parameter undefined. Use the non-default path if required.
FFSs	Gets or sets the list of Freedom scripts to run.		By default, all scripts defined in the current flow are used.
InputDRD	Gets or sets the input recognition data object.	InArgument<IDocument Recognition>	This is the required input argument. Cannot be empty in the Freedom activity.

Parameter	Description	Type	Notes
Pages	Gets or sets the pages that should be recognized.	InArgument<Int32[]>	By default, this parameter is undefined which means that form recognition is applied to all pages in the collection. If you need to run the activity on specific pages, you can use a VB or LINQ expression that returns an array of integers that contains the selected page numbers. See Custom Recognition activities for examples of page definitions.
ScriptThreshold	Gets or sets the minimum acceptable script result confidence.	InArgument<Int32>	By default, all script results are considered valid.
StopOnFirstMatch	Gets or sets the value indicating whether the activity will stop after the first match.	InArgument<Int32>	The default value is True.
Events			
PostMatchFFS	The event is fired after running the Freedom script.	InArgument<String>	
PostMatchPage	The event is fired after page recognition is finished.	InArgument<String>	
PreMatchFFS	The event is fired before running the Freedom script.	InArgument<String>	To apply form recognition to all pages in the collection, use the default empty array. If you need to run the activity on specific pages, you can use a VB or LINQ expression that returns the array of integers that contains the selected page numbers. See Custom Recognition activities for examples of page definitions.
PreMatchPage	The event is fired before page recognition starts.	InArgument<String>	
Learning			

Parameter	Description	Type	Notes
TemplateDateFilter	Specifies the date of the template update.	InArgument<Int32>	Any templates that are updated before this date, are be ignored (skipping old templates).
TemplateDaysFilter	Specifies the maximum number of days passed from the last template update.	InArgument<Int32>	Any templates that were updated earlier are be ignored (skipping old templates).
TemplateMatchThreshold	Gets or sets the value of the template recognition threshold.	<InArgument<Int32>	If the matching confidence is lower than this value, the learning script is ignored.
TemplateSampleFilter	Specifies the minimum number of template samples.	InArgument<Int32>	Any template that is sampled fewer times than defined is ignored.
Misc			
DisplayName	Activity name displayed in the Workflow Designer.	InArgument<String>	The default name is FreedomActivity.
Result	Output document recognition data.	OutArgument <IDocumentRecognition>	Contains the final document recognition data.

FieldOCR activity

The FieldOCR activity performs field recognition.

Group	Parameter	Description
Common	FieldOcrMode	Specifies FiledOcrMode to be recognized.
	FieldsSettingsMap	Specifies field settings.
	InputDRD	Input document recognition data. Cannot be empty in this activity.
	Pages	Gets or sets the array of pages that should be recognized. By default, the field OCR service is applied to all pages in the collection.
Events	PostEngine	The event is fired after field recognition.
	PostVote	The event is fired after applying the field OCR voting.
Misc	DisplayName	Activity name displayed in the Workflow Designer. The default name is FreedomActivity.
	Result	Output document recognition data.

PageOCR activity

The PageOCR activity performs full page recognition.

Group	Parameter	Description
Common	EngineName	Gets or sets the full-page OCR engine name.
	ImageEnhancement	Gets or sets an image enhancement string to be applied before page recognition.
	InputDRD	Input document recognition data.
	Pages	Gets or sets the array of pages that should be recognized. By default, full-page OCR is applied to all pages in the collection.
	Rotate_180	Gets or sets a value that indicates whether the page should be rotated by 180 degrees.
	Rotate_270	Gets or sets a value that indicates whether the page should be rotated by 270 degrees.
	Rotate_90	Gets or sets a value that indicates whether the page should be rotated by 90 degrees.
Misc	DisplayName	Activity name displayed in the Workflow Designer. The default name is FreedomActivity.
	Result	Output document recognition data.

Smart activity

The Smart activity is a wrapper for the eFlow Smart classification functionality.

Group	Parameter	Description
Common	FilteredClasses	Gets or sets the list of filtered classes within the specific model.
	Pages	Gets or sets the array of pages that should be recognized. By default, recognition is applied to all pages in the collection.
	SmartModelName	Sets or gets the Smart model name.
Misc	DisplayName	Activity name displayed in the Workflow Designer. The default name is SmartActivity.
	InputDRD	Input recognition data object.
	Result	Output document recognition data.

Image classifier activity

The Image Classifier activity performs image classification using machine learning algorithms based on the document structure.

Group	Parameter	Description
Common	ImageClassifier ModelName	Gets or sets the image classification model name.
	Pages	Gets or sets the array of pages that should be recognized. By default, recognition is applied to all pages in the collection.
Misc	DisplayName	Activity name displayed in the Workflow Designer. The default name is ImageClassifierActivity.

Group	Parameter	Description
	ImageClassifierResultClasses	The list of ImageClassifierOutput objects. Each object contains the name of the recognized class, its page index, and confidence. Unlike the Smart activity, the output is not saved in the collection DRD file.
	InputDRD	Input recognition data object.
	Result	Output document recognition data.

Voting activity

The Voting activity selects the best recognition result by comparing the document recognition data obtained from several sources.

Group	Parameter	Description
Common	Pages	Gets or sets the array of pages that should be included in the voting. By default, the Voting activity is applied to all pages in the collection.
Misc	DisplayName	Activity name displayed in the Workflow Designer. The default name is VotingActivity.
	InputDRDs	Gets or sets the list of input document recognition data objects. This parameter cannot be empty.
	Result	Output document recognition data.

FormType activity

The FormType activity defines the default form type that will be assigned to pages where recognition was not successful.

Group	Parameter	Description
Common	OnlyUnRecPages	
	Pages	Gets or sets the array of pages that should be recognized. By default, recognition is applied to all pages in the collection.
Misc	DisplayName	Activity name displayed in the Workflow Designer. The default name is FormTypeActivity.
	FormType	Gets or sets the default form type per flow.
	InputDRD	Input recognition data object.
	Result	Output document recognition data.

Regional Field OCR activity

The Regional FieldOCR activity applies field OCR to the specific region in the document.

Group	Parameter	Description
Advanced	ROIInfo	Specifies the field ROIInfo name.
Common	Engine	Gets or sets the field OCR engine name.

Group	Parameter	Description
	EngineIsVirtual	Gets or sets the value indicating whether it is a virtual OCR engine.
	FieldName	Gets or sets the name of the field to apply the OCR.
	Pages	Gets or sets the array of pages that should be recognized. By default, OCR is applied to all pages in the collection.
	Region	Gets or sets the region to apply the recognition to.
	TableRowIndex	Gets or sets the index of the row in the table.
Events	PostEngine	The event is fired after field recognition.
	PostVote	The event is fired after applying the field OCR voting.
Misc	DisplayName	Activity name displayed in the Workflow Designer. The default name is RegionalOCRActivity.
	InputDRD	Input recognition data object.
	Result	Output document recognition data.

FormSplitter activity

The FormSplitter activity creates new forms according to predefined conditions. If this activity is not included in the workflow, the forms and attachments remain the same as they were before recognition.

Group	Parameter	Description
Misc	DisplayName	Activity name displayed in the Workflow Designer. The default name is FormSplitterActivity.
	InputDRD	Gets or sets the list of the document recognition data objects. This parameter cannot be empty.
	Result	Output document recognition data.
SetAttachment	AttachmentAction	Specifies how to add the attachment.
	AttachmentFieldName	Gets or sets the name of the field that will be used to check if the page is an attachment.
	AttachmentFieldPattern	Gets or sets the regular expression used to evaluate the field value. If a match is found, this is the attachment page.
	AttachmentSeparators	Gets or sets the pages that will be the attachment starting points (such as {1,4,5}).
	AttachmentSetCriteria	Gets or sets the condition to add the attachments to the form.
	NumberOfPages	Gets or sets the number of attachment pages.
SplitForms	FormFieldName	Gets or sets the name of the field that will be used to check if the page should start a new form.
	FormFieldPattern	Gets or sets the regular expression used to evaluate the field value. If a match is found, this is the separator page.

Group	Parameter	Description
	FormSeparatorPages	Gets or sets the pages that will start new forms (such as {1,4,5}).
	FormSetCriteria	Gets or sets the condition to start a new form.
	NumberOfPages	Gets or sets the number of pages per form.

Form Criteria enumeration

This enumeration contains the conditions that should be met to create to create a new form. You must select one of these criteria when configuring the activity parameters.

Enumeration	Description
None	The pages are not be organized. Every form will contains one page only.
FieldValueChange	Creates a new form when the specific field content changes.
EveryNumberOfPages	
NumberOfForms	Every form will contain a predefined number of pages.
FieldPatternFound	The new form is created when the field content matches the pattern. The field name and regular expression describing the pattern are the required activity parameters.
FormTypeChange	The new form is created when the form type of the page changes.
CustomSeparators	The new form is created according to custom definitions.
Other	

CollectionSplitter activity

The CollectionSplitter activity splits the collections according to predefined conditions. If this activity is not included in the workflow, the collection structure remains the same as it was before the recognition.

Group	Parameter	Description
Collection	CollectionFieldName	Gets or sets the name of the field that is used to check if the page is a separator.
	CollectionFieldPattern	Gets or sets the regular expression used to evaluate the field value. If a match is found, this is the separator page.
	CollectionSetCriteria	Gets or sets the condition to split the collection.
	CollectionsSeparators	Gets or sets the array of form numbers. Each form in this list starts a new collection.
	EveryNumberOfForms	Gets or sets the number of forms per collection.
Misc	DisplayName	Activity name displayed in the Workflow Designer. The default name is SplittingActivity.
	InputDRD	Gets or sets the list of input document recognition data objects. This parameter cannot be empty.

Group	Parameter	Description
	Result	Output document recognition data.
Naming	CollectionNaming	Gets or sets the method of naming the collections after splitting.
	CustomCollectionNames	Gets or sets the custom collection names or suffixes. You can use a VB expression to get the names list.

CollectionSetCriteria enumeration

This enumeration contains the conditions that should be met to split the collection.

Enumeration	Description
None	The collection is not split.
FieldValueChange	The collection is split when specific field content changes.
EveryNumberOfForms	The collection is split after every N form.
FieldPatternFound	The collection is split if the field content matches the pattern. The field name and the regular expression describing the pattern are required activity parameters.
FormTypeChange	The collection is split if the form type changes.
CustomSeparators	The collection is split according to the defined custom separators.

CollectionNaming enumeration

This enumeration contains the methods of collection names assignment after splitting.

Enumeration	Description
None	Collection names are be created by the core mechanism.
Default	Collection names are be in the form [OriginalCollectionName]_[SeparatorValue]. For example, 00000016_EasyOrderForm if FormTypeChanged is selected.
Consecutives	Collection names include consecutive numbers after splitting. For example, [OriginalCollectionName]_[0002].
Suffixes	Suffixes are added to collection names.
FullNames	Custom collection names are used.

Tracing activity

The Tracing activity logs text messages.

Group	Parameter	Description
Misc	DisplayName	Activity name displayed in the Workflow Designer. The default name is TracingActivity.
	Message	Gets or sets the string value to be written in the log.
	PageNumber	Gets or sets the page number.

UserTag activity

The UserTag activity performs user tags manipulations.

Group	Parameter	Description
Common	RemoveExisting	Indicates whether existing data should be removed.
	UserTagName	Gets or sets the user tag name.
	UserTagValue	Gets or sets the user tag value.
Misc	CollectionData	Gets or sets the collection data. This parameter cannot be empty.
	DisplayName	Activity name displayed in the Workflow Designer. The default name is UserTagActivity.

SpecialTag activity

The SpecialTag activity performs special tag manipulations.

Group	Parameter	Description
Common	RemoveAll	Indicates whether all data should be removed.
	RemoveExisting	Indicates whether existing data should be removed.
	SpecialTagName	Gets or sets the special tag name.
Misc	CollectionData	Gets or sets the collection data. This parameter cannot be empty.
	DisplayName	Activity name displayed in the Workflow Designer. The default name is SpecialTagActivity.

Image classifier

This section explains how to prepare the Image Classification data model and use it at runtime in the Image Classifier recognition activity.

Image classifier activity

The **Image Classifier** activity performs image classification using advanced machine learning algorithms based on the image structure. Usually, this activity is placed in the Recognition workflow before Integra or Freedom activities to make the recognition process faster and more effective.

Image Classification algorithms perform the classification at runtime using the data model created at design time with the help of the Image Classifier Toolkit.

The activity checks occasionally whether the data model was changed. You can create a custom station to update the data model automatically, for example, if invoices from a new vendor were added.

Unlike Smart, this activity stores the results in output parameters only and does not save them in DRD or collection files.

See Built-in Recognition activities for more information about the **Image Classifier** recognition activity parameters.

Image classifier or Smart?

To a certain extent, the Image Classifier activity is similar to the Smart activity, however, these activities implement different classification algorithms, you should select the appropriate one depending on the project.

Image Classifier	Smart
Recognition methods are based on the document structure.	Performs classification using the document contents.
Use this activity if most of the images are quite similar. Example: To recognize invoices from the same vendor. Use it for Integra projects that contain many EFIs or when you need to separate forms and attachments so that the Integra activity is applied only to forms.	Use this activity when classification needs to be based on the recognized content. Example: To distinguish between different types of letters, such as CVs and complaints.

Image classifier toolkit

Overview

The Image Classifier Toolkit is used to configure the classification parameters and perform machine learning before applying the Image Classifier activity at runtime.

The Image Classifier Toolkit is not included in the LaunchPro tools. To run the Image Classifier Toolkit, start `<eFLOWInstallPath>\Bin\efImageClassifierToolkit.exe`.

The Image Classifier Toolkit can work with single or multipage TIFF images. If multipage images are used, the module splits them and treats them as single-page images. There are two ways to organize the images:

- Open one folder per image type. Select this option if the images are already separated, for example, when eFlow uses the output of another document management software.
- Put all the images in one folder and later use the **Automatic Learning** option.

Image classification process

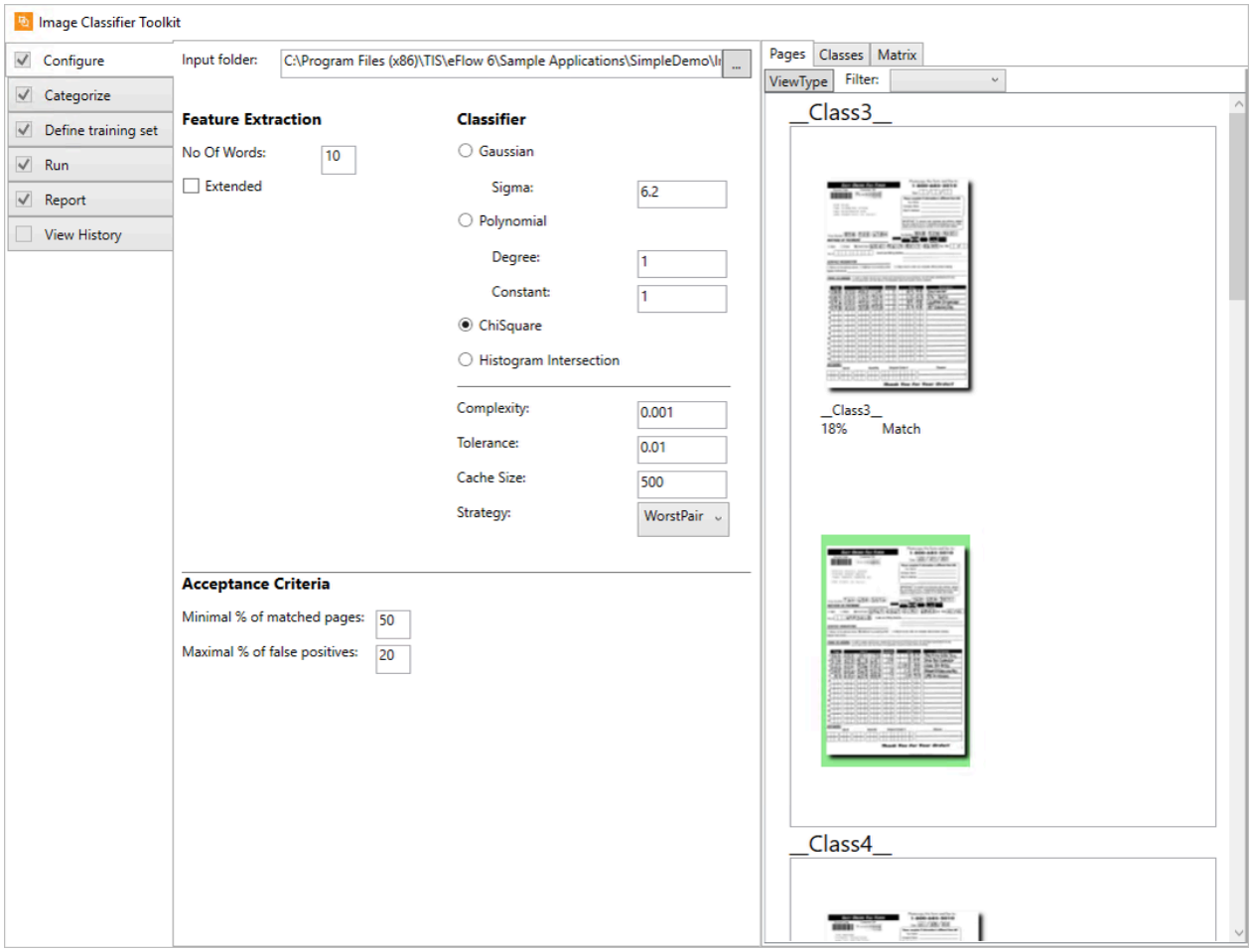
The Image classification process includes the following steps in the given order:

1. Configure
2. Categorize
3. Define training set

4. Run
5. Report
6. View History

Once the step is completed, it is selected in the upper left corner of the Image Classifier Toolkit dialog box.

Configure

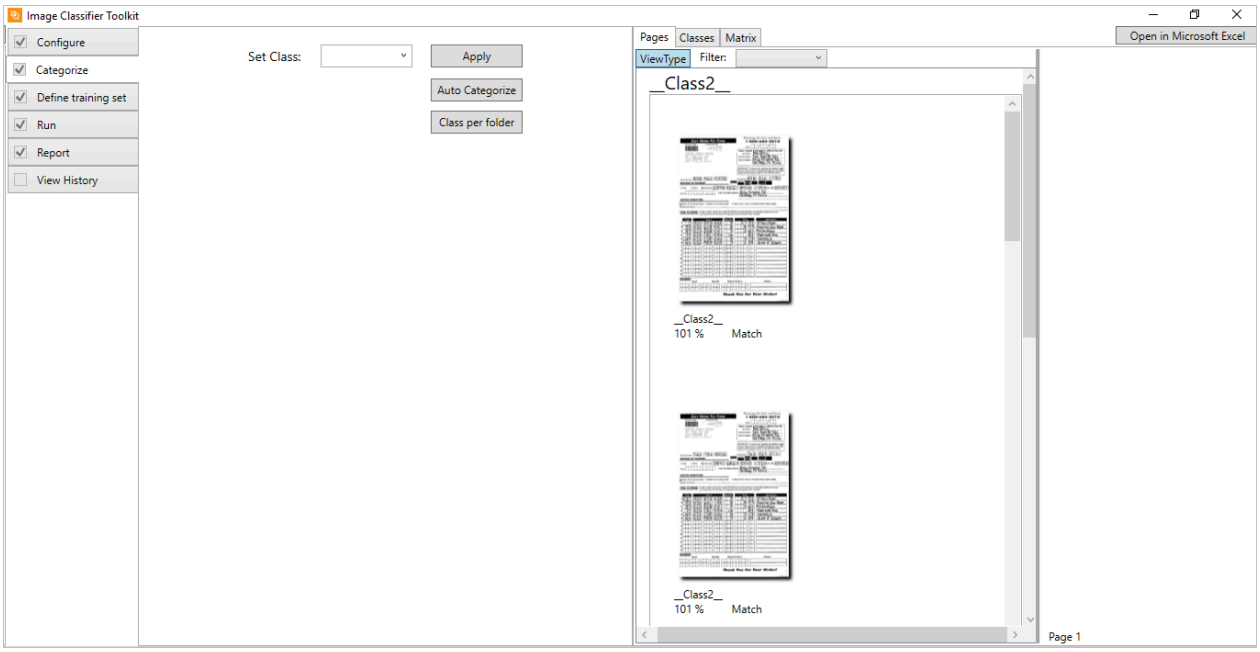


Define the basic parameters required to start the classification process as described in the following table.

Area	Parameter	Description	Notes
Feature Extraction	Input folder	Path to the folder containing the images to be used in the classification process.	If the images are already classified, this should be the path to the root folder. The subfolder names specify the class names. Once the path is defined, image thumbnails appear in the right-hand panel of the dialog box.
	No Of Words	Default value = 30	These parameters affect the algorithms used internally during the classification. We recommend that you use the default parameters to start the classification process.
	Extended		
Classifier	Gaussian		These parameters affect the algorithms used internally during the classification. We recommend that you use the default parameters to start the classification process.
	Polynomial		
	ChiSquare		
	Histogram Intersection		
Acceptance Criteria	Minimal % of matched pages	The minimum rate of documents that were classified correctly.	Make sure both criteria are met to build a successful classification model.
	Maximal % of false positives	The maximum percentage of documents that were erroneously classified as a different document ("False Positive").	

Categorize

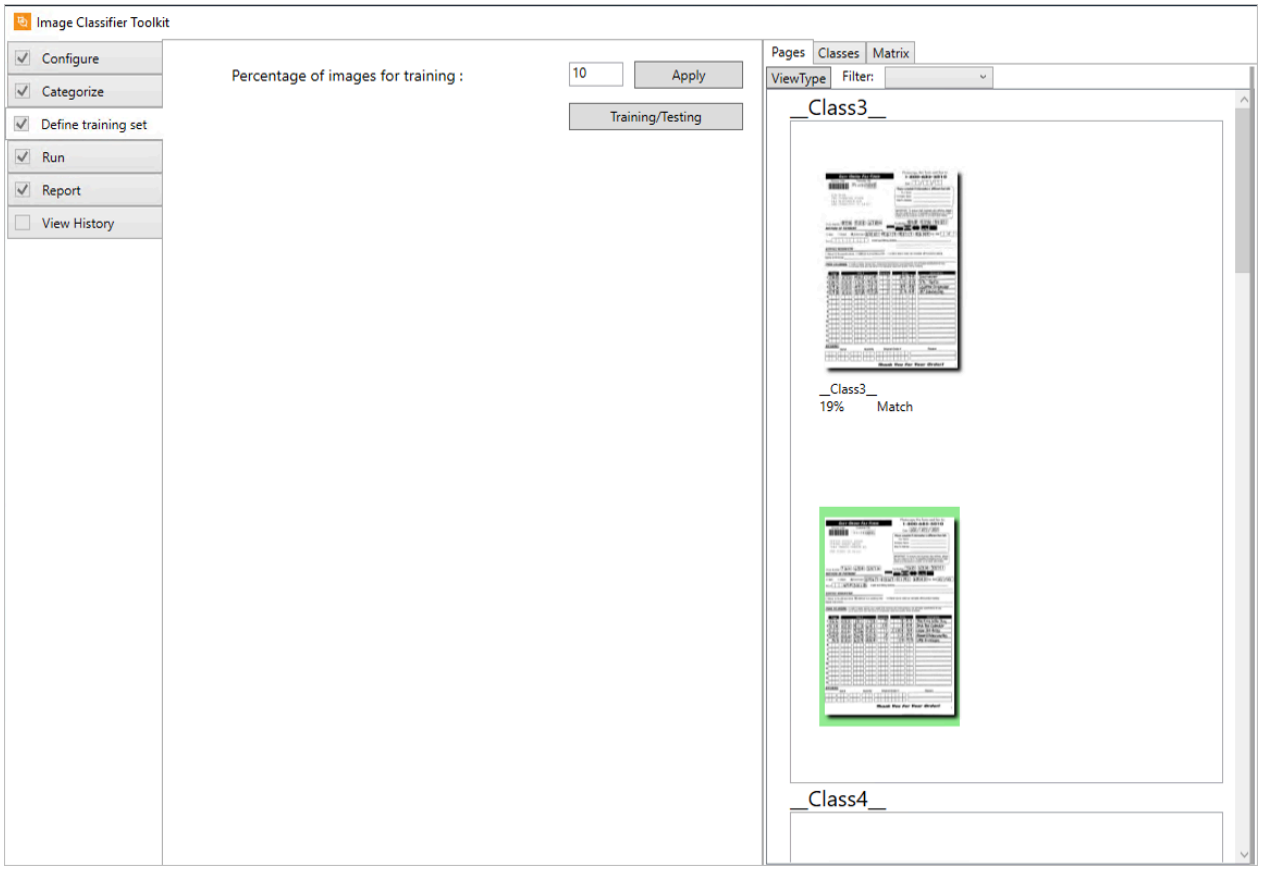
In this step, each image is associated with a class. The categorization can be done in different ways depending on the original images.



Parameter	Description	Notes
Auto categorize	The image classification algorithm will divides all images into classes. The result is displayed on the right side of the work area. The names of the classes are be determined by the tool.	This method does not provide the best results always and may require completing the process manually.
Set class	1. Select one or several images and associate them with one of the existing classes, or create a new class. 2. Click **Apply** to finish the classification.	This method is often used to manually fix errors created by automatic categorization, but can also be applied as the primary manual classification method.
Class per folder	The image classification algorithm will associate the classes with the images according to their folders.	Class names are derived from the image folder names. Use this method if the documents were previously manually classified and divided into separate folders.

Define training set

Define the image sets for training and testing the algorithms.

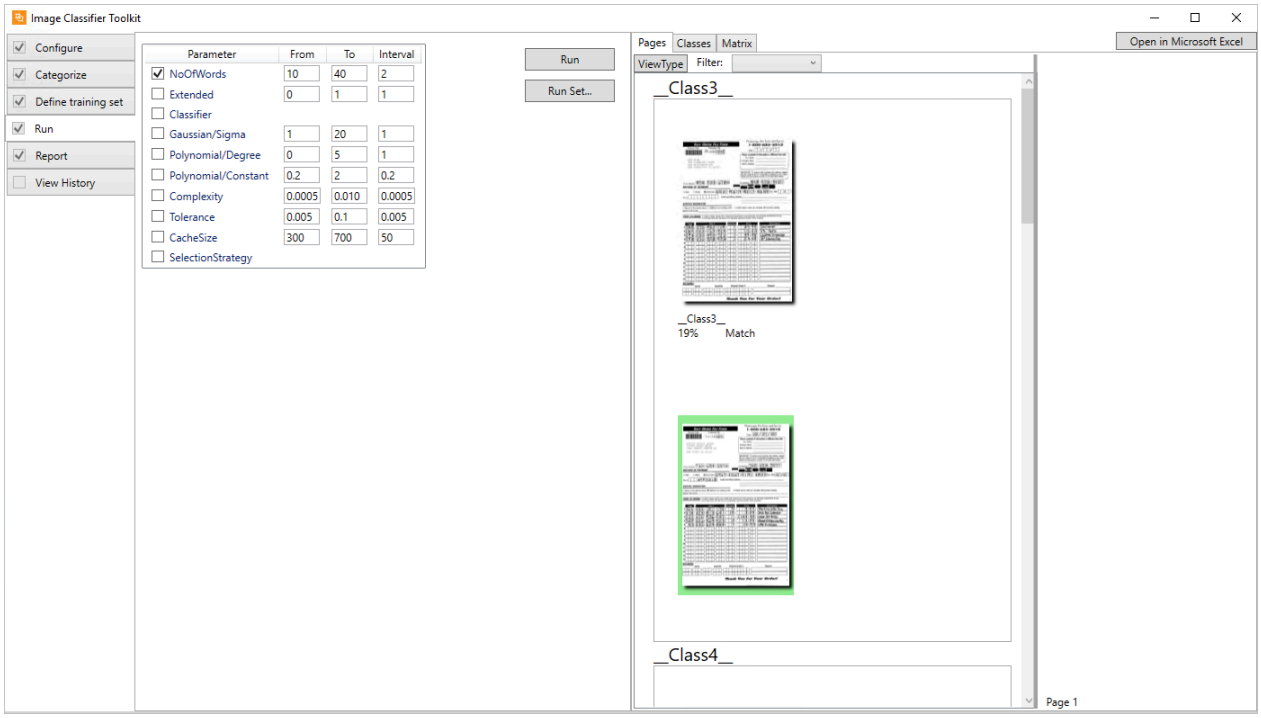


Parameter	Description	Notes
Percentage of images for training	Unlike Smart, the Image Classifier does not use different sets for training and testing the model. When running the classifier for the first time, the images are selected randomly depending on this setting. You can change this setting later and run the process again.	Training and testing sets are be created for each class separately.
Apply	The training and testing sets are created as per the settings.	Images selected for training in every class are surrounded by a green frame.
Training/Testing	You can revert a specific image from training to testing and vice versa, select the image and click this button.	Use this switch to select the relevant images for testing the changes in the model. Finally, 100% of images should go to training.

Run

You can performs the system training and testing based on the training and testing sets defined earlier. The results display the success rate of the automatic classification.

You can change the configuration parameters and rerun the step several times until you get the best results. When the classification model is ready, the results appear in the **Report** screen.

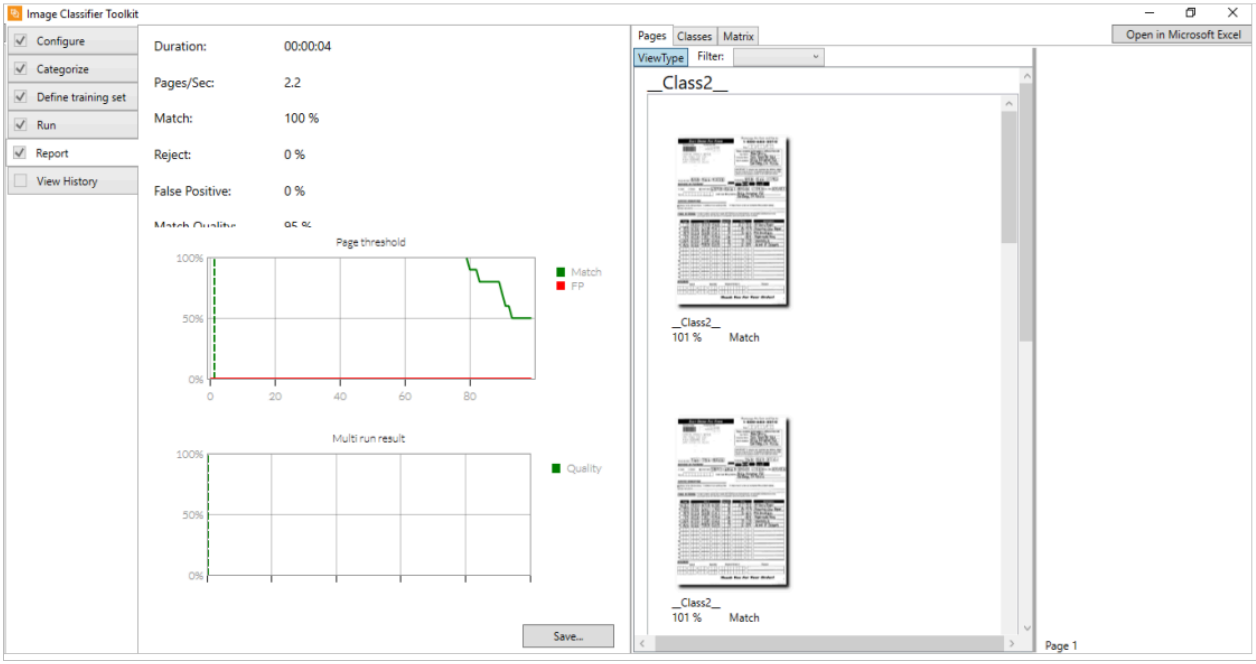


Parameter	Description	Notes
Run	Run the algorithm using the configuration that was defined at the beginning of the image classification process.	We recommend that you begin with the **Run** option and then continue with **Run Set**.
Run Set	This is an optimization process that runs set of tests, where the algorithm applies different parameters each time, according to **From**, **To**, and **Interval** values. The parameter set that provides the best result will be chosen.	

Report

You can analyze the quality of the image classification results and select the best configuration.

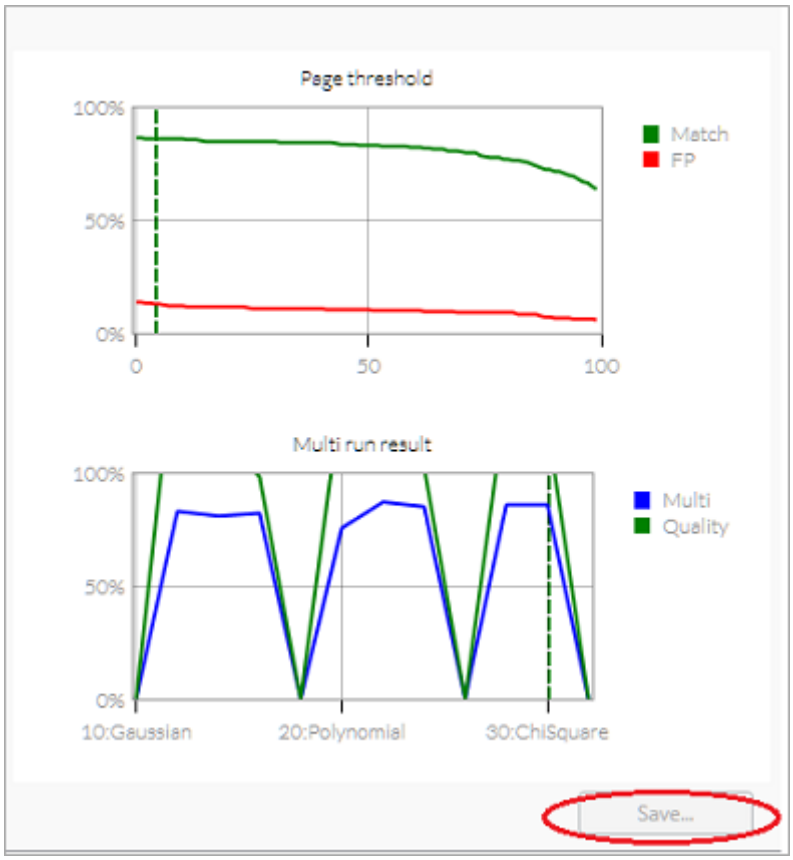
The results shown on this page are based on the images included in the testing set. The training set is used to create the classification model.



Parameter	Description	Notes
Duration	The total duration of the training and testing processes.	Measured in seconds.
Match	The percentage of recognized images.	In small data samples, the matching values of some images may exceed 100%.
Reject	The percentage of unrecognized images.	
False Positive	The percentage of false positive results.	This value should not exceed the allowed maximum defined in the configuration.
Match Quality	Provides an indication of the image classification results quality. Higher numbers correspond to better quality.	It is calculated as the average value of "good" matching results per run (those above the matching criterion).
Acceptance Criteria	Shows whether the test passed the acceptance criteria.	The value indicates whether the minimal matching and maximal false positives criteria are met.
Trained Pages	The total number of pages used for training.	

Parameter	Description	Notes
Tested Pages	The total number of pages used for testing.	

The results are also displayed in graphic form.



Page matching threshold

This graph contains the results of the current run only. It allows the evaluation of the data model quality.

Contains the graphs of matching and false positive rates per threshold.

The dotted line shows the maximal threshold value where both graphs answer the acceptance criteria, and the matching rate is as high as possible.

Multi run results

This graph consolidates the results of several runs. You select the configuration parameters and run the classification process on the testing image set. Every point on the graph corresponds to one run, while the total amount of runs is equal to the number of possible permutations of the selected configuration parameters.

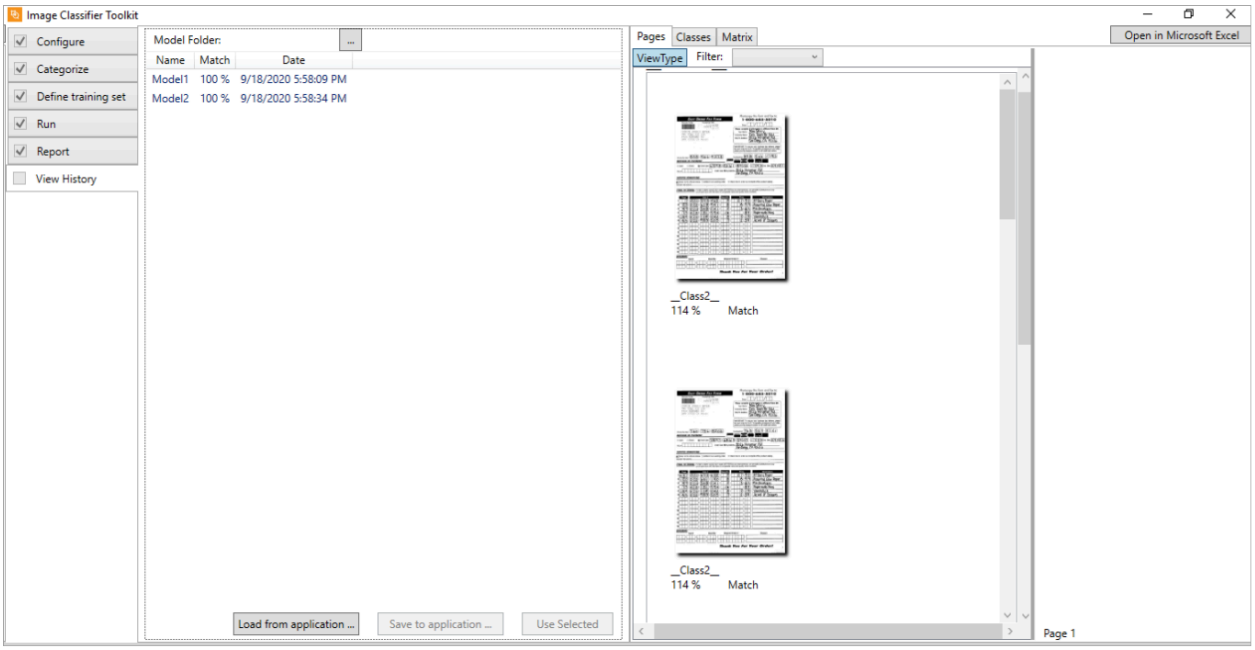
Every run is characterized by its image quality and matching. When you click the parameter combination in the **Parameter Value** combo box, the configuration is updated in the upper part of the page and the dotted line on the graph shows the results for the selected combination.

To select the best run automatically, click **Best** and the system will analyze the results and show the most successful configuration.

Once satisfactory results are obtained, you can save the model to add it later to the relevant eFlow application setup. This is done by selecting **Save** in the lower part of the **Report** screen.

View history

You can compare different models saved earlier and the best model to the setup of the relevant eFlow application.



Parameter	Description	Notes
Model Folder	Select the folder that contains the classification data models saved previously.	Once the folder is selected, a list of available models appear below the path.
Load from application	Get an image classification model associated with the specific application.	
Save to application	Add the model to the selected eFlow application.	The model is saved as a part of the application setup.

Parameter	Description	Notes
Use Selected	Select the model to use in subsequent runs in the toolkit.	The Image Classifier toolkit can only run one model at a time, so you must select the one you need from the list of available data models.

Backward compatibility

You can reproduce the old-style eFlow recognition station behavior by selecting a predefined eFlow workflow definition.

The XAML files required for backward compatibility are included in the standard installation at the following location.

`<eFLOWInstallPath>Customization Samples\Recognition Workflow\XAMLSamples.`

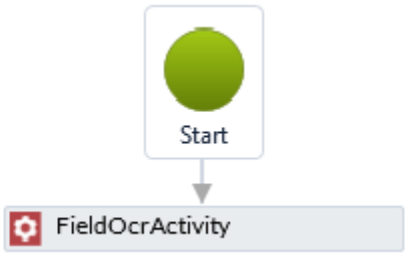
Form ID

To create a station that will perform structured forms recognition similar to the old Form ID station, select FormId.xaml in the station configuration dialog box.

FormId.xaml contains the Integra workflow activity that implements the Form ID station functionality.

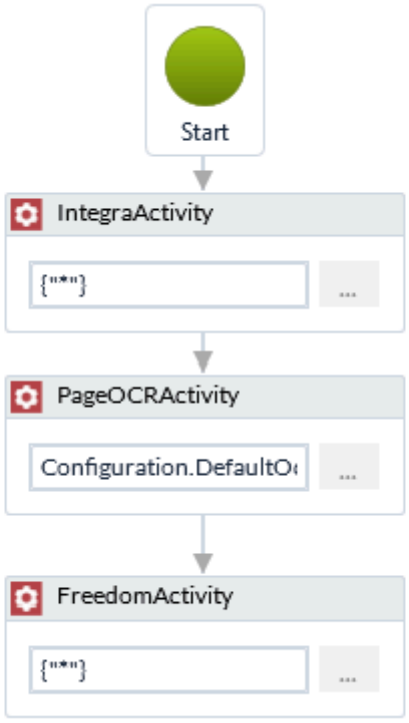
Free Process / Recognition

FreeProcess.xaml performs field OCR/ICR recognition. Select this XAML to replace the Free Process or the Recognition station.



Free match

FreeMatch.xaml represents the workflow that performs unstructured forms recognition. It invokes several activities in a row to reproduce the Free Match station behavior.

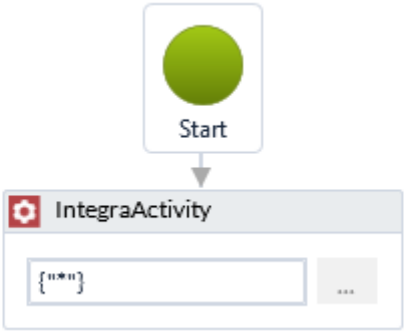


Page OCR

PageOCR.xaml performs full-page OCR recognition. It runs the default OCR page engine defined in the configuration.

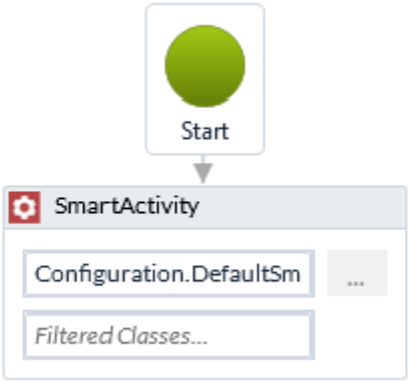
Processing

Processing.xaml contains a workflow that recognizes the structured form and then performs the field OCR recognition.



Categorizer

Categorizer.xaml runs the Smart activity. It replaces the Smart Categorizer station.



Custom Recognition activities

This section describes different ways to create custom recognition activities.

Create a new custom activity

If you are new to WPF, we recommend learning more about this technology before starting to write custom activities.

To learn how to create a new activity in Microsoft Visual Studio, refer to "How To: Create an Activity" on Microsoft Website.

1. Once you have created a new activity project, you need to add the relevant eFlow references:
   - RecognitionServices.dll
   - TiS.RecognitionWorkflow.Activity.dll
2. To get the XAML activity in the toolbox, ensure that the build action for this activity is set to XamlAppDef.
3. Drag a Flowchart or Sequence activity from the **Toolbox** to start editing your activity.
4. Add all the desired elements to the activity and compile the project.
5. Import XAML into your CAB using the eFlow Design module, as described in the Recognition workflow overview.

Recognition activities samples

The recognition activities samples are included at the following location.

```
C:\Program Files (x86)\TIS\eFlow 6\Customization Samples\Recognition Workflow
\VS2010
```

SampleActivities.sln consists of the following projects.

- TiS.Recognition.Basic

  This project contains XAML definitions of the basic recognition stations. See Backward Compatability for a detailed explanation.
- TiS.Recognition.Customization

  This project contains code samples and XAML files that can be used as part of the eFlow recognition customizations.
- TiS.Recognition.Advanced

  This project contains XAML files that can be used directly in the eFlow recognition station configuration.

The CabSample folder contains CAB and two sample images that can be used for testing.

You need not copy the customization libraries to the `<eFLOWInstallPath>\Bin` folder; they automatically become a part of CAB when the recognition XAML is imported into the workflow.

You can use these samples as a starting point to create a custom activity instead of creating it from scratch.

1. Open the `<eFLOWInstallBin>\Customization Samples\Recognition Workflow/VS2010/TiS.Recognition.SampleActivities.sln` file in Microsoft Visual Studio 2010 or later.
2. Make the following changes in `<eFLOWAppData>\Configuration\TISconfiguration.xml`:
   a. Set <UseConfigSource> to true.
   b. Make sure that <eFlowConfigSource> contains the full path to `<eFLOWInstallBin>\Bin\Default.exe.config`.

Chapter 6

Troubleshooting

This section describes the issues that you may encounter and their resolution.

eFlow client

Once the eFlow server-side is installed correctly, you must check the eFlow client-side installation.

Review the eFlow client-side basic configuration

1. Open `C:\Program Files(x86)\TiS\eFlow 6\Bin\ConfigSources\TisConfiguration.config`.
2. Check the validity of the parameters configured by the installation:
   - <LocalMachineName>
   - <ServerMachineName>
   - <eFlowDataPath>
   - <eFlowInstallPath>
   - <Communication protocol>

     The default value is https. Ensure that the protocol name is written correctly and is identical to the protocol selected on the server.

Verify the validity of the client-side configuration of WCF services

1. Open `C:\Program Files(x86)\TiS\eFlow 6\Bin\ConfigSources\BindingSection.config`.
2. Open `C:\Program Files(x86)\TiS\eFlow 6\Bin\ConfigSources\ClientSection.config`.
3. Make sure that the configuration is correct. For example, the server name is valid.

eFlow installs the default client configuration files together with those changed during installation. They can be a useful reference for troubleshooting.

eFlow certificate problems

The Event log may contain messages indicating eFlow certificate problems.

In this case, you may need to re-install the certificates.

eFlow licensing problems

eFlow only runs on a machine that has a serial number. eFlow uses the mainboard serial number for license encryption. On virtual machines, if the serial number is blank, you must check whether the machine has a serial number. To do so, enter the following command at the Windows Command prompt.

```
gwmi win32_baseboard
```

If no serial number is returned, contact the IT team and get the problem fixed.

SQL server

Open SQL Server Management Studio and log in with the SQL server user credentials defined in your project. By default, eFlow works with "sa" as the username and password.

1. If login fails, check your SQL server settings.
2. If the login succeeds, verify if the **eFlow_Management** database is created.

Testing the validity of this database is out of the scope of this checklist.

eFlow communication ports

Make sure that the ports used by eFlow are open for communication. See Communication ports.

User permissions

Verify that the eFlow services and interactive users have all the necessary permissions. See Set user permissions for details.

eFlow event logger

This section describes how to configure the eFlow event logger and use it effectively for troubleshooting.

Logger configuration

The eFlow logger messages appear in the **Event Viewer** under the **Applications and Services** with the name **TIS_Log**.

You can set the different logging parameters for each of the eFlow installations (server, client, and Web stations) in the logger configuration file (TisLogger.config) available in the `\Bin \ConfigSources` folder.

You can filter the eFlow events by their source:

- Select the **TIS_SERVER** event source to see eFlow server messages.
- Select the **TIS_APP** event source to see eFlow client and web application messages.

Logging levels

The eFlow event logger logs the events by referring to the following logging level list:

1. All (logs everything)
2. Debug
3. Info
4. Warn
5. Error
6. Fatal
7. Off (does not log anything)

This list contains event types in a sequence of levels that the event logger uses for appending. By default, only the Fatal and Error event types are logged. To log the events as per your requirements, you must change the default configuration in the TisLogger.config file located at `C:\inetpub \wwwroot\Tis Web Site\eFlow_6\Bin\ConfigSources`.

Normally, the eFlow event logger takes `root` value as a reference and logs the event types from `levelMin` (minimum) to `levelMax` (maximum) level as per the logging level list. The following are a few examples of to configuration the event logger.

- To log all event types from Debug to Fatal, set the values as shown below.

```
<root>
                <level value="All" />
                <appender-ref ref="EventLogAppender" />
</root>
```

```
<filter type="log4net.Filter.LevelRangeFilter">
                <acceptOnMatch value="true" />
                <levelMin value="Debug" />
                <levelMax value="Fatal" />
 </filter>
```

- To log only a specific type of event, set the values as shown below.

```
<root>
            <level value="Info" />
            <appender-ref ref="EventLogAppender" />
```

```
 </root>
```

```
<filter type="log4net.Filter.LevelRangeFilter">
                <acceptOnMatch value="true" />
                <levelMin value="Info" />
                <levelMax value="Info" />
 </filter>
```

- If you set the value of the `root` higher than the `levelMin` as per the logging level list, then the events are logged from `root` to `levelMax`.

```
<root>
            <level value="Error" />
            <appender-ref ref="EventLogAppender" />
 </root>
```

```
<filter type="log4net.Filter.LevelRangeFilter">
                <acceptOnMatch value="true" />
                <levelMin value="Info" />
                <levelMax value="Fatal" />
 </filter>
```

We recommend using the Info and Debug logging levels for testing or troubleshooting purposes only. Prolonged use of detailed logging may lead to an excessive number of logged messages which can significantly reduce system performance.

Write custom information to the TIS_Log

When customizing eFlow, if you want to add application specific data to TIS_Log, use the TISLogger available in the eFlow Core.

The following code samples show how to custom the event log.

```
#using
Tis.Core.TisCommon.Common
```

```
Log.Write(Log.Severity.ERROR, System.Reflection.MethodInfo.GetCurrentMethod(), "An
 error occurred");
```

```
Log.WriteError("Failed to save my object.")
```

```
Log.WriteException(oExc);
```

Troubleshooting checklist

This section explains how to troubleshoot if eFlow is not working properly.

Event viewer

Start troubleshooting by examining the Event Viewer (for more information, refer to "Event Viewer" on the Microsoft Website). This is the most important tool that helps you to understand the problem and to plan your next steps.

eFlow log messages appear in the **TIS_Log** node. See eFlow event log for details.

The **Application** and **System** logs in the **Windows Logs** node can be very useful for tracking generic errors.

eFlow server

Perform the following steps to verify if the eFlow server-side is installed properly.

Verify the eFlow server-side basic configuration

1. Open `C:\inetpub\wwwroot\Tis Web Site\eFlow_6\Bin\ConfigSources\TisConfiguration.config`.
2. Check the validity of the parameters configured by the installation:
   - Platform configuration
     - <LocalMachineName> = <ServerMachineName> - these names should be identical.
     - <eFlowDataPath>
   - SQL server configuration
     - <DBName>
     - <DBPassword>
     - <DBUserName>
     - <IntegratedSecurity>
     - <LocalSQLServer>
3. Open your Internet browser and access one of the eFlow services.

Skip this step if you have selected the net.tcp or net.pipe communication protocol.

For example, https://<FULL_SERVER_NAME>:55444/eFlow_6/DomainManagementServer.svc:

- https :<FULL_SERVER_NAME> must contain the domain name, for example, mymachine.mydomain.local.
- http :<FULL_SERVER_NAME> should contain the machine name only.

If you do not get any errors, the server-side eFlow is configured properly. Continue to check the client-side installation.

Open the IIS manager

1. **TiS Web Site** should be available in the list of IIS sites.
2. The **eFlow 6** and **TisDefaultSTS** web applications should be available under the **TIS Web Site**.
   If you only see the folder names but not the web applications (no web application icon), convert the folder to an application:
   Right-click the folder name and select **Convert to application**.
3. Right-click **eFlow_6 application** > **Manage** > **Browse**.
   If you cannot see the eFlow 6 folder content in the browser, inspect the error message that appears in the browser.
   Check for the errors in the **Event Viewer** under the **TiS_Log**, **Windows Logs\Application** and **Windows Logs\System** nodes.

4. Make sure that **TisAppPool** is configured properly and running.

   If the TisAppPool identity is not configured as expected, you must change this property.

   Right-click **TisAppPool** > **Advanced Settings** > **Identity** and set the relevant application pool identity.

5. If **TisAppPool** is stopped, try to restart it.

   If **TisAppPool** returns to the stopped state shortly after the restart, check for the errors in the **Event Viewer** under the **TiS_Log**, and Windows Logs\Application, and Windows Logs\System nodes.

6. Check the SSL certificate (**https** communication protocol only):

   Right-click **TiS Web Site** > **Edit Bindings** > **https**

   If the SSL certificate is not set, select the relevant one from the drop-down list.

   If the certificate selection fails, you must re-install the SSL certificate.

eFlow client

Once you know that the eFlow server-side is installed correctly, it is time to check the eFlow client-side installation.

Go through the eFlow client-side basic configuration

1. Open `C:\Program Files(x86)\TiS\eFlow 6\Bin\ConfigSources\TisConfiguration.config`.
2. Check the validity of the parameters configured by the installation:
   - <LocalMachineName>
   - <ServerMachineName>
   - <eFlowDataPath>
   - <eFlowInstallPath>
   - <Communication protocol>

The default value is https.

Make sure that the protocol name is written correctly and is identical to the protocol selected on the server.

Check the validity of the client-side configuration of WCF services

1. Open `C:\Program Files(x86)\TiS\eFlow 6\Bin\ConfigSources\BindingSection.config`.
2. Open `C:\Program Files(x86)\TiS\eFlow 6\Bin\ConfigSources\ClientSection.config`.
3. Make sure that the configuration is not corrupted (for example, it may contain an invalid server name).

eFlow installs the default client configuration files together with those changed during installation. It can be a useful reference for troubleshooting.

eFlow certificate problems

The Event log may contain messages indicating that there are some eFlow certificate problems.

In this case, you may need to re-install the certificates.

eFlow licensing problems

eFlow uses the mainboard serial number for license encryption. On virtual machines, if the serial number is blank, check whether the machine has a serial number by entering the following command at the Windows Command prompt.

```
>gwmi win32_baseboard
```

If no serial number is returned, request the IT team. eFlow to fix the problem, as eFlow cannot run on a machine without a serial number.

SQL server

Open SQL Server Management Studio and log in with the SQL server user credentials defined in your project. By default, eFlow works with "sa" as username and password.

1. If login fails, check your SQL Server settings.
2. If the login succeeds, check if the **eFlow_Management** database is created.

Testing the validity of this database is out of the scope of this checklist.

eFlow communication ports

Make sure that the ports used by eFlow are open for communication, as described in Communication ports.

User permissions

Check that the eFlow services and interactive users have all the necessary permissions. See Set user permissions for details.

Under the hood: Installation

This section describes the settings that are normally created automatically by the eFlow installation.

Registry settings

The eFlow installation adds a new key to the system registry.

- 64-bit machines: `HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\TopImageSystems\eFlow 6`
- 32-bit machines: `HKEY_LOCAL_MACHINE\SOFTWARE\TopImageSystems\eFlow 6`

This key contains the following values.

Value	Description	Notes
Version	The installed eFlow version number.	Used by the TIS About dialog box.
eFlow	The path to the eFlow client installation.	This information helps to find the location of the configuration files when upgrading eFlow.
OCRs	The path to the eFlow OCR installation.	When eFlow is installed or upgraded, this value is written in the client configuration file.

Basic configuration parameters

Installation paths, SQL server settings, and other such configuration parameters are set during the eFlow installation and saved in the relevant TisConfiguration.config files. You can change them later if needed.

For details, see Configuration files.

Web Services configuration

Server-side configuration

The eFlow Web.config file is available in the TIS Web Site physical folder and contains configuration data for the following services:

1. eFlow Web services
2. eFlow license activation service
3. eFlow security service (ClaimsService)

It also includes the <serviceAutoStart> configuration section containing the eFlow services that need to be started automatically.

Client-side configuration

Configuration files for all eFlow executables should contain a Web services configuration. By default, all executable files share the same settings for the <binding>, <behavior>, and <client> configuration sections. Instead of replicating these sections in all configuration files, separate configuration section files are created. They are installed in the `<eFlowInstallPath>\bin\ConfigSources` folder.

Configuration files of all the eFlow executable modules include the following references.

```
<system.serviceModel>  <bindings
configSource="ConfigSources\BindingSection.config"></bindings>  <client
configSource="ConfigSources\ClientSection.config"></client>  <behaviors
configSource="ConfigSources\BehaviorSection.config"></behaviors>  </
system.serviceModel>
```

The eFlow installation modifies the server name in the default configuration files. In standalone installations, it is set to 127.0.0.1 and in client installations, you can set it to the server name or its IP address. You can also change other endpoint configuration properties if required.

Original configuration files are saved with the extension .old to provide a reference if something goes wrong during the installation, or if the configuration files are changed manually.

IIS configuration

TIS website

The eFlow installation creates a new website named Tis Web Site and two web applications under this site: eFlow_6 and TisDefaultSTS

The application name eFlow_6 is hardcoded in eFlow Core and cannot be changed.

TIS website bindings

You select the eFlow communication protocol during the installation.

The **https** communication protocol is selected by default.

A self-signed SSL certificate is created by the installation and connected to the **https** settings.

You can select any other SSL certificate installed on the server machine, and the eFlow installation will connect it to the https settings.

If the website bindings are not configured properly during the installation, or you need to change the port number, you can do so either using the IIS manager or with the appcmd.exe (For more information, refer to "IIS AppCmd Quick Reference" on the Microsoft website) command line utility.

Application pool identity

The eFlow installation creates a new application pool named TisAppPool.

You select the application pool identity during the installation.

By default, TisAppPool runs under the ApplicationPoolIDentity identity. In this case, IIS creates a local user account [IIS AppPool\TisAppPool.

The application pool identity can be changed after the installation if needed.

Refer to "Specify an Identity for an Application Pool (IIS 7)" on Microsoft Website for detailed information about changing the application pool user.

File system and registry access permissions

Server

1. eFlow application data folder: The installation grants full access to the TisAppPool user account and domain users.
2. eFlow server folder (TIS Web Site physical path): the TisAppPool user account should have full access.

Client

eFlow application data folder: Manual station users and the eFlow Autorun Starter service account should have full access.

SQL server login

The eFlow installation runs the script creating a new SQL server login. This user is given db_owner permissions for the eFlow management and workflow database.

Authentication mode	User
Windows	TisAppPool
SQL server	SQL user

Debug permissions for TisAppPool user

The TisAppPool user should be granted debug permissions on a standalone machine.

STS and eFlow server certificates

eFlow installs default certificates both on the server and client side. If the certificates are not installed properly, you can verify them by checking the certificates store directly or analyzing the event log. In this case, you may have to reinstall the certificates.

Server

Install TiS-eFlow-rp.pfx and TiS-eFlow-rpsts.pfx in the `LocalMachine\Personal certificate` store. These certificates contain both private and public keys. The TisAppPool user account must have full access permissions to these certificates.

The original certificates are located under `<PATH_TO_TIS_Website>\TisDefaultSTS\Certificates`.

Client

The TiS-eFlow-rp.cer and TiS-eFlow-rpsts.cer certificates are installed in the `LocalMachine\TrustedPeople certificate` store. These certificates contain a public key only.

The original certificates are located under `<eFLOWPath>\Certificates`.

Default STS configuration

The installation adds the current user credentials to `DomainSecurity.xml under \Server \System\Setup`. If, after installation, the DomainSecurity.xml does not contain the installing user credentials, it should be edited manually. For more information, see Default STS configuration.

You can start working with eFlow using the default STS provided with the installation; however, we strongly recommend that you do not use it for real projects. You must adjust the security model to your project requirements as soon as possible.

Login failures

This section describes different situations when eFlow login fails, and possible reasons and solutions for this issue.

This content is constantly updated, and your input is highly appreciated.

Introduction

The error message shown in the following image can appear in different situations and is not informative. You must explore the event log carefully to understand the underlying problem.

Invalid security settings

If logged error messages show that security settings were not configured properly and authentication failed, you must check the security configuration.

1. Verify that the value of <AuthenticationType> element in TisConfiguration.config is **Windows** or **User**.

   This file is available at:

   `C:\inetpub\wwwroot\Tis Web Site\eFlow_6\Bin\ConfigSources`
2. Make sure that the user is defined correctly in DomainConfiguration.xml available at the following location:

`<AppData>\Server\System\Setup`

- The role assigned to this user exists in the Design module.
- The username is available in the node where the application name is empty or in the specific application.
- The user also appears in the **System** application.

3. If **User** authentication is selected, verify the validity of the UsersId.xml file.

See eFlow Security Mechanism for more information about security settings.

System application was not created

1. In the **eFlow_Management** database, verify that the **AppName: System** entry is available in the **E_App** table.

   If the table is empty, it means that the system application was not created.
2. Stop **TisAppPool** (go to **IIS** > **Application pools**).
3. Delete the databases **eFlow_Management** and eFlow_Monitor.
4. Start **TisAppPool** (go to **IIS** > **Application pools**).
5. Verify that the **AppName: System** entry is available in the **E_App** table. If the entry is available, it means that the system application is created properly.

FIPS encryption

One of the possible sources of login failure can be FIPS encryption. To correct this, you must disable the **Local Security Setting System cryptography: Use FIPS compliant algorithms for encryption, hashing, and signing** policy in Windows. For more information, refer to "FIPS 140-2" on Wikipedia Website.

To resolve this issue:

1. Go to **Start** > **Control Panel** > **Administrative tools** > **Local Security Policy**.

   The **Group Policy** dialog box opens.
2. Select the **Local Policies** node and then **Security Options**.
3. Look for the relevant entry: **System cryptography: Use FIPS compliant algorithms for encryption, hashing, and signing**.

   If this entry is enabled, disable it.
4. Restart the machine.

Missing service principal name in configuration

Symptoms

Log in to the client application fails when TisAppPool is running under the domain account. The following error is written in the event log:

```
Login fail for user "DOMAIN\USER"- Security Support Provider Interface (SSPI)
authentication failed.
```

```
The server may not be running in an account with identity 'host/web-web-01'.
```

```
If the server is running in a service account (Network Service for example), specify
the account's ServicePrincipalName
   as the identity in the EndpointAddress for the server. If the server is running in
a user account, specify the account's
   UserPrincipalName as the identity in the EndpointAddress for the server.
```

Solution

1. If TisAppPool is running as a service account, use the Setspn command line utility to add an SPN:

   ```
   setspn -A HOST/WEB-WEB-01:55443 MYCOMPANY\MYNAME
   ```

   For more information, refer to "How to use SPNs when you configure Web applications that are hosted on Internet Information Services" on Microsoft Website.

   WEB-WEB-01 is the server name, 55443 is the port, and the last part is the domain and user account that is used in IIS for **TisAppPool**.

2. Open BindingSection.config on the client machine and add the <userPrincipalName> tag:

   ```
    <issuer address="https://WEB-WEB-01.mycompany.local:55443/
   TisDefaultSts/TisDefaultSts.svc/Windows" binding="ws2007HttpBinding"
    bindingConfiguration="ws2007eFlowSts_Windows">
    <identity>
    <dns value="localhost" />
    <userPrincipalName value="MYUSER@MYCOMPANY.local"/>
    </identity>
   </issuer>
   ```

Error (417) Expectation Failed Symptoms

Symptoms

- The remote client cannot log in to any eFlow application. The following error is received:

  ```
  2015-01-27 10:54:17,590 [1] FATAL Source App: efEnterpriseManager.exe, Message:
  Login fail for user: 'DOMAIN\USER'
       - The remote server returned an error: (417) Expectation Failed.
  TISAppenderLog4net.Log4NetException: The remote server returned an
       unexpected response: (417) Expectation Failed. --->
  System.ServiceModel.ProtocolException: The remote server returned an
       unexpected response: (417) Expectation Failed. ---> System.Net.WebException:
  The remote server returned an error: (417)
       Expectation Failed.
  ```

- Using a web browser, you can connect to all server services, including ClaimService, STS, and so on.

Reason

The problem may appear when a proxy server is used. For more information, refer to "HTTP POST Returns Error: 417 "Expectation Failed"" on Stackoverflow Website.

Possible solutions

- Do not use a proxy server.
- Add the following section to all *.exe.config files or to Default.exe.config in the client Bin folder:

```
<system.net>
<settings>
<servicePointManager expect100Continue="false" />
</settings>
</system.net>
```

TisAppPool user password was changed

Symptoms

TisAppPool was working normally but then it stopped working without any apparent reason.

Reason

TisAppPool is running under the domain user account, and the password of this account was changed.

Solution

Change the new password in the TisAppPool service configuration and restart it.

Server error messages

This section describes common error messages related to the eFlow server.

Error "no https handler available"

While troubleshooting the eFlow installation you will try to browse to any of the service *.svc files, as described in the Troubleshooting checklist. Sometimes you will get error 404: No https handler available to process this request.

Error details: The page you requested has a file name extension that is not recognized and is not allowed.

Open the IIS Manager and check the existing IIS handler mappings for TiS Web Site.

Normally you should see at least one of three *.svc mappings:

- svc-Integrated handling *.svc - handler is System.ServiceModel.Activation.HttpHandler.
- svc-ISAPI-2.0-64 handling *.svc - handler this time is IsapiModule.
- svc-ISAPI-2.0 handling *.svc - handler is also IsapiModule.

If you do not have these handler mappings, you must add them.

Try registering WCF from the command prompt by running the following command:

```
%\Microsoft.NET\Framework\v3.0\Windows Communication Foundation\ServiceModelReg.exe /
i /x
```

You must run the command prompt as administrator.

Error "Could not load type 'System.ServiceModel.Activation.HttpModule'"

If you install .NET framework 4.0 on the IIS server and then enable .NET 3.0 or 3.5 WCF features, the following error may appear when browsing the TiS website.

```
Could not load type 'System.ServiceModel.Activation.HttpModule' from assembly
'System.ServiceModel, Version=3.0.0.0,
   Culture=neutral, PublicKeyToken=b77a5c561934e089'. Description: An unhandled
exception occurred during the execution of the current
   web request. Please review the stack trace for more information about the error and
where it originated in the code.Exception
   Details: System.TypeLoadException: Could not load type
'System.ServiceModel.Activation.HttpModule' from assembly
   'System.ServiceModel, Version=3.0.0.0, Culture=neutral,
PublicKeyToken=b77a5c561934e089'.
```

To resolve this issue, run the following command from the Command prompt:

```
aspnet_regiis.exe /iru
```

aspnet_regiis.exe utility location:

```
%windir%\Microsoft.NET\Framework\v4.0.30319 (32-bit machine)

%windir%\Microsoft.NET\Framework64\v4.0.30319 (64-bit machine)
```

For more information, refer to the following article Error message after you install the .NET Framework 4.0: "Could not load type 'System.ServiceModel.Activation.HttpModule'" on Microsoft Website.

Error message "Server is too busy"

Verify that **TisAppPool** is running properly.

Troubleshooting eFlow stations

This section is based on recently solved support cases. We believe that this information may help many eFlow customers.

Design

Multipage forms are not supported

Design does not allow you to define a form that contains more than one page.

Reason

This functionality became obsolete with the introduction of the Recognition workflow, which provides much more flexibility than was available before.

Solutions

If the real paper document consists of multiple pages (for example, census questionnaires never fit one page), you can select between the straightforward approach and more sophisticated ones:

- Create a form with one page, add all fields from all document pages, and link all the EFIs to this page.
- Use form and collection separators in input stations to prepare forms or collections that contain pages from one document. This approach can work well only if the order and number of pages in the original documents are always as expected.
- Create a custom recognition activity that implements some validation mechanisms in addition to page recognition. This activity will split and merge the documents according to the recognition results.

Input

Station fails to split/merge collections

It is not possible to split or merge collections that contain an Input form.

Reason

This behavior is by design. The Input form contains information that is relevant to the specific collection only, and merging or splitting collections will possibly make this information irrelevant.

Collect: Some collections are not sent to the server

When a user attempts to import the same image file second time, the station issues an error:

StationWork.SendAllBatchesInternal : Failed sending collections to the server.

Error: The transaction has aborted.

Reason

If the KeepBatchName property is selected in the flow settings in Design, it prevents the system from sending collections with the names that already exist on the server.

Organize

Workflow design limitation

In a workflow, you cannot place the Organize station between two Recognize stations or between the Recognize and Tile stations.

Reason

Organize clears field data created by Recognize. Since the DRD file does not contain field data, stations that rely on this data are not able to work properly.

Validate

B/W or color images

The Validate station always displays color JPG images attached to the current collection in .PRV (compressed preview) format, so that downloading heavy images can be ignored as possible.

Solution

To see the B/W TIFF (high resolution) images in the Validate station, select one of the following options:

- Use the **OnLogin** event to call the **SwitchToJpeg** command. This will toggle the image type selection as the station starts.
- Change the settings in Design: **Workflow** > **Validate station** > **Properties** > **Get Attachment Types**. Ensure .TIFF appears instead of .JPEG.

Recognize

Freedom script field definition

The candidate list shows that the character was recognized correctly as **S**, but the actual result is **5**.

Reason

If the field definition in the Freedom script contains the **Numeric** field type, eFlow tries to adjust the recognition results to the field type, so **S** becomes **5**.