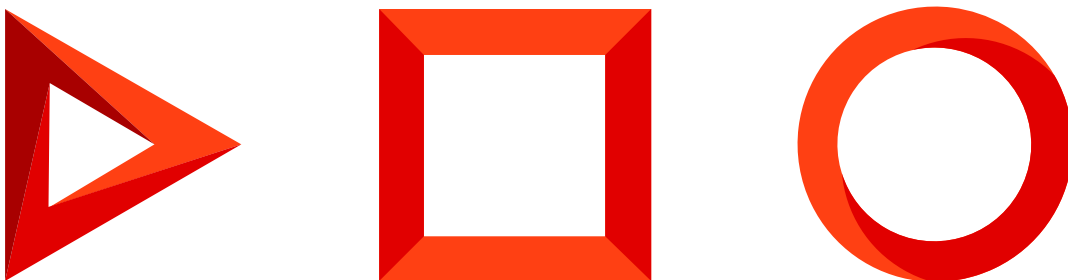


.NET Framework application server on Windows

Version 8.0



This documentation is provided under restrictions on use and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this documentation, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

Table of Contents

Enable required Windows components	4
Enable required Windows Components on Windows Server 2016	5
Enable required Windows Components on Windows 10	15
Modify ConnectionStrings.config for MS SQL Server	16
Set up ConnectionStrings.config	16
Required ConnectionStrings.config settings	17
Optional ConnectionStrings.config settings	18
Modify ConnectionStrings.config for Oracle Database	19
Set up ConnectionStrings.config	20
Required ConnectionStrings.config settings	20
Optional ConnectionStrings.config settings	21
Modify ConnectionStrings.config for PostgreSQL	21
Set up ConnectionStrings.config	21
Required ConnectionStrings.config settings	22
Optional ConnectionStrings.config settings	22
Modify Web.config	23
Additional Web.config setup for Oracle	24
Set up Creatio to work with Oracle 19c	24
Set up Creatio application server on IIS	26
Set up application website in IIS	26
Add an application pool	28
Set up WebSockets	34
Check WebSocket settings for Windows Server 2012 or Windows Server 2016	35
WebSocket setup FAQ	36
Switch a Creatio website from HTTP to HTTPS	37
IIS setup	37
Web.config setup	42

Enable required Windows components

PRODUCTS: **ALL CREATIO PRODUCTS**

Make sure that you install the following components on the web server before you create and set up a website:

- Windows components. Note that the Microsoft Visual C++ 2010 component is required.
- Web Server IIS components.

To ensure correct compilation of the application, download and install .NET 6 SDK and .NET Framework SDK v 4.7.2.

Grant permissions to read, create and delete files and subfolders of the \Terrasoft.WebApp\Terrasoft.Configuration catalog to the user who runs the application pool in IIS.

[Download 64-bit .NET 6 SDK](#)

[Download 64-bit .NET Framework SDK v 4.7.2](#)

Attention. Production environment for Creatio .NET Framework application requires a Windows Server OS. You can deploy application server on Windows 10 only for development and pre-production environments.

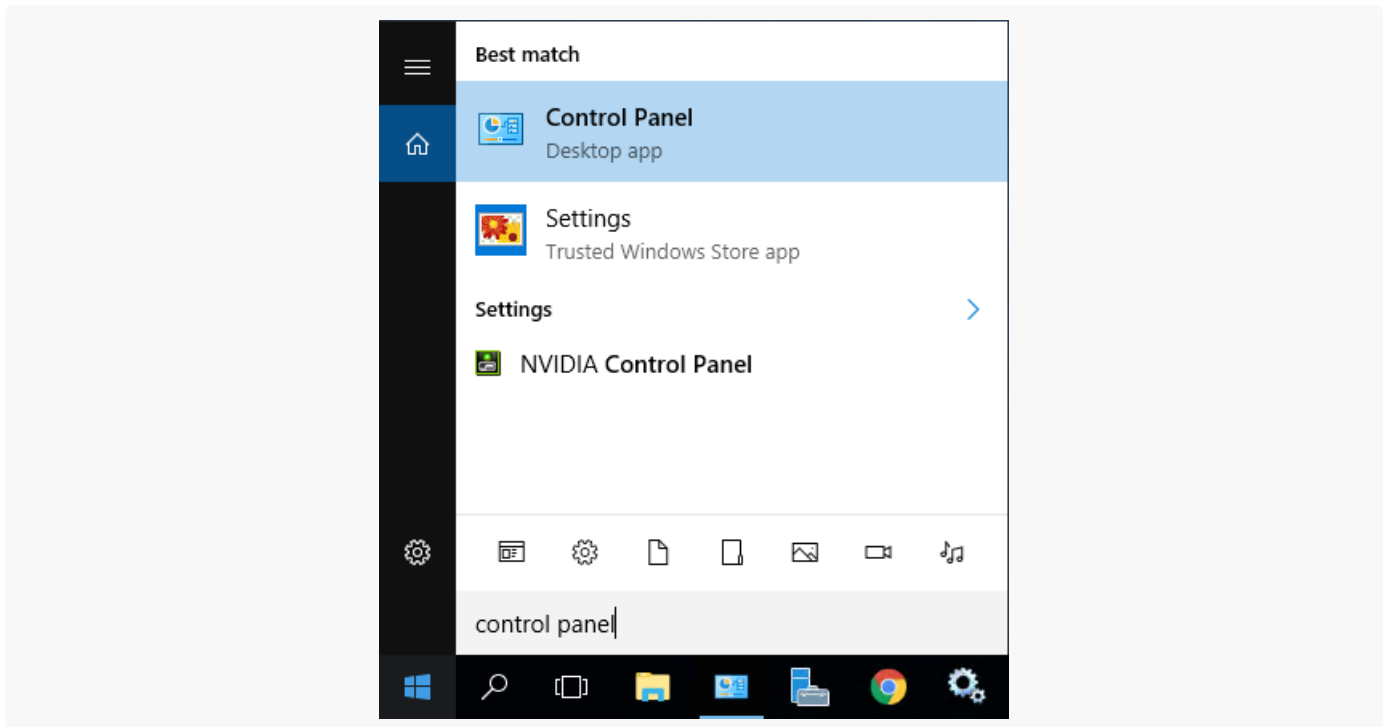
Component	Component items
Common HTTP Features	Static Content Default Document HTTP Errors HTTP Redirection
Application Development	ASP.Net .Net extensibility ISAPI extensions ISAPI Filters WebSocket Protocol
Microsoft .Net framework 3.5.1	Windows Communication Foundation HTTP Activation Windows Communication Foundation Non-HTTP Activation
Microsoft .Net Framework 4.7 Advanced Services and up (Windows 8, Windows 10, Windows Server 2012, Windows Server 2016).	ASP.NET 4.6.2 or 4.7; WCF services HTTP Activation Message Queuing (MSMQ) Activation Named Pipe Activation TCP Activation TCP Port Sharing
Health and Diagnostics:	HTTP Logging Logging Tools Request Monitor Custom Logging
Security	Basic Authentication Request Filtering IP and Domain Restriction

Enable required Windows Components on Windows Server 2016

To check the availability of the needed components:

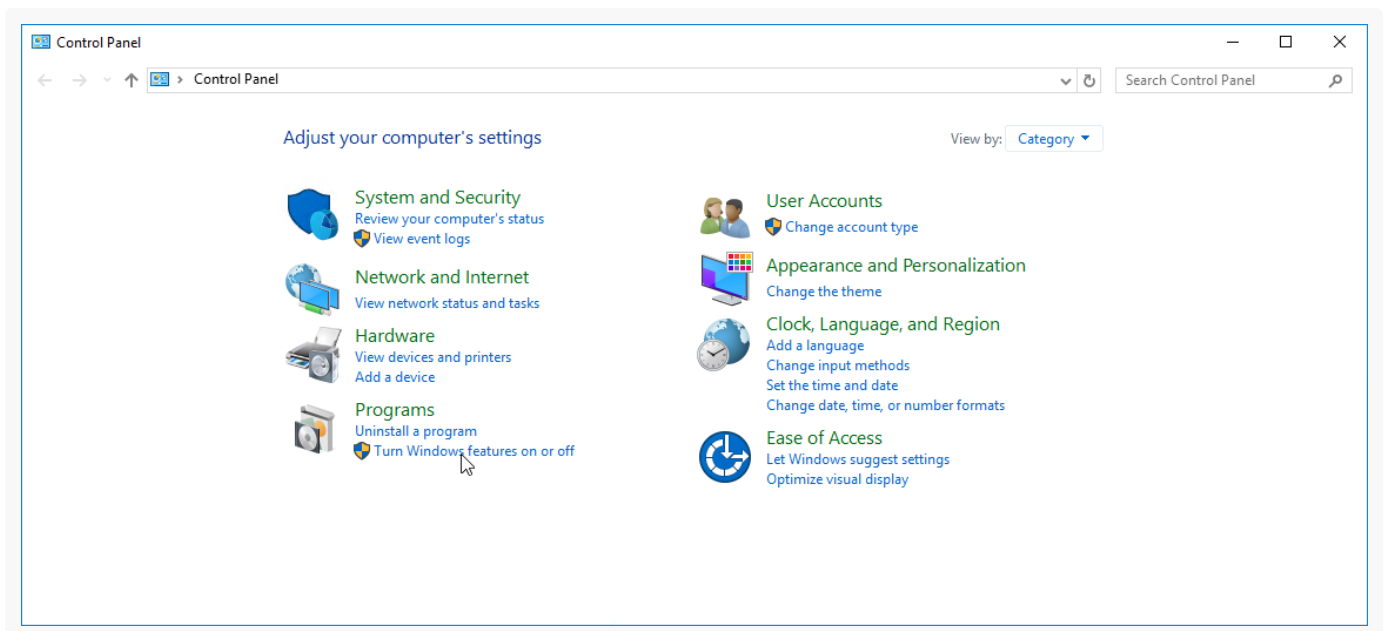
1. Enter the “control panel” in the [*Start*] menu and select the [*Control Panel*] (Fig. 1).

Fig. 1 The [*Control Panel*] section in the [*Start*] menu



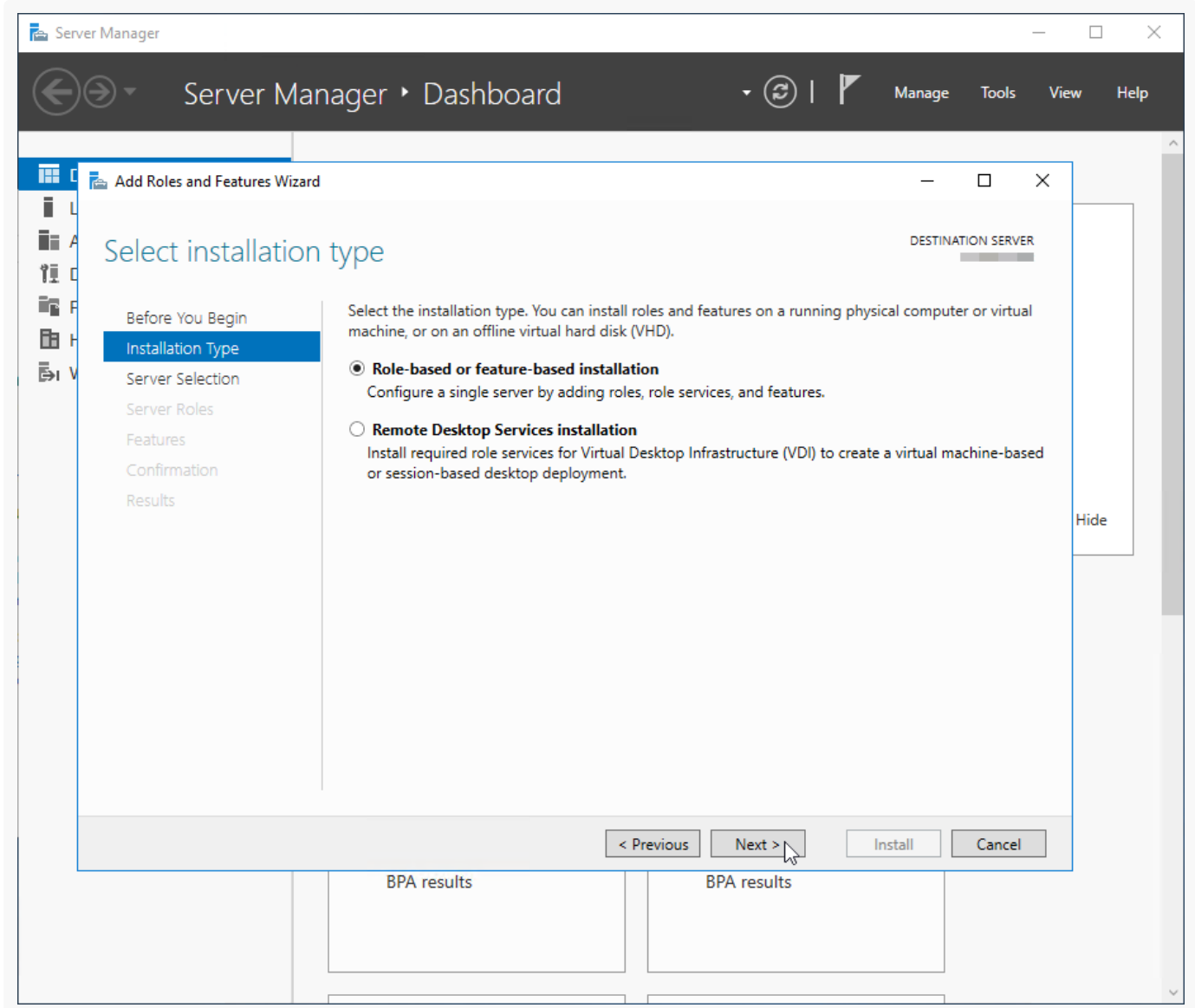
2. From the [*Control Panel*], select the [*Turn Windows features on or off*] option (Fig. 2).

Fig. 2 Selecting the [*Turn Windows features on or off*] option



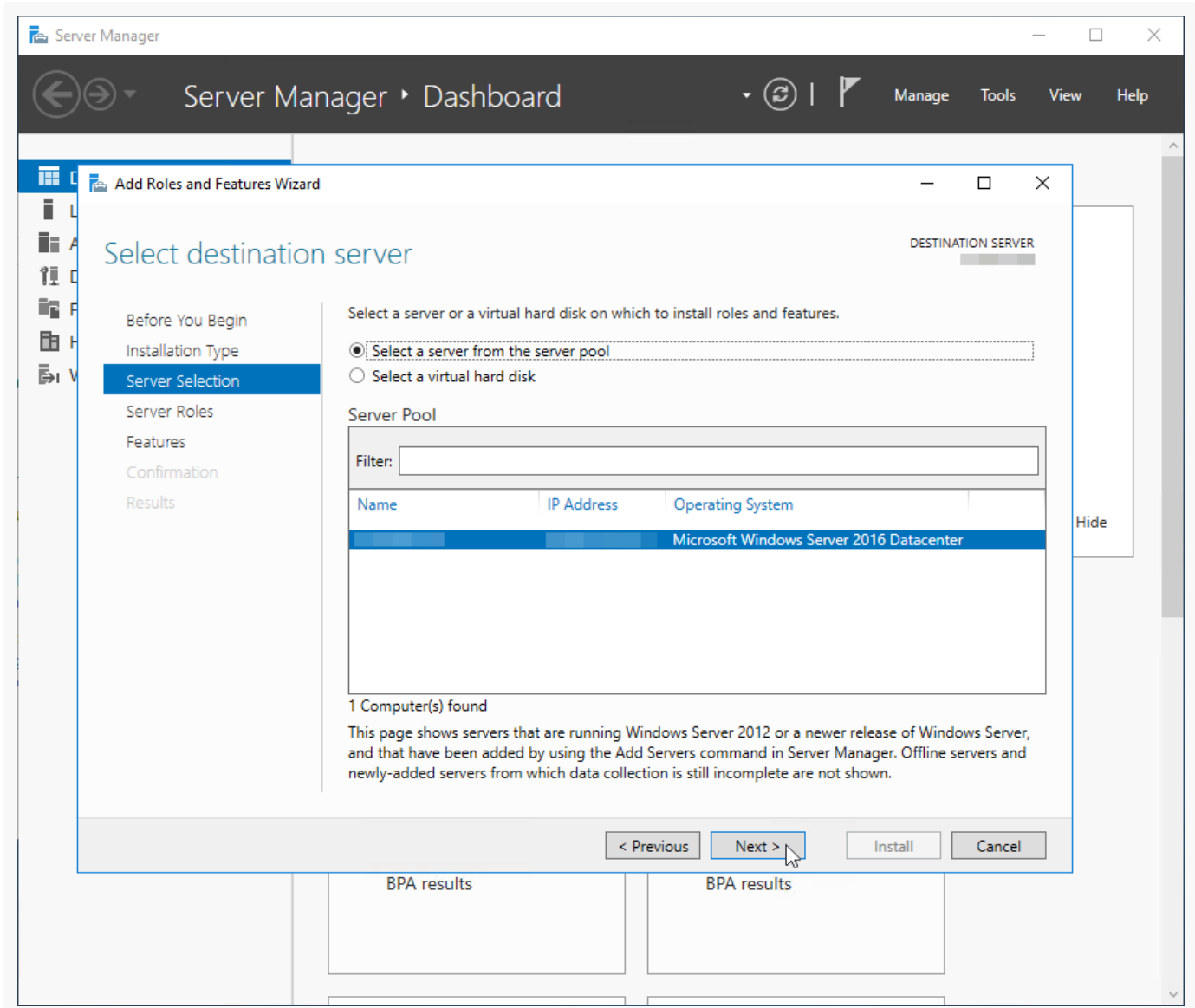
3. In the [*Add Roles and Features Wizard*], select [*Role-based or feature-based installation*] → [*Next*] (Fig. 3).

Fig. 3 Selecting the role-based installation



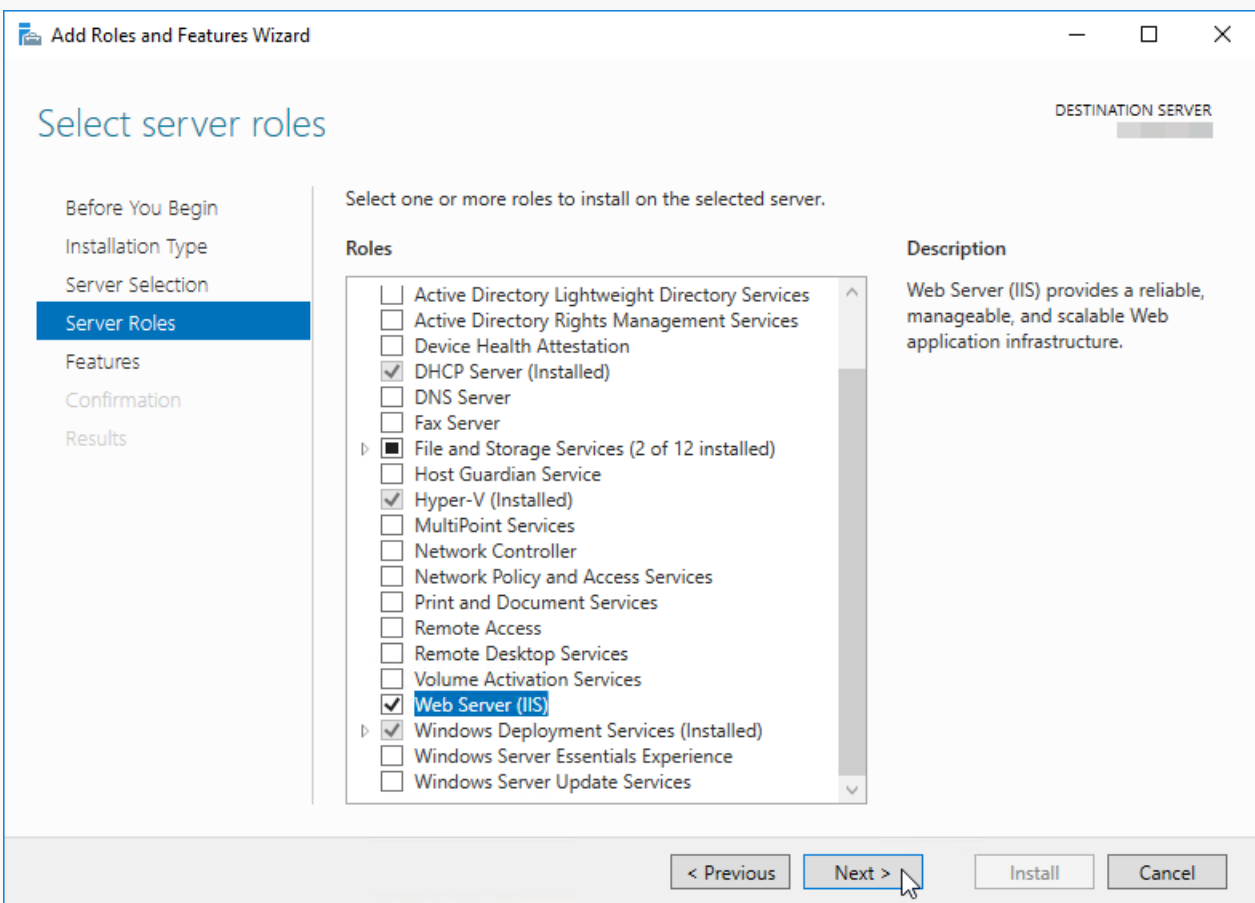
4. Select the destination server from the available server pool and click [Next] (Fig. 4).

Fig. 4 Selecting the destination server



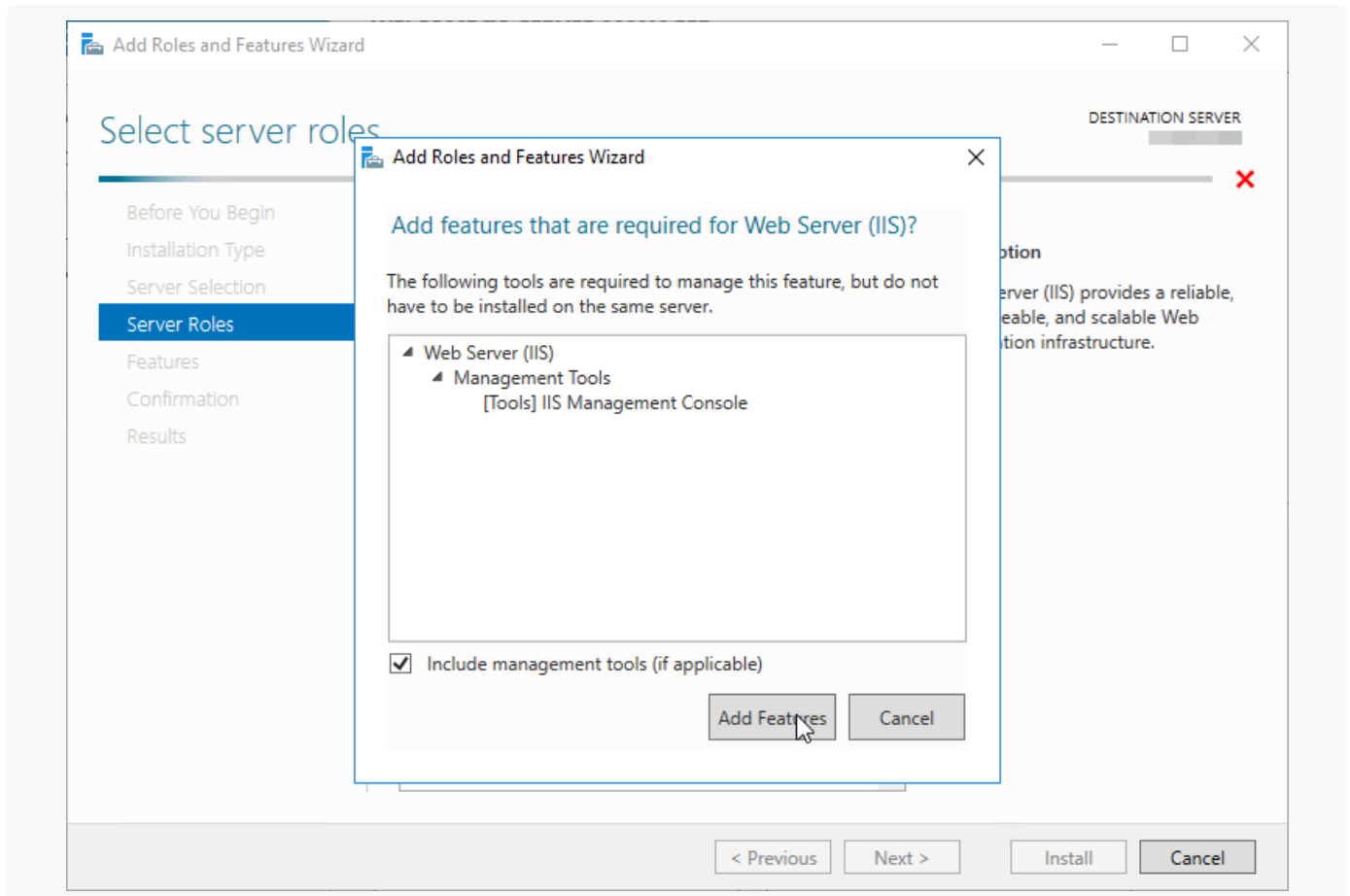
5. Select the Web Server (IIS) role to apply to the selected server. Click [Next] (Fig. 5).

Fig. 5 Selecting the Web Server (IIS) role



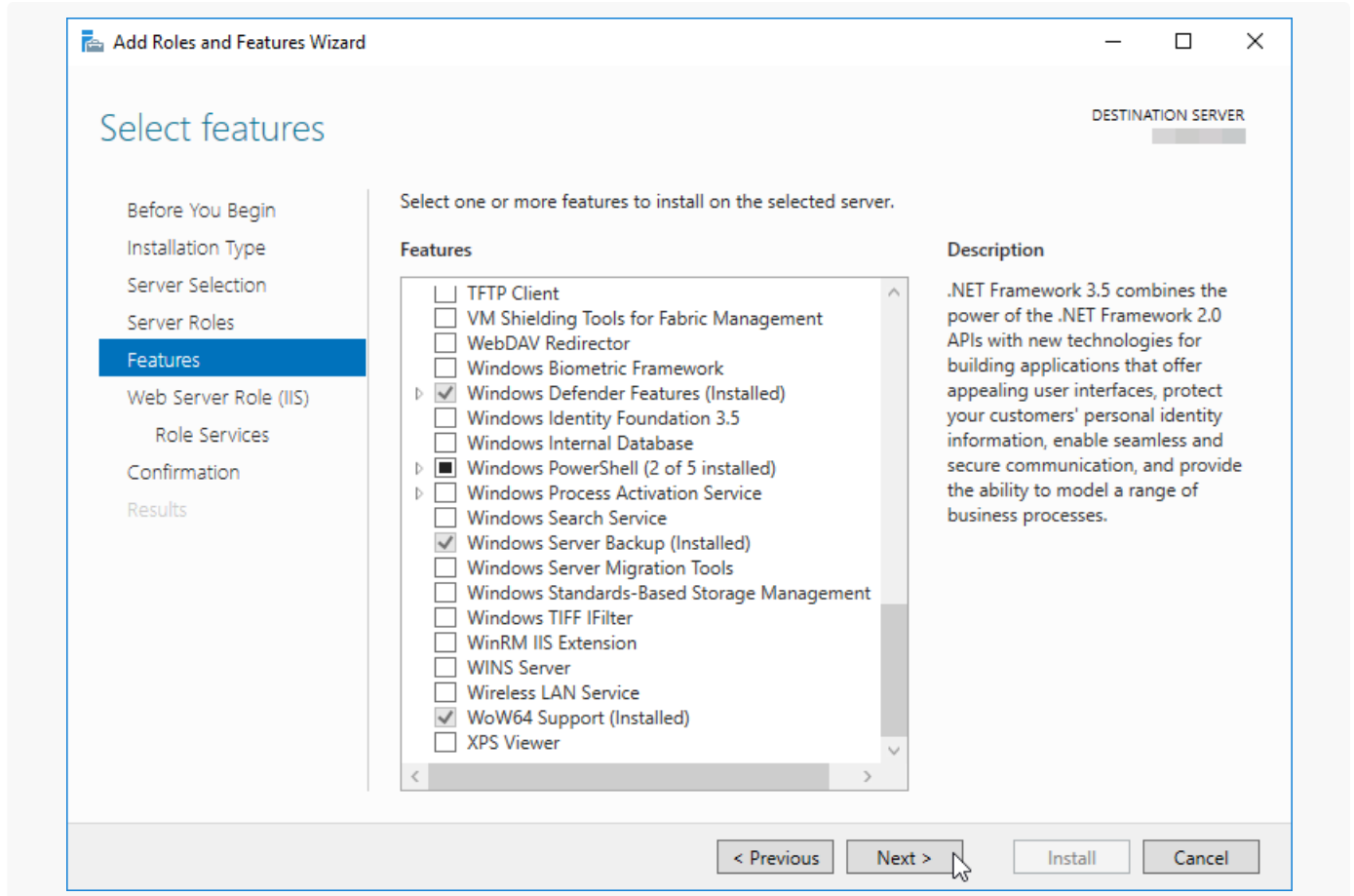
6. Click [Add features] (Fig. 6).

Fig. 6 Confirming selected features



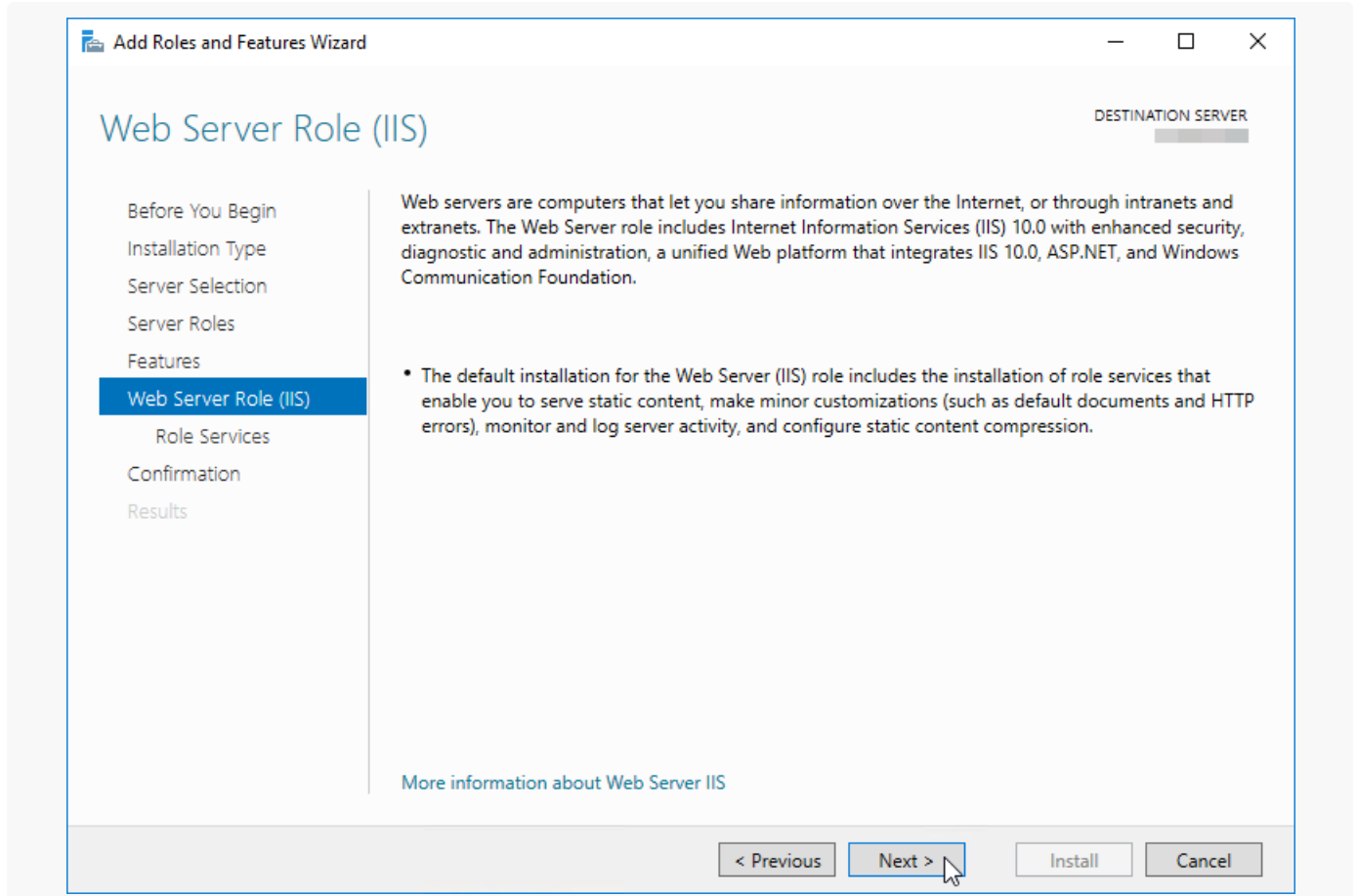
7. Select features and click [*Next*] (Fig. 7).

Fig. 7 Selecting features



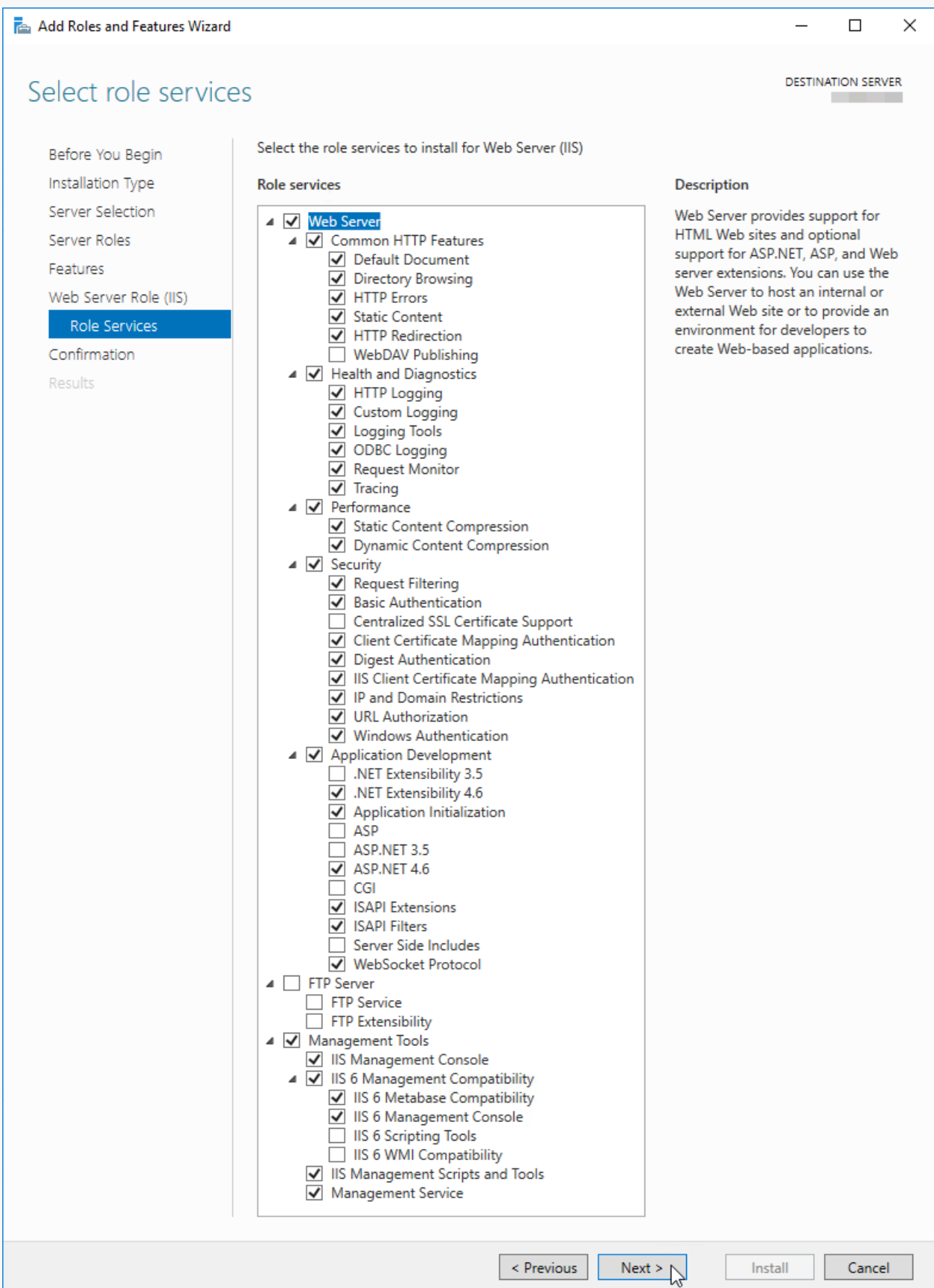
8. Click [*Next*] to finish proceed to the next step ().

Fig. 8 Confirm the web server role



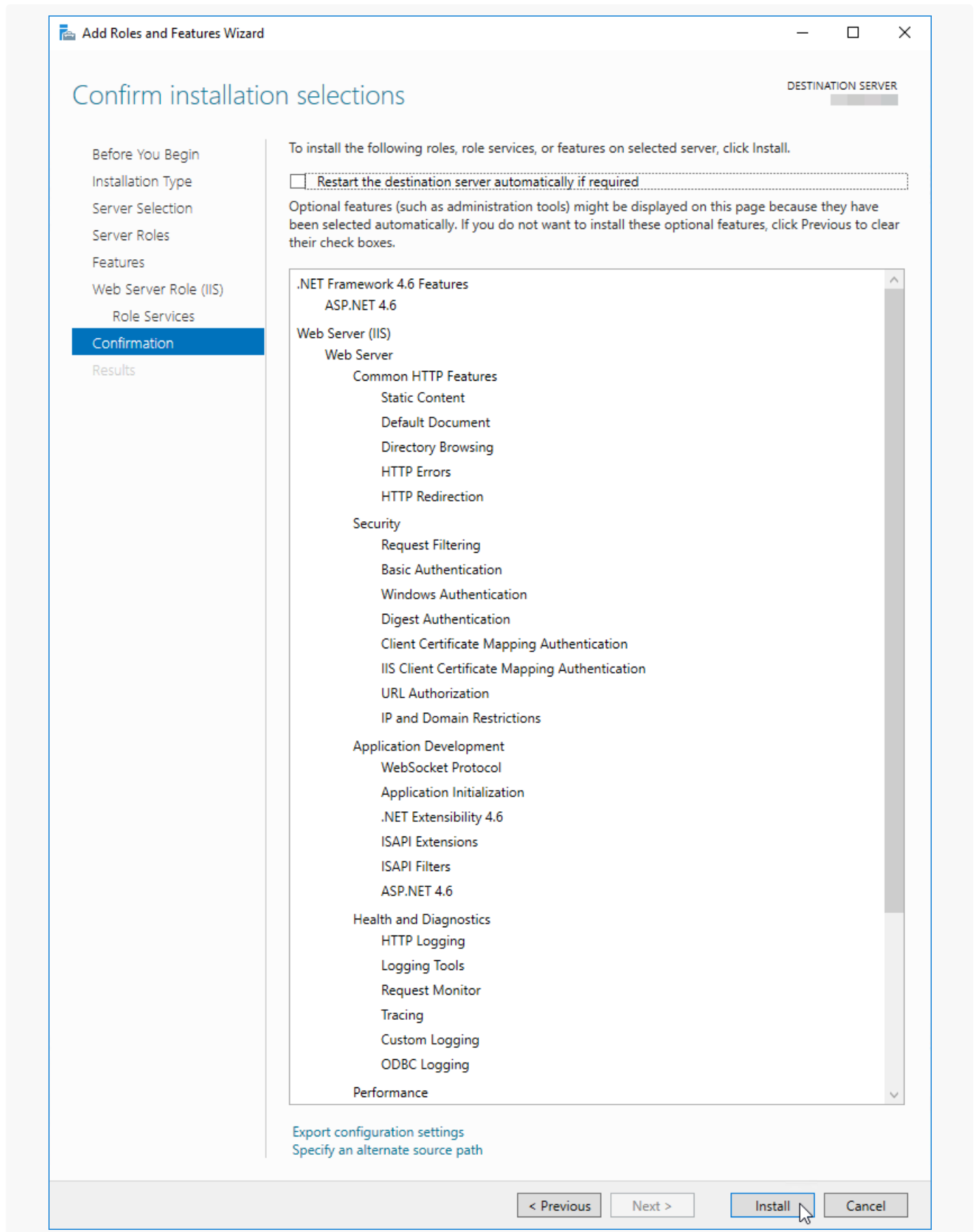
9. Make sure that the you have the same components selected as on the picture (Fig. 9).

Fig. 9 Required components



10. Click [*Install*] (Fig. 10).

Fig. 10 Confirm installation



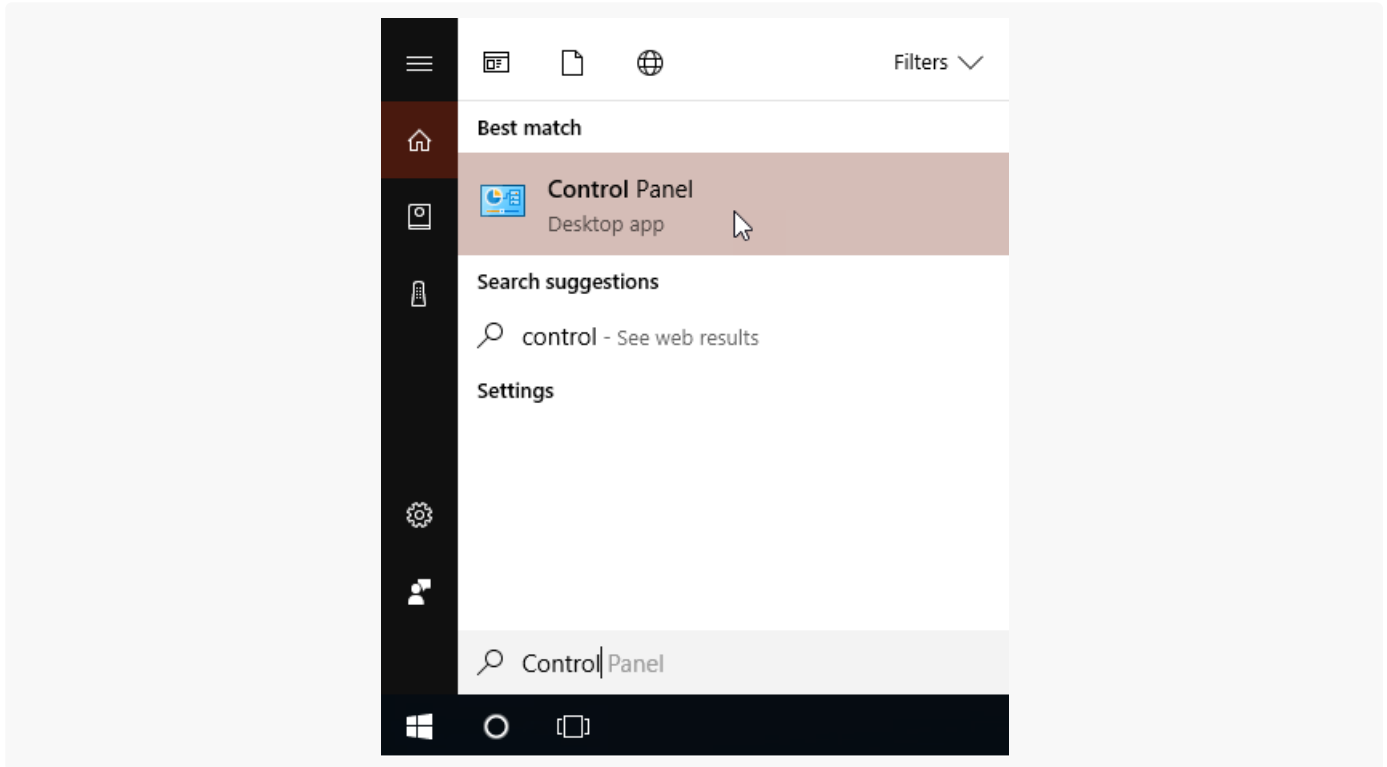
11.Reboot the server.

Enable required Windows Components on Windows 10

To check the availability of the needed components:

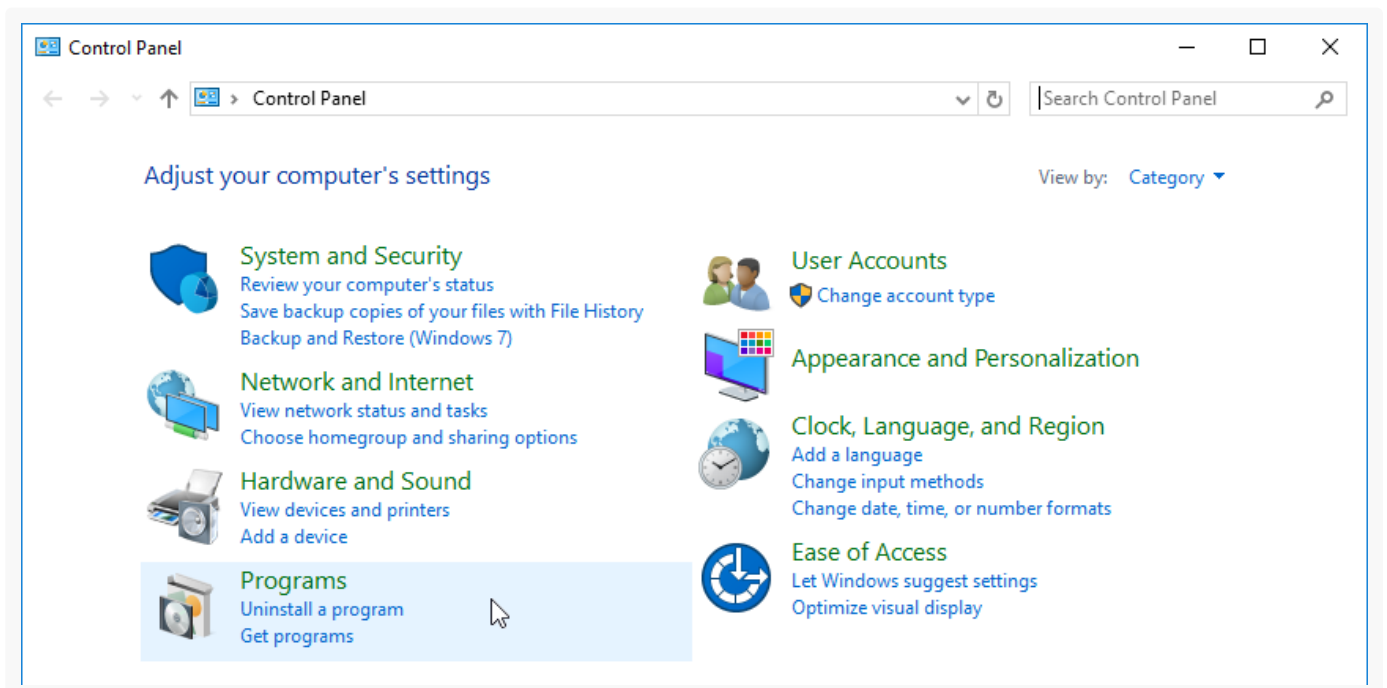
1. Enter the “control panel” in the [*Start*] menu and select the [*Control Panel*] (Fig. 1).

Fig. 1 The [*Control Panel*] section in the [*Start*] menu



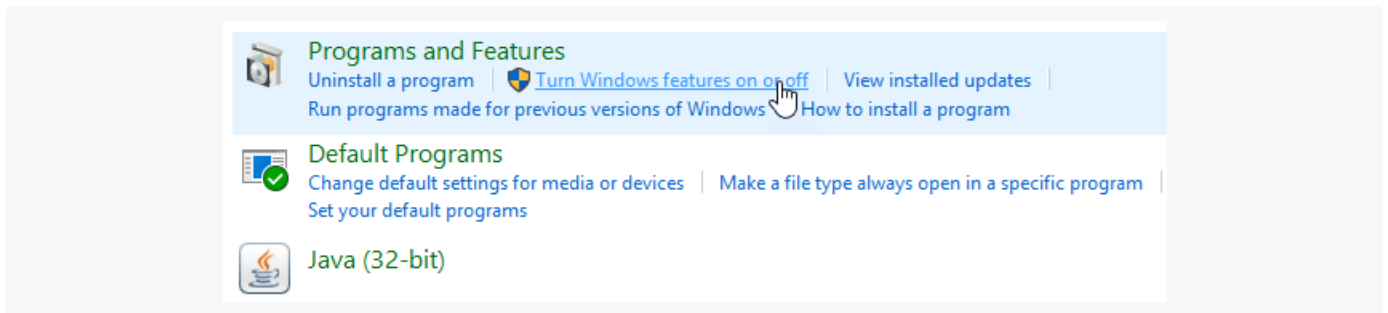
2. Select the [*Programs*] option in the opened window (Fig. 2).

Fig. 2 The [*Programs*] menu



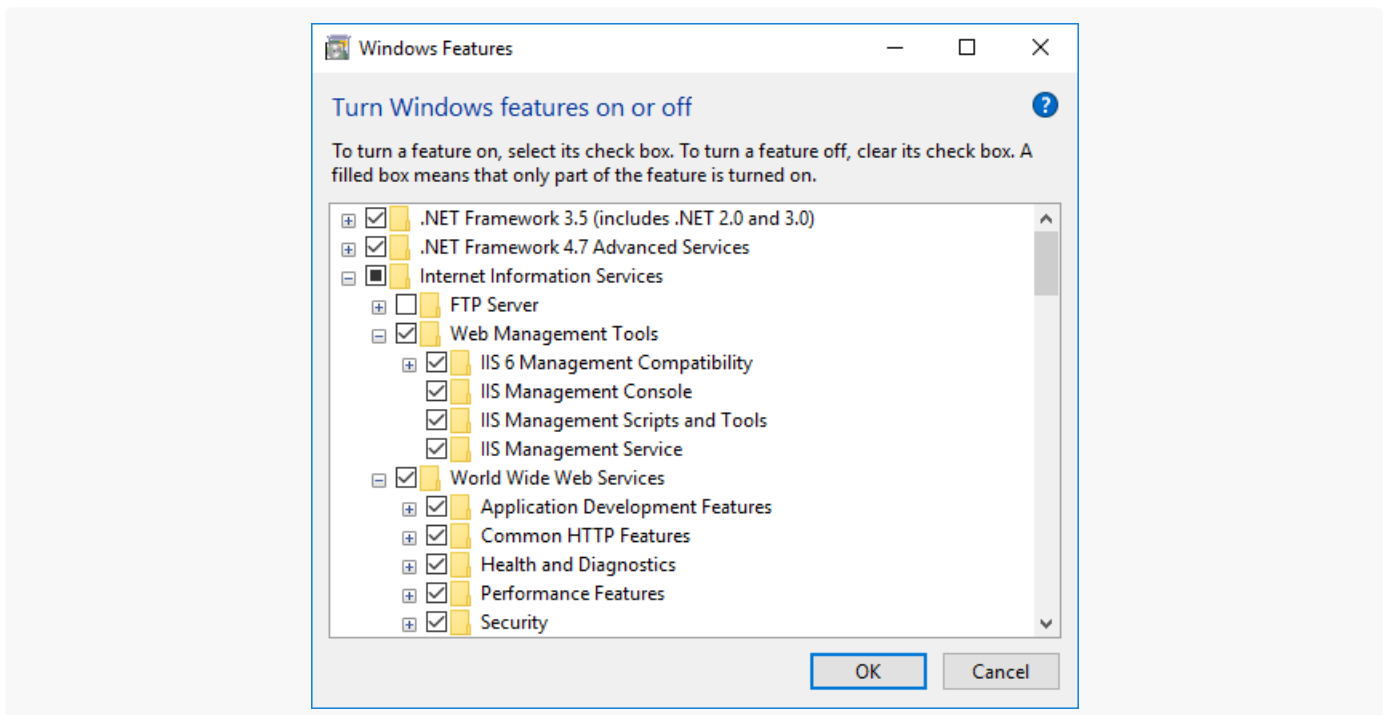
- From the [*Programs and Features*] menu, select the [*Turn Windows features on or off*] option (Fig. 2).

Fig. 3 Selecting the [*Turn Windows features on or off*] option



- Select all required components in the [*Windows Features*] window (Fig. 4).

Fig. 4 Selecting Web Server IIS and Windows components



Modify ConnectionStrings.config for MS SQL Server

PRODUCTS: **ALL CREATIO PRODUCTS**

The ConnectionStrings.config file in the Creatio root directory stores the connection parameters of the database and external services for your application.

Set up ConnectionStrings.config

- Go to the root directory of the Creatio application `~\WebAppRoot\Creatio`.

2. Open the ConnectionStrings.config file in a text editor.
3. Specify the connection parameters (**connectionStrings**) of your site.

A sample ConnectionStrings.config file.

```
<?xml version="1.0" encoding="utf-8"?>

<connectionStrings>

<add name="db" connectionString="Data Source=[ Database server name ]; Initial Catalog=[ Databas

<add name="redis" connectionString="host=[ Redis server machine name ];db=[ Redis DB number ];pc

<add name="defRepositoryUri" connectionString="" />

<add name="defWorkingCopyPath" connectionString="%TEMP%\%WORKSPACE%" />

<add name="defPackagesWorkingCopyPath" connectionString="%TEMP%\%APPLICATION%\%WORKSPACE%\Terras

<add name="clientUnitContentPath" connectionString="%TEMP%\%APPLICATION%\%WORKSPACE%\ClientUnits

<add name="sourceControlAuthPath" connectionString="%TEMP%\%APPLICATION%\%WORKSPACE%\Svn" />

<add name="elasticsearchCredentials" connectionString="User=[ ElasticSearch username ]; Password

</connectionStrings>
```

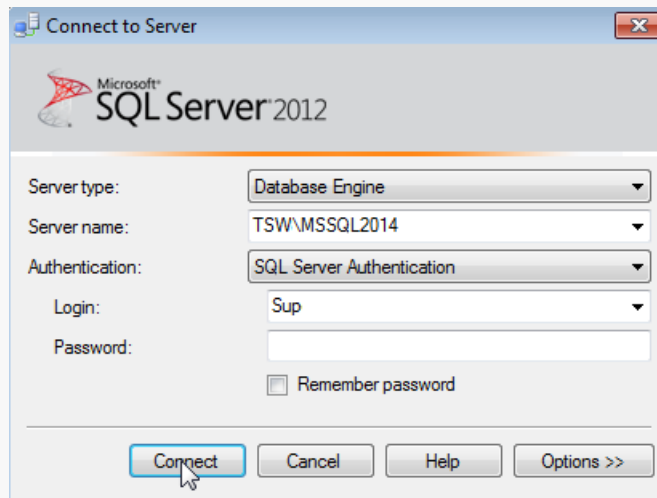
Required ConnectionStrings.config settings

Creatio requires the database and caching server connection parameters for operation.

- **name="db"** manages the connection to the restored database.

You can see the database server name (**Data Source**) in the authorization window while connecting to the server using Microsoft SQL Server Management Studio (Fig. 1).

Fig. 1 The SQL server authorization window



The database name (Initial Catalog) must match the [*Database*] field value you specified when restoring the database.

By default, Creatio uses **Windows authentication** (Integrated Security) based on the SPPI interface to connect to the database server. To ensure successful connection to the database, specify the Windows user on whose behalf you will connect to the database server.

```
<add name="db" connectionString="Data Source=[ Database server name ];
Initial Catalog=[ Database name ];
Persist Security Info=True; MultipleActiveResultSets=True;
Integrated Security=SSPI; Pooling = true; Max Pool Size = 100; Async = true" />
```

If you want to log in to the database server using the **Microsoft SQL user credentials**, create the credentials on the Microsoft SQL server and specify them in the ConnectionStrings.config file. Replace the **Integrated Security=SSPI** variable with the **User ID** and **Password** variables in the database connection string (add name="db"):

```
<add name="db" connectionString="Data Source=TSW\MSSQL2014;
Initial Catalog=7.10.2.1416_SalesEnterprise_Demo;
Persist Security Info=True; MultipleActiveResultSets=True;
User ID=Sup; Password=password; Pooling = true; Max Pool Size = 100; Async = true" />
```

- **name="redis"** manages the interaction with the Redis server.

```
<add name="redis" connectionString="host=[ Redis server machine name ];db=[ Redis DB number ]
maxReadPoolSize=10;maxWritePoolSize=500" />
```

Optional ConnectionStrings.config settings

The external service connection parameters are optional. Fill them out only if your Creatio configuration requires it. For example, do that if you want to integrate the version control system.

- **tempDirectoryPath** is the path to the temporary directory the package installation mechanism requires:

```
<add name="tempDirectoryPath" connectionString="[ Path to the temporary directory the package
```

- **defPackagesWorkingCopyPath** is the path to the working copy of Creatio custom packages. Fill out this parameter only if you use the SVN version control system. The working copy contains custom packages organized as directories and files. The built-in Creatio SVN client synchronizes the working copy with the repository of the SVN version control system. Set up this parameter when integrating the version control system. Creatio will use it only in the default development mode; i. e., if the file system development mode is disabled. The default value is a temporary directory, which the operating system may clear. We recommend specifying a custom directory. If you specify an existing Creatio directory, for example, `.\Terrasoft.WebApp\Terrasoft.Configuration\Pkg`, that may cause compilation errors.

```
<add name="defPackagesWorkingCopyPath" connectionString="[ Path to the working copy of custom
```

- **sourceControlAuthPath** is the path to the authorization data of the built-in client of the SVN version control system (if used): The default value is a temporary directory, which the operating system may clear. If you use a version control system, we recommend specifying the path to a permanent directory in this parameter.

```
<add name="sourceControlAuthPath" connectionString="[ Path to the authorization data of the ve
```

- **Influx** manages the interaction with the site analytics collection service. Fill out this parameter only if you need to collect the functionality use analytics for debugging.

```
<add name="influx" connectionString="url=[ Site analytics collection service address ]; user=
```

- **clientPerformanceLoggerServiceUri** manages the interaction with the logging service. Fill out this parameter only if you need to collect the data about how Creatio pages load.

```
<add name="clientPerformanceLoggerServiceUri" connectionString="[ Logging service address ]"
```

- **messageBroker** manages the interaction with the RabbitMQ service. Fill out this parameter only if you need to set up horizontal load scaling using RabbitMQ.

```
<add name="messageBroker" connectionString="amqp://[ MessageBroker username ]:[ Password ]@[
```

Modify ConnectionStrings.config for Oracle Database

PRODUCTS: [ALL CREATIO PRODUCTS](#)

The ConnectionStrings.config file in the Creatio root directory stores the connection parameters of the database and external services for your application.

Set up ConnectionStrings.config

1. Go to the root directory of the Creatio application `~\WebAppRoot\Creatio`.
2. Open the ConnectionStrings.config file in a text editor.
3. Specify the connection parameters (**connectionStrings**).

A sample ConnectionStrings.config file.

```
<?xml version="1.0" encoding="utf-8"?>
<connectionStrings>
  <add name="db" connectionString="Data Source=(DESCRIPTION =
  (ADDRESS_LIST = (ADDRESS = (PROTOCOL = TCP)(HOST = [ Database server name ])(PORT = 1521))) (CC
  <add name="redis" connectionString="host=[ Machine name ];db=[ Redis DB number ];port=6379;
maxReadPoolSize=10;maxWritePoolSize=500" />
  <add name="defRepositoryUri" connectionString="" />
  <add name="defWorkingCopyPath" connectionString="%TEMP%\%WORKSPACE%" />
  <add name="defPackagesWorkingCopyPath"
connectionString="%TEMP%\%APPLICATION%\%WORKSPACE%\TerrasoftPackages" />
  <add name="clientUnitContentPath"
connectionString="%TEMP%\%APPLICATION%\%WORKSPACE%\ClientUnitSrc" />
  <add name="sourceControlAuthPath"
connectionString="%TEMP%\%APPLICATION%\%WORKSPACE%\Svn" />
</connectionStrings>
```

Required ConnectionStrings.config settings

Creatio requires the database and caching server connection parameters for operation.

- **name="db"** manages the connection to the restored database, where:
 - [*Database server name*] is the network address of the database server.
 - [*Oracle service name*] is the service name.
 - [*Schema name*] is the schema name of the restored database.
 - [*Schema password*] is the schema password of the restored database.

```
<add name="db" connectionString="Data Source=(DESCRIPTION =
  (ADDRESS_LIST = (ADDRESS = (PROTOCOL = TCP)(HOST = [ Database server name ])(PORT = 1521)))
```

- **name="redis"** manages the interaction with the Redis server.

```
<add name="redis" connectionString="host=[ Machine name ];db=[ Redis DB number ];port=6379;
```

```
maxReadPoolSize=10;maxWritePoolSize=500" />
```

Optional ConnectionStrings.config settings

The external service connection parameters are optional. Fill them out only if your Creatio configuration requires it. For example, do that if you want to integrate the version control system.

- **tempDirectoryPath** is the path to the temporary directory the package installation mechanism requires:

```
<add name="tempDirectoryPath" connectionString="[ Path to the temporary directory the package
```

- **defPackagesWorkingCopyPath** is the path to the working copy of Creatio custom packages. Fill out this parameter only if you use the SVN version control system. The working copy contains custom packages arranged as directories and files. The built-in Creatio SVN client synchronizes the working copy with the repository of the SVN version control system. Set up this parameter when integrating the version control system. Creatio will use it only in the default development mode; i. e., if the file system development mode is disabled. The default value is a temporary directory, which the operating system may clear. We recommend specifying a custom directory. If you specify an existing Creatio directory, for example, `.\Terrasoft.WebApp\Terrasoft.Configuration\Pkg`, that may cause compilation errors.

```
<add name="defPackagesWorkingCopyPath" connectionString="[ Path to the working copy of custom
```

- **sourceControlAuthPath** is the path to the authorization data of the built-in client of the SVN version control system (if used): The default value is a temporary directory, which the operating system may clear. If you use a version control system, we recommend specifying the path to a permanent directory in this parameter.

```
<add name="sourceControlAuthPath" connectionString="[ Path to the authorization data of the ve
```

Modify ConnectionStrings.config for PostgreSQL

PRODUCTS: [ALL CREATIO PRODUCTS](#)

The ConnectionStrings.config file in the Creatio root directory stores the connection parameters of the database and external services for your application.

Set up ConnectionStrings.config

1. Go to the root directory of the Creatio application `~\WebAppRoot\Creatio`.
2. Open the ConnectionStrings.config file in a text editor.
3. Specify the connection parameters (**connectionStrings**).

A sample ConnectionStrings.config file.

```
<?xml version="1.0" encoding="utf-8"?>
<connectionStrings>
<add name="db" connectionString="Server=[ Database server name ];Port=[ Database server port ];C
<add name="redis" connectionString="host=[ Machine name ];db=[ Redis DB number ];port=6379;maxRe
<add name="redisSentinel" connectionString="sentinelHosts=localhost:26380,localhost:26381,localh
<add name="defPackagesWorkingCopyPath" connectionString="%TEMP%\%APPLICATION%\%APPPPOOLIDENTITY%\
<add name="tempDirectoryPath" connectionString="%TEMP%\%APPLICATION%\%APPPPOOLIDENTITY%\%WORKSPAC
<add name="sourceControlAuthPath" connectionString="%TEMP%\%APPLICATION%\%APPPPOOLIDENTITY%\%WORK
<add name="elasticsearchCredentials" connectionString="User=[ ElasticSearch username ]; Password
<add name="influx" connectionString="url=http://10.0.7.161:30359; user=; password=; batchInterve
</connectionStrings>
```

Required ConnectionStrings.config settings

Creatio requires the database and caching server connection parameters for operation.

1. For the restored database (**name="db"**).

```
<add name="db" connectionString="Server=[ Database server name ];Port=[ Database server port
```

2. For Redis Server (**name="redis"**):

```
<add name="redis" connectionString="host=[ Machine name ];db=[ Redis DB number ];port=6379;ma
```

Optional ConnectionStrings.config settings

The external service connection parameters are optional. Fill them out only if your Creatio configuration requires it. For example, do that if you want to integrate the version control system.

- **tempDirectoryPath** is the path to the temporary directory the package installation mechanism requires:

```
<add name="tempDirectoryPath" connectionString=[ Path to the temporary directory the package
```

- **defPackagesWorkingCopyPath** is the path to the working copy of Creatio custom packages. Fill out this parameter only if you use the SVN version control system. The working copy contains custom packages organized as directories and files. The built-in Creatio SVN client synchronizes the working copy with the repository of the SVN version control system. Set up this parameter when integrating the version control system. Creatio will use it only in the default development mode; i. e., if the file system development mode is disabled. The default value is a temporary directory, which the operating system may clear. We recommend specifying a custom directory. If you specify an existing Creatio directory, for example, `.\Terrasoft.WebApp\Terrasoft.Configuration\Pkg`, that may cause compilation errors.

```
<add name="defPackagesWorkingCopyPath" connectionString="[ Path to the working copy of custom
```

- **sourceControlAuthPath** is the path to the authorization data of the built-in client of the SVN version control system (if used): The default value is a temporary directory, which the operating system may clear. If you use a version control system, we recommend specifying the path to a permanent directory in this parameter.

```
<add name="sourceControlAuthPath" connectionString="[ Path to the authorization data of the ve
```

- **Influx** manages the interaction with the site analytics collection service. Fill out this parameter only if you need to collect the functionality use analytics for debugging.

```
<add name="influx" connectionString="url=[ Site analytics collection service address ]; user=
```

- **clientPerformanceLoggerServiceUri** manages the interaction with the logging service. Fill out this parameter only if you need to collect the data about how Creatio pages load.

```
<add name="clientPerformanceLoggerServiceUri" connectionString="[ Logging service address ]"
```

- **messageBroker** manages the interaction with the RabbitMQ service. Fill out this parameter only if you need to set up the horizontal load scaling using RabbitMQ.

```
<add name="messageBroker" connectionString="amqp://[ MessageBroker username ]:[ Password ]@[
```

Modify Web.config

PRODUCTS: [ALL CREATIO PRODUCTS](#)

Generate a unique `machineKey` value for your Creatio application. To do this:

1. Download the PowerShell script. [Download the script](#).
2. Run the PowerShell console as an administrator.
3. Specify the path to the root Creatio directory in the PowerShell terminal and run the script. Example of the command to run:

```
.\UpdateMachineKey.ps1 "[Path to the root Creatio directory]"
```

As a result, the script will generate a unique `machineKey` value in Web.config files located in the root Creatio directory and the Terrasoft.WebApp directory.

Additional Web.config setup for Oracle

After you set up the database connection parameters in the ConnectionStrings.config file for the Oracle database, set up the Web.config configuration file that includes certain configuration parameters required for Creatio to operate as intended.

To set up Web.config:

1. Ensure the Creatio website can access the configuration parameters described in the ConnectionStrings.config configuration file. To do this, open the Web.config configuration file in the root Creatio directory and set the `currentSchemaName` attribute of the `general` parameter in the `db` section to the name of the schema specified in the ConnectionStrings.config configuration file.

```
<configuration>
  ...
  <terasoft>
    <db>
      <general ... currentSchemaName="SOME_SCHEMA_NAME_OF_ORACLE_DATABASE" />
    </db>
  </terasoft>
  ...
</configuration>
```

where `currentSchemaName` is the name of the Oracle database schema.

2. Save the changes.

Set up Creatio to work with Oracle 19c

You can set up Creatio to work with Oracle 19c in Creatio version 8.0 Atlas and later. Since version 8.0.2 Atlas, Creatio works with Oracle 19c out of the box.

We recommend using Oracle 19c as Creatio DBMS. Familiarize yourself with Oracle 19c capabilities in the [official vendor documentation](#). You can use Oracle 19c thanks to the Managed ODAC 12 library.

Note. We recommend performing the setup on a pre-production environment first. Learn more in developer documentation: [Environments](#).

In general, the procedure to set up Creatio to work with Oracle 19c consists of the following steps:

1. Enable the Managed ODAC 12 library. [Read more >>>](#)
2. Update the Oracle DBMS to version 19c. [Read more >>>](#)

Enable the Managed ODAC 12 library

Make sure that you are using Creatio version 8.0 Atlas or later. If you are using an earlier version, update Creatio. To do this, follow the [update guide](#).

To **enable the Managed ODAC 12 library**:

1. Check the Oracle 19c database connector the website uses.

```
<configuration>
  ...
  <system.data>
    <DbProviderFactories>
      <add name="Oracle.ManagedDataAccess, Managed Driver" invariant="Oracle.ManagedDat
    </DbProviderFactories>
  </system.data>
  ...
</configuration>
```

2. Back up the Web.config configuration file in the root Creatio directory.
3. Set up the interaction between the Creatio website and database via the Managed ODAC 12 library. To do this, set the `executorType` parameter in the Web.config configuration file to

```
Terrasoft.DB.Oracle.OracleManagedExecutor, Terrasoft.DB.Oracle :
```

```
<configuration>
  ...
  <terrasoft>
    <db>
      <general ... executorType="Terrasoft.DB.Oracle.OracleManagedExecutor, Terrasoft.D
    </db>
  </terrasoft>
</configuration>
```

where `executorType` is the library that manages the Creatio database.

4. Set up the interaction between the scheduler and database tables via the Managed ODAC 12 library. To do this, set the `quartz.dataSource.SchedulerDb.provider` key in the Web.config configuration file to

```
OracleManagedProvider :
```

```
<configuration>
  ...
  <quartzConfig defaultScheduler="BPOnlineQuartzScheduler">
    <quartz isActive="true">
      ...
      <add key="quartz.dataSource.SchedulerDb.provider" value="OracleManagedProvider" /
    </quartz>
  </quartzConfig>
</configuration>
```

where `quartz.dataSource.SchedulerDb.provider` is the key that enables the scheduler to interact with database tables.

5. Run Creatio and check the functionality in operation. If you notice issues, restore the Web.config configuration file from backup and repeat the setup.
6. Repeat the setup in the production environment or transfer changes to the production environment. Learn more in developer documentation: [Environments](#).

Update the Oracle DBMS to version 19c

To update the Creatio Oracle DBMS to version 19c, follow the [official vendor documentation](#).

As a result, your Creatio application will use Oracle 19c DBMS.

Set up Creatio application server on IIS

PRODUCTS: [ALL CREATIO PRODUCTS](#)

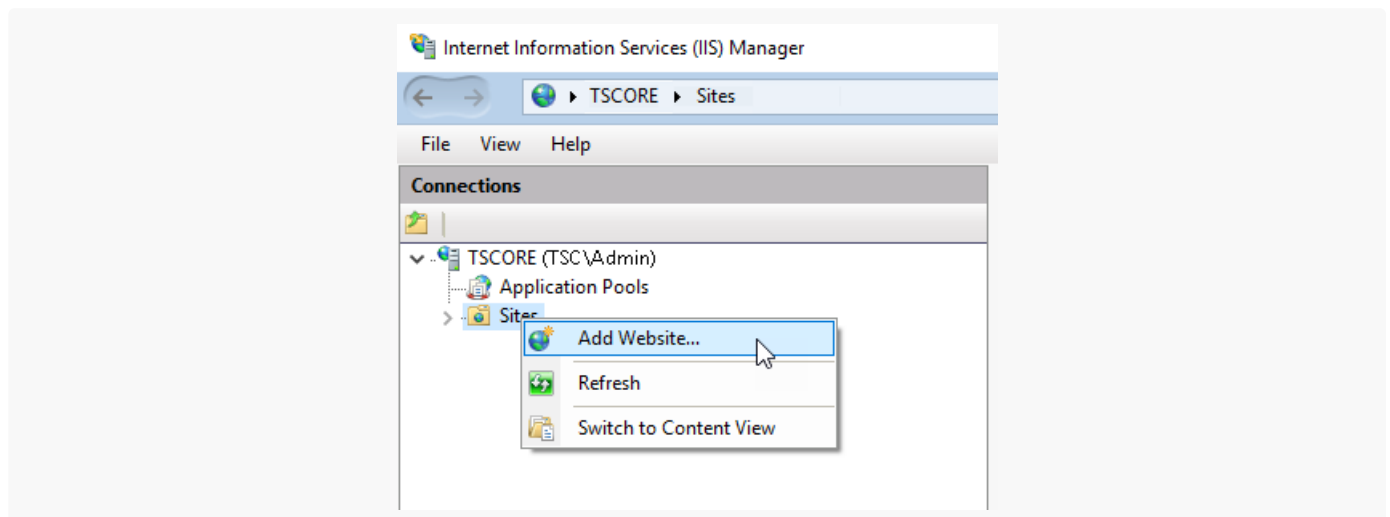
Setting up Creatio application server (web server) on IIS involves setting up application website in IIS and adding an application pool.

Set up application website in IIS

To create and set up a website:

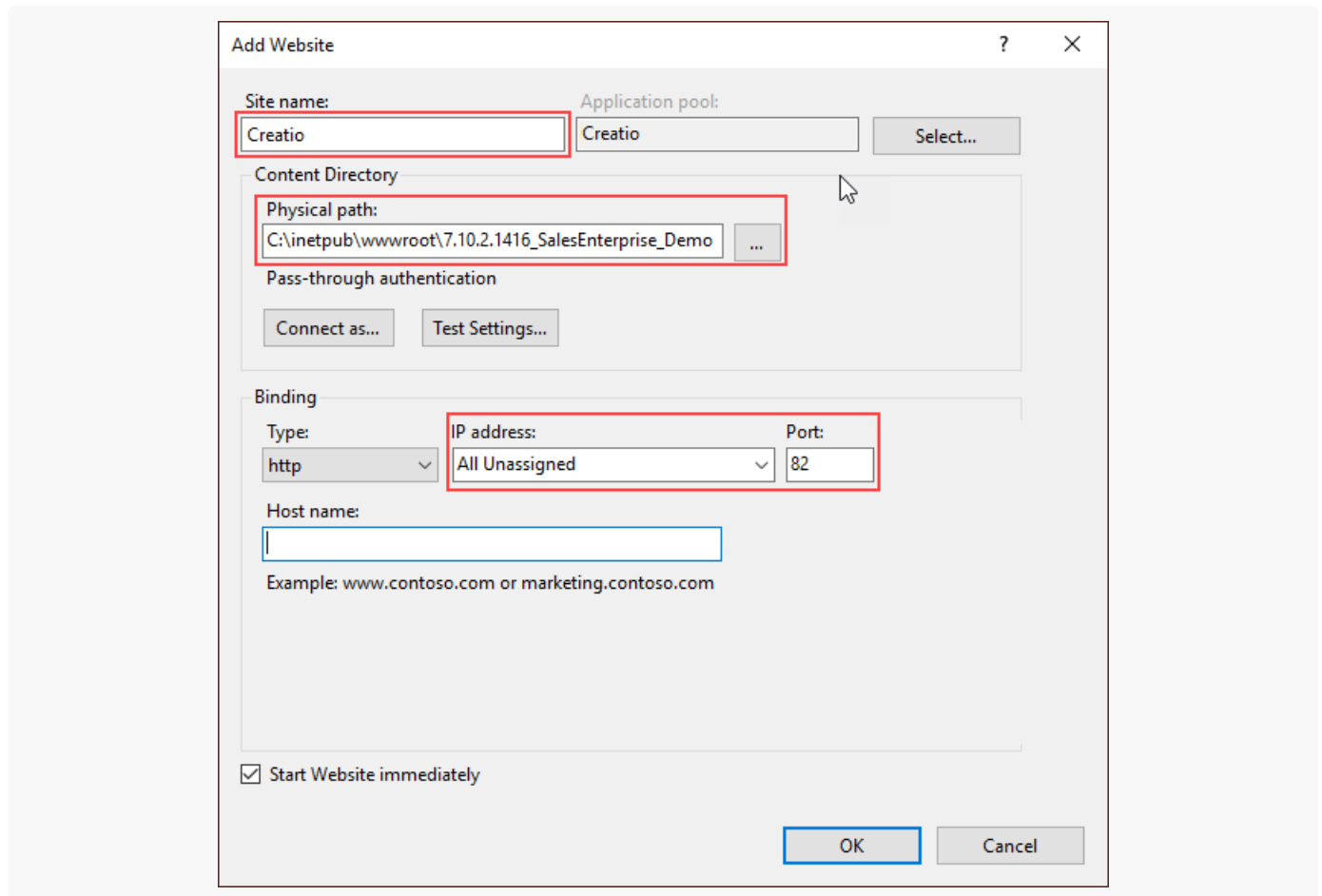
1. Go to the IIS control window, right-click the [*Sites*] folder, and select the [*Add Website*] option from the context menu (Fig. 1).

Fig. 1 The [*Add website*] option

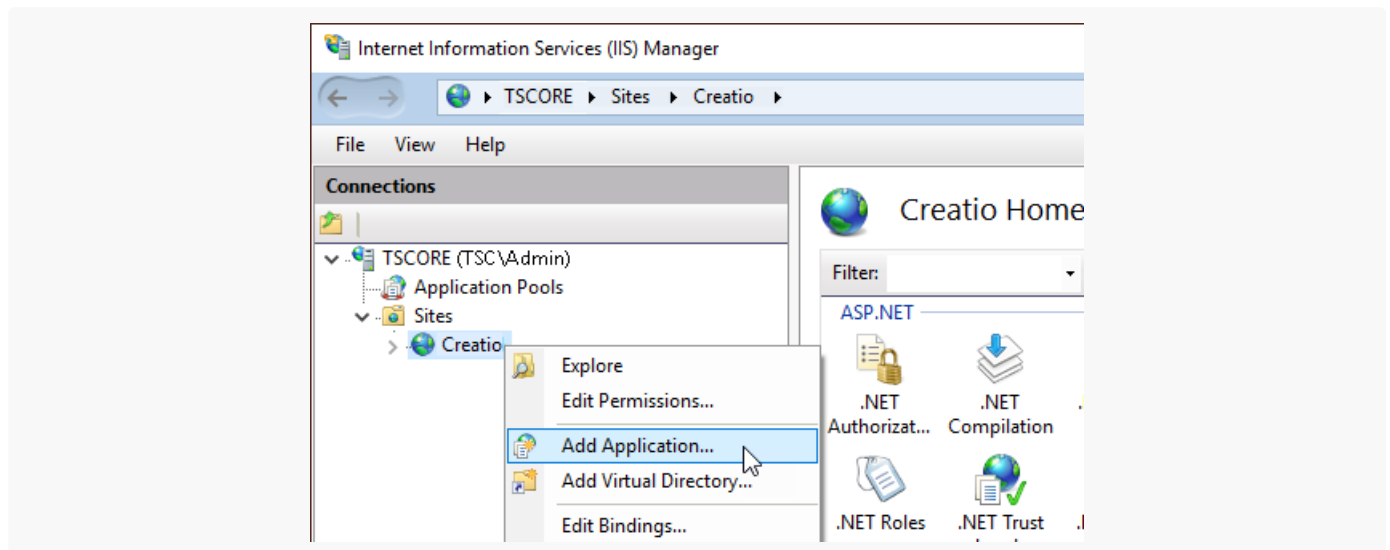


2. Specify the name of the website, the path to the root folder with Creatio files, IP address and website port (Fig. 2). The default website path is `C:\inetpub\wwwroot`. If needed, specify your own IP address.

Fig. 2 The new website parameters window

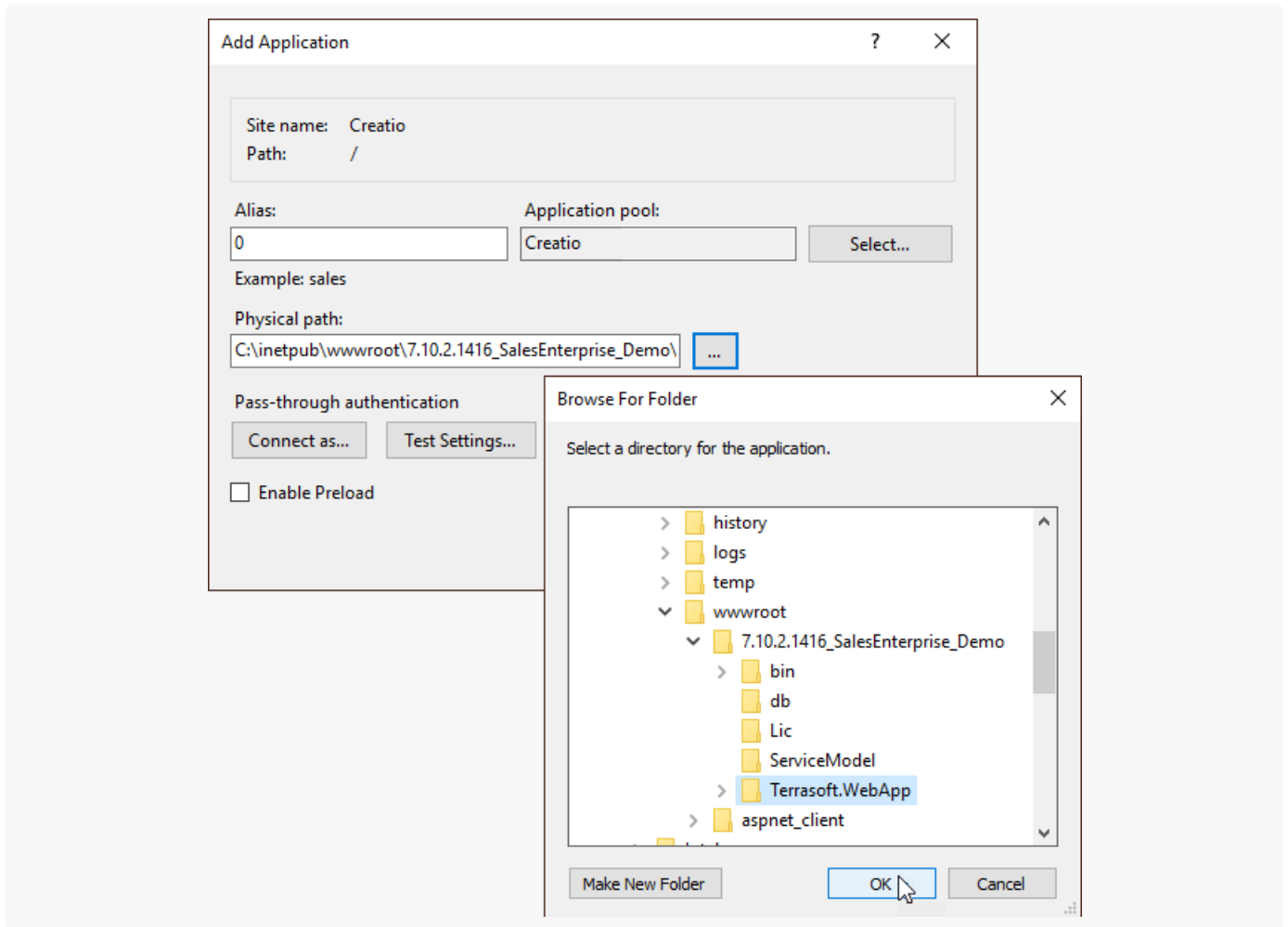


3. Right-click the created website in the [*Connections*] area and select the [*Add Application*] option (Fig. 3).

Fig. 3 The [*Add Application*] option

4. Enter the "0" application name in the [*Alias*] field. Specify the "Terrasoft.WEBApp" directory (Fig. 4).

Fig. 4 The application parameters selection window

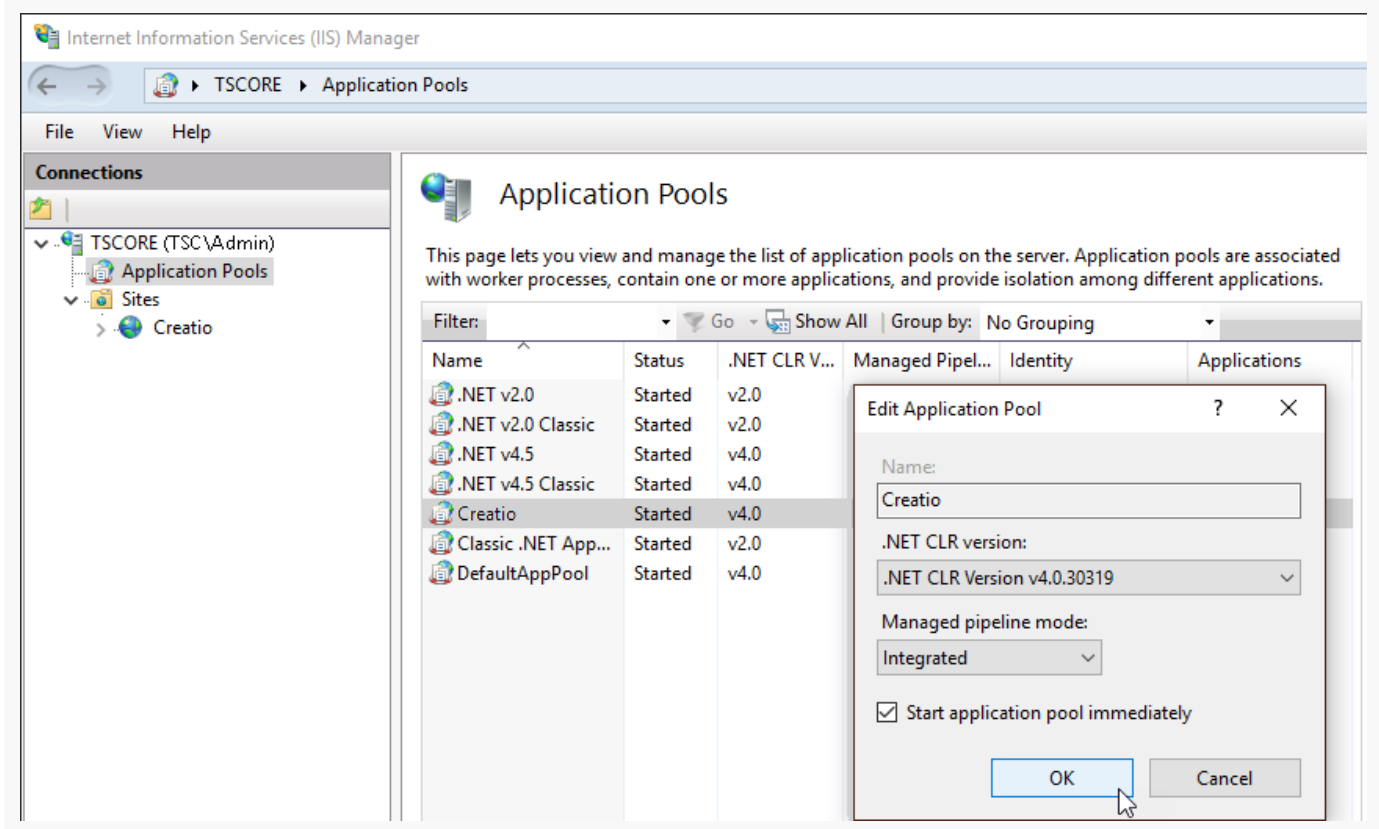


Add an application pool

To add an application pool:

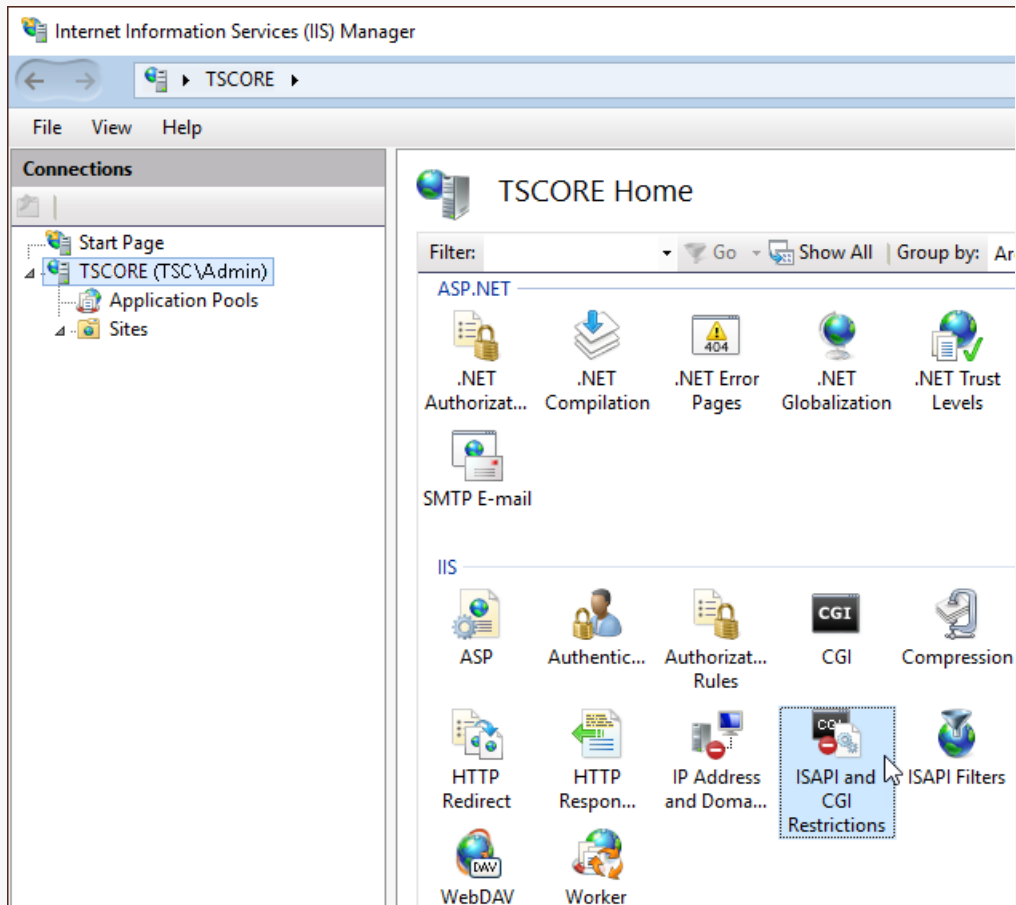
1. Go to the [*Application Pools*] section in the [*Connections*] area of the IIS control window.
2. Select the [*Creatio*] pool.
3. Select the [*Integrated*] mode in the [*Managed pipeline mode*] field.
4. Specify Asp.Net version 4.0.30319 in the [*.Net Framework version*] field (Fig. 5).

Fig. 5 The input window for Applications Pools parameters



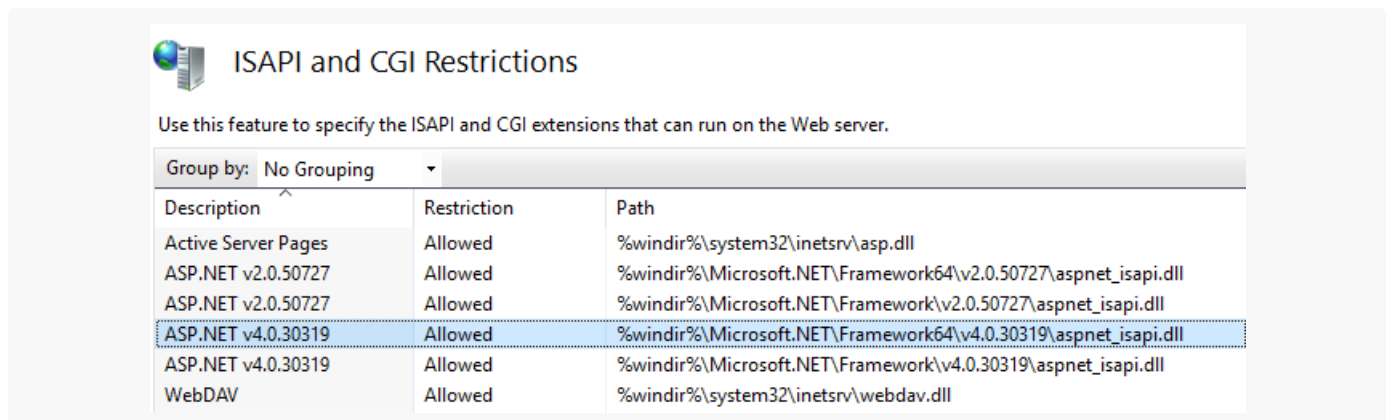
5. Go to the ISAPI and CGI Restrictions on the web server level (Fig. 6) and check if the specified ASP.Net version is allowed.

Fig. 6 The ISAPI and CGI Restrictions menu



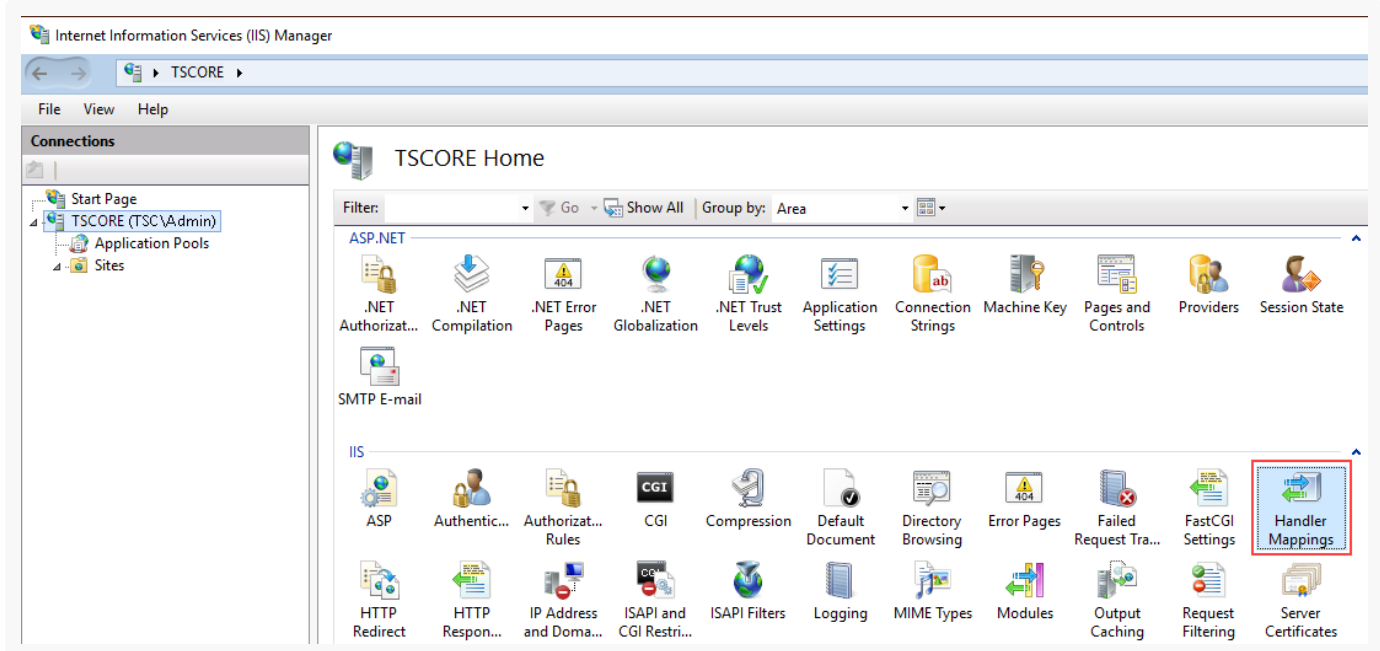
1. Make sure that the [*Allowed*] status is set in the [*Restriction*] field for the ASP.Net version (Fig. 7).

Fig. 7 The status of the ASP.Net version



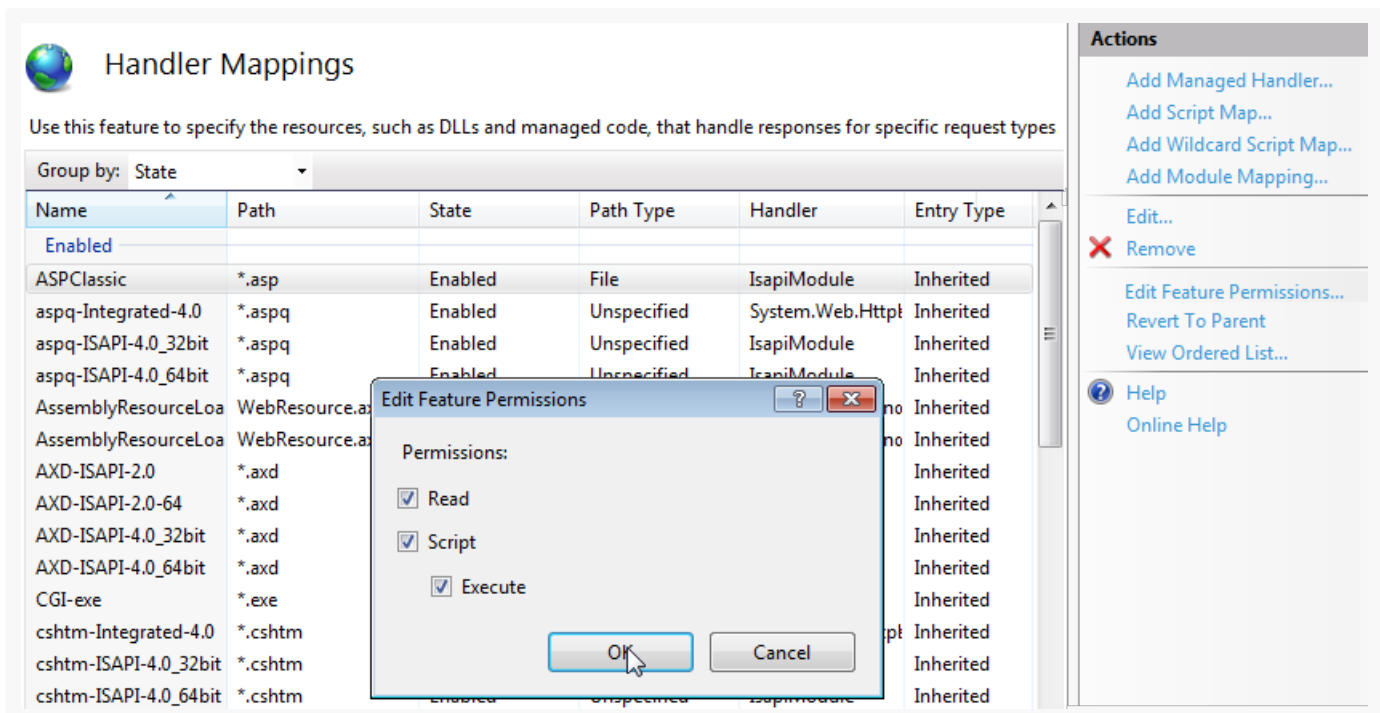
2. Open the Handler Mappings on the server level and make sure that all the required permissions are active (Fig. 8).

Fig. 8 The Handler Mappings menu



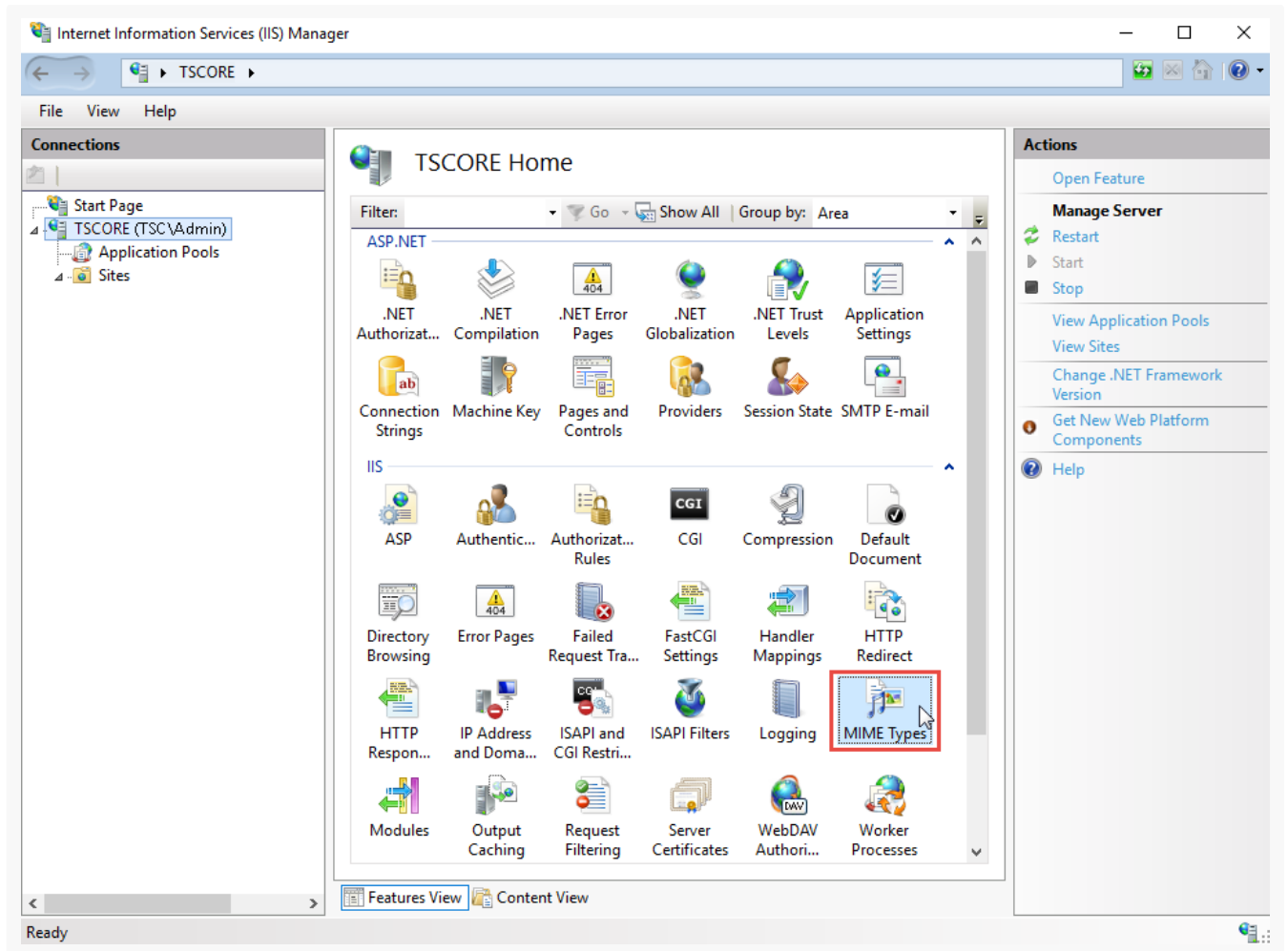
1. Click [*Edit Feature Permissions*] in the [*Actions*] area.
2. Make sure that all the required checkboxes are selected in the [*Edit Feature Permissions*] window (Fig. 9).

Fig. 9 The [*Edit Feature Permissions*] window



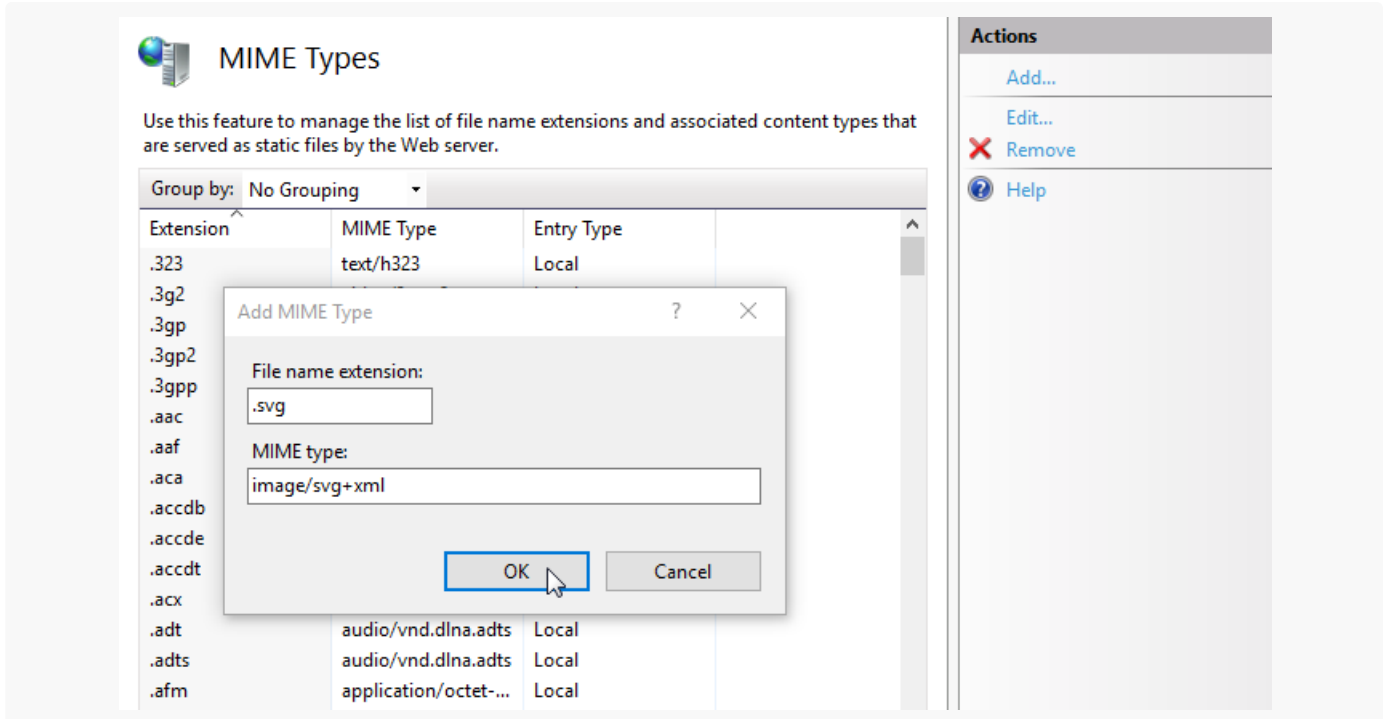
3. Make sure that MIME-type for .svg and .json files is configured in the new application. This configuration can be performed both on the server (in this case, all applications on this server inherit it) and application level. To check the configuration:
 - a. Go to MIME Types on the server or application level (Fig. 10).

Fig. 10 The MIME Types menu



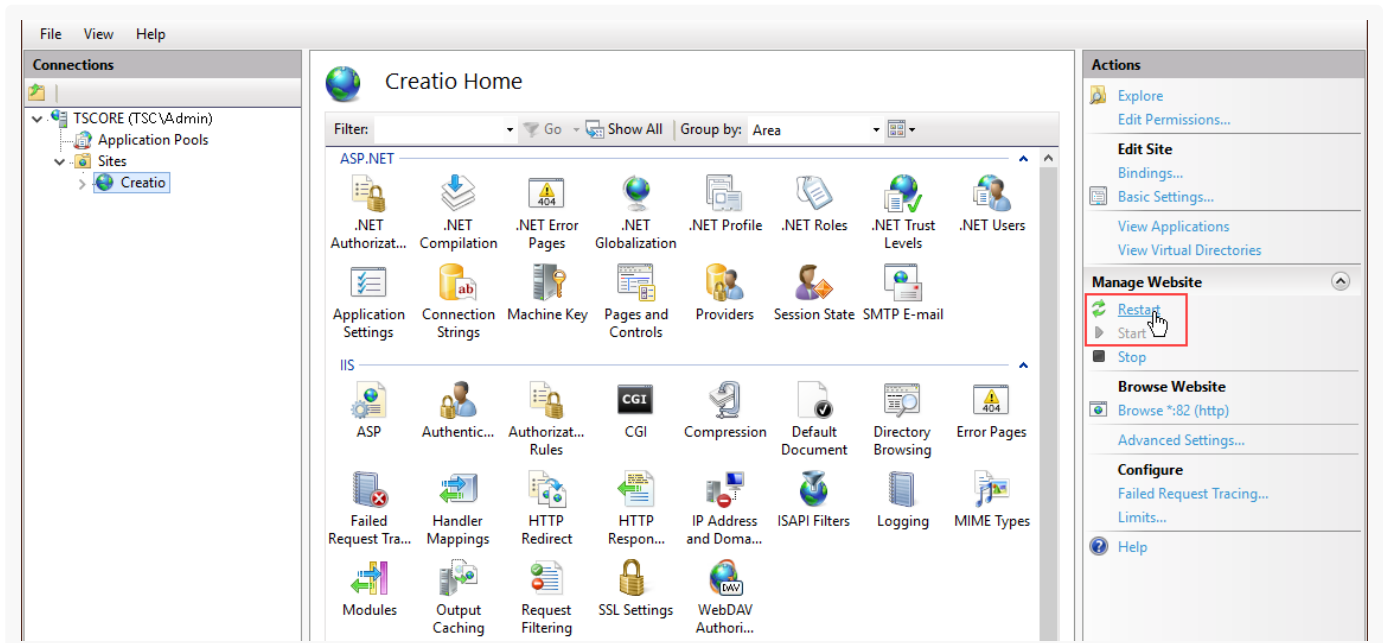
- b. Make sure that configuration for .svg and .json files is available. If the configuration is available, go to step 12.
4. If the configuration is not available, click [Add...] in the [Actions] area. This will open a new window. Specify .svg and MIME type of the data that corresponds to this extension (Fig. 11) in the window that opens. Repeat the step for .json extension (“application/json” MIME type).

Fig. 11 The MIME data type for .svg files



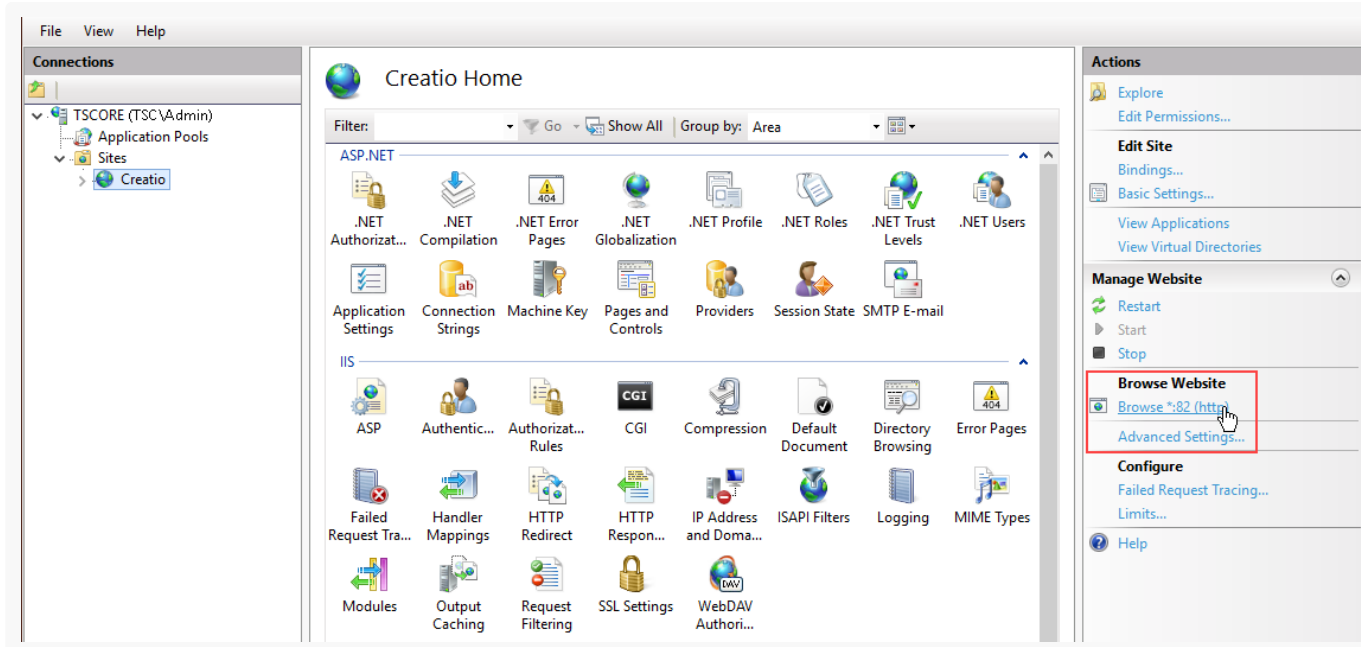
- Restart the website using the [*Restart*] command in the [*Manage Website*] area (Fig. 12).

Fig. 12 The [*Restart*] command in the [*Manage Websites*] area



- Open the site by going to the address or using the [*Browse*] command (Fig. 13). Make sure that the Creatio login page is displayed.

Fig. 13 The [*Browse*] command in the [*Actions*] area



Note. To log in to a newly deployed application, use the default Supervisor user account. Login: Supervisor; Password: Supervisor. We highly recommended changing the Supervisor password immediately.

7. To enable additional UI language:

- a. Go to the [*Languages*] section in the system designer.
- b. Select the needed language and click [*Open*].
- c. Select the [*Active*] and [*Use by default*] checkboxes on the opened page. Save the changes.

To enable a language, the user who has run the IIS application pool needs to have access permissions to read, edit and delete application files and content subordinate catalogs (catalog `.\Terrasoft.WebApp\conf`).

8. Click [*System settings*] in the System Designer and change the [*Order of first/last names*] system setting value to "Last name, First name [*Middle name*]." This is required to correctly display contact names in individual columns: [*Last name*], [*First name*], [*Middle name*].

Set up WebSockets

PRODUCTS: [ALL CREATIO PRODUCTS](#)

Creatio uses the WebSocket protocol to run custom processes, manage notifications, and integrate with telephony. To ensure all system functions operate correctly, enable WebSockets and configure them on the Creatio application server.

The application server must have **Windows Server 2012**, **Windows Server 2016** or **Windows 8/10** deployed and Internet Information Services (IIS) version 8 or later installed. Creatio configuration files are set up out-of-the-box. You only need to perform the setup on the server side. However, if you deploy and set up Creatio for the first time, we recommend that you check the configuration file settings and make sure that the WebSockets operate correctly. Learn more: [Check WebSocket settings for Windows Server 2012 or Windows](#)

[server 2016.](#)

To use the encrypted HTTPS connection, perform additional setup. Learn more in a separate article: [Switch a Creatio website from HTTP to HTTPS.](#)

Attention. If you use a proxy server in your local network, set it up to proxy the WebSocket protocol. The setup instructions are normally available in the proxy server documentation.

This article covers the WebSocket setup procedure on the application side **in Creatio configuration files.**

Note. Learn more about installing components that enable WebSocket protocol into the server in a separate article: [Enable required Windows components.](#)

Check WebSocket settings for Windows Server 2012 or Windows Server 2016

To check WebSocket settings in Creatio deployed on a server running Windows Server 2012 or Windows Server 2016:

1. Open the Web.config file in the Creatio root directory and make sure inheritance is disabled. The request length limits and execution timeout must also be specified.

```
<location path="." inheritInChildApplications="false">
  <system.web>
    ...
    <httpRuntime maxRequestLength="73400" executionTimeout="28800" targetFramework="4.7" />
```

2. Open the Web.config file in [*Path to Creatio root folder*]\Terrasoft.WebApp\ directory and make sure the default wsService type value is "**Terrasoft.Messaging.MicrosoftWSService.MicrosoftWSService, Terrasoft.Messaging.MicrosoftWSService**". The HTTP request length and execution timeouts, as well as additional module calls, must also be specified.

Attention. We recommend using "MicrosoftWSService" instead of "SuperWSService" for Microsoft Windows Server 2012.

Note. The portForClientConnection="0" value means the web application port is used.

```
<wsService type="Terrasoft.Messaging.MicrosoftWSService.MicrosoftWSService,
Terrasoft.Messaging.MicrosoftWSService" encrypted="false" portForClientConnection="0" />
...
<location path="." inheritInChildApplications="false">
```

```

<system.web>
...
<httpRuntime maxRequestLength="102400" executionTimeout="28800"
targetFramework="4.6.2" />
<httpHandlers>
...
<add verb="GET" path="*ViewModule.aspx.ashx" type="Terrasoft.Messaging.MicrosoftWSSer
Terrasoft.Messaging.MicrosoftWSService" />
...
<system.webServer>
...
<handlers>
...
<add name="WSHandler" verb="*" path="*ViewModule.aspx.ashx"
type="Terrasoft.Messaging.MicrosoftWSService.WSHandler, Terrasoft.Messaging.MicrosoftWSServic

```

Note. You can check the WebSocket connection in the browser console. If the connection is successful, the console will contain a record in the following format: WebSocket-connection opened for url:ws://demo.creatio.com/0/Nui/ViewModule.aspx.ashx

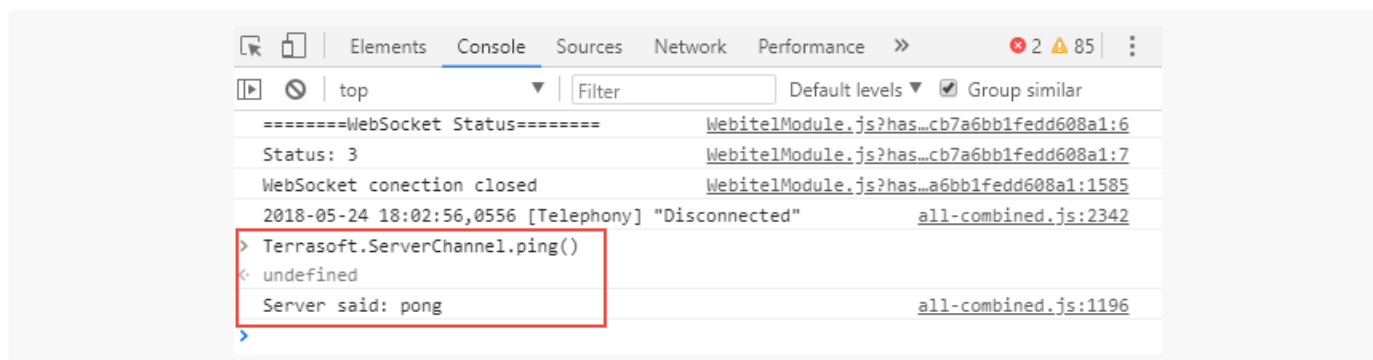
WebSocket setup FAQ

How can I make sure the WebSockets are set up correctly?

You can make sure the WebSockets are set up correctly in several ways:

- Use the [Excel data import](#) functionality. If the WebSockets are set up correctly, Creatio will import the data.
- Run the following command at the browser console: `Terrasoft.ServerChannel.ping()`. If the WebSockets are set up correctly, the server will return “pong” (Fig. 1). If the server returns any other response, review the WebSocket configuration.

Fig. 1 Test the WebSocket setup using the browser console



- Manually run a business process that contains the start timer and auto-generated page. If the WebSockets are set up correctly, the auto-generated page will open.

Note. Learn more about business process elements in a separate guide: [Process elements reference](#).

I set up WebSockets, but they will not work. Why?

If the WebSockets will not work after the setup, make sure:

- The server has all WebSocket protocol components deployed. Learn more in a separate article: [Enable required Windows components](#).
- The WebSocket usage protocol is installed into the proxy server if you use it in your local network.
- Your antivirus and firewall do not block the WebSocket operation. If you cannot disable these programs on the server, add the IP address and port of your Creatio site to the list of exceptions for inbound and outbound connections.
- Your browser extensions and add-ons, including VPN, do not block the WebSocket operation.

Switch a Creatio website from HTTP to HTTPS

PRODUCTS: [ALL CREATIO PRODUCTS](#)

The HTTPS protocol ensures secure connection between a client and a web service. Switching from HTTP to HTTPS is recommended to increase system security and enable additional services, such as WebRTC support in Webitel. Please note that this article refers only to on-site applications. To switch to HTTPS, you need to change several options of the website in IIS and edit the Web.config file. Creatio cloud uses secure connection by default.

Note. You will not be able to use the advantages of HTTPS if Creatio application is deployed on Windows Server 2008.

IIS setup

Before configuring HTTPS, do the following:

- Obtain a digital certificate from the certification center in PFX format;

Note. If you are using a self-signed certificate, Creatio mobile application will not be able to connect to the Creatio site due to the security policies of mobile applications.

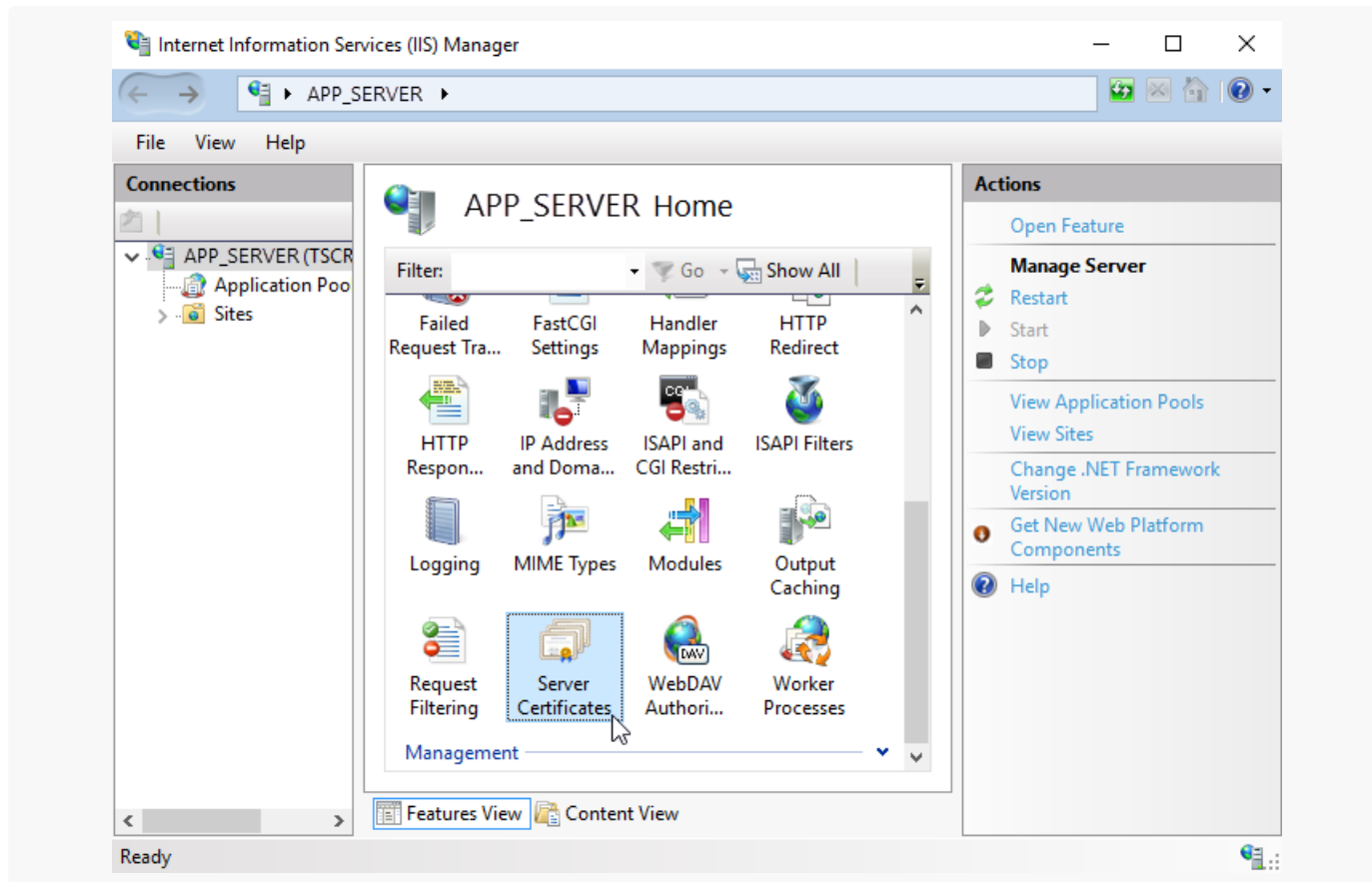
- [Set up websockets](#) for the correct operation of all system components;
- Additionally, check the list of installed [IIS components](#) to avoid errors during Creatio setup and operation.

The received digital certificate must be loaded into the server certificate storage:

1. Open Internet Information Services (IIS) Manager.

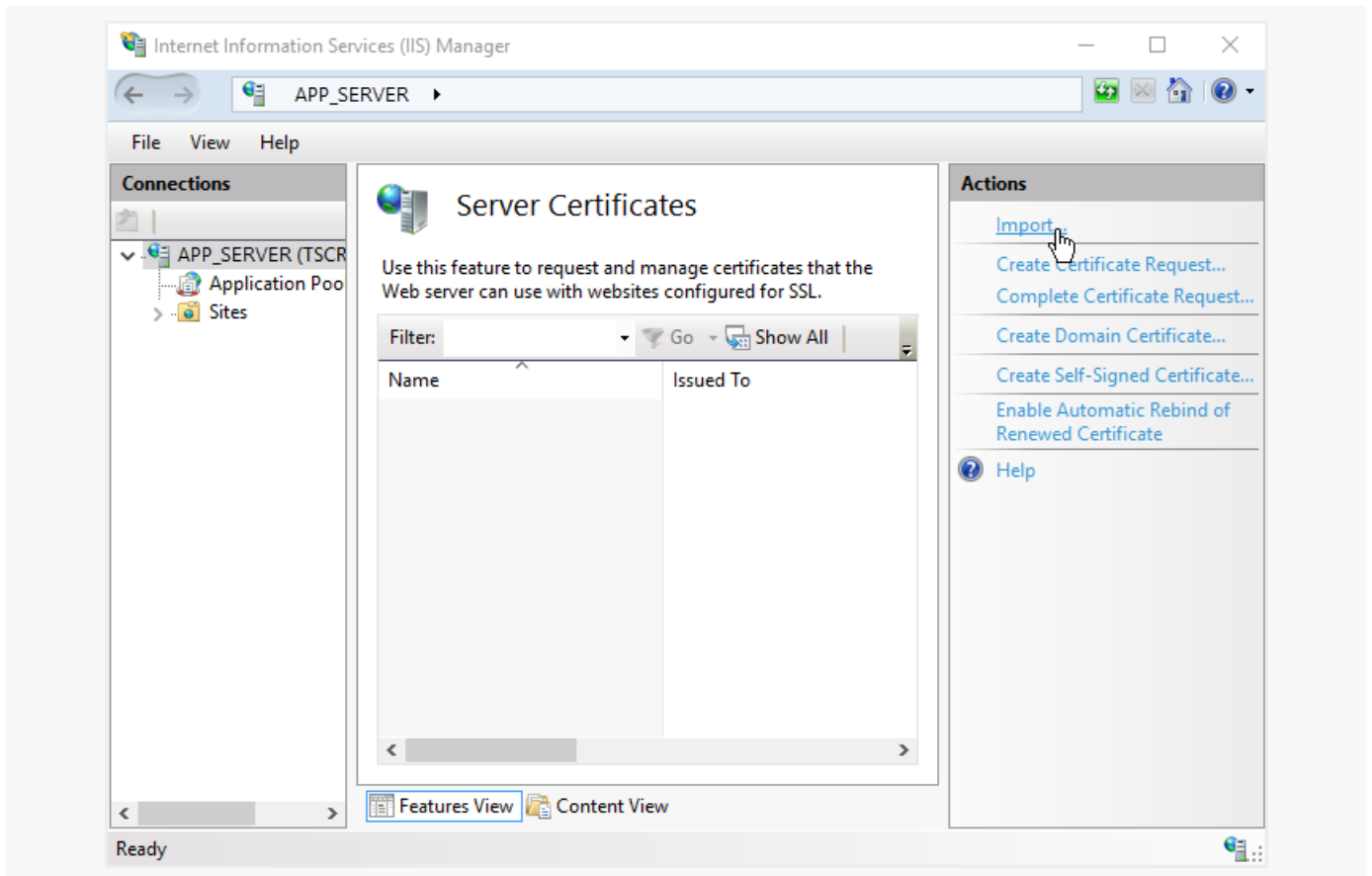
- In the main IIS window, double-click the [*Server Certificates*] detail ([Fig. 1](#)).

Fig. 1 Selecting the [*Server Certificates*] detail



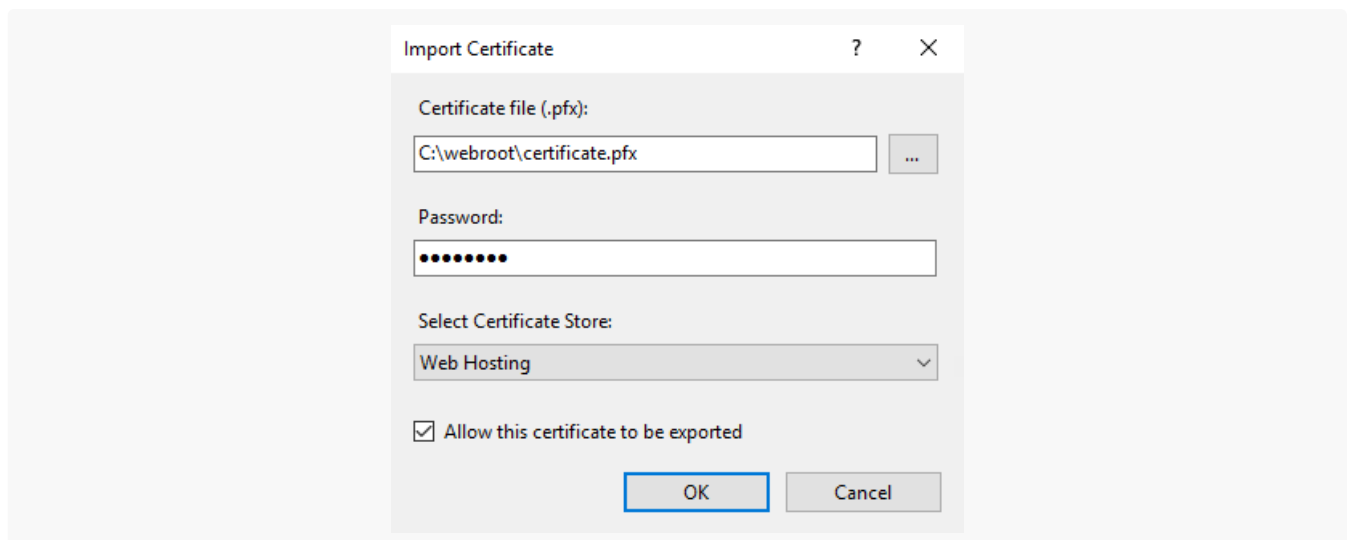
- In the [*Server Certificates*] window, click the [*import*] link in the action menu to the right ([Fig. 2](#)).

Fig. 2 Opening the [*Import*] window



4. In the import dialog box, specify:
 - a. Path to the import file hosted on the server
 - b. Password (if required)
 - c. Certificate storage ([Fig. 3](#))

Fig. 3 Certificate import window

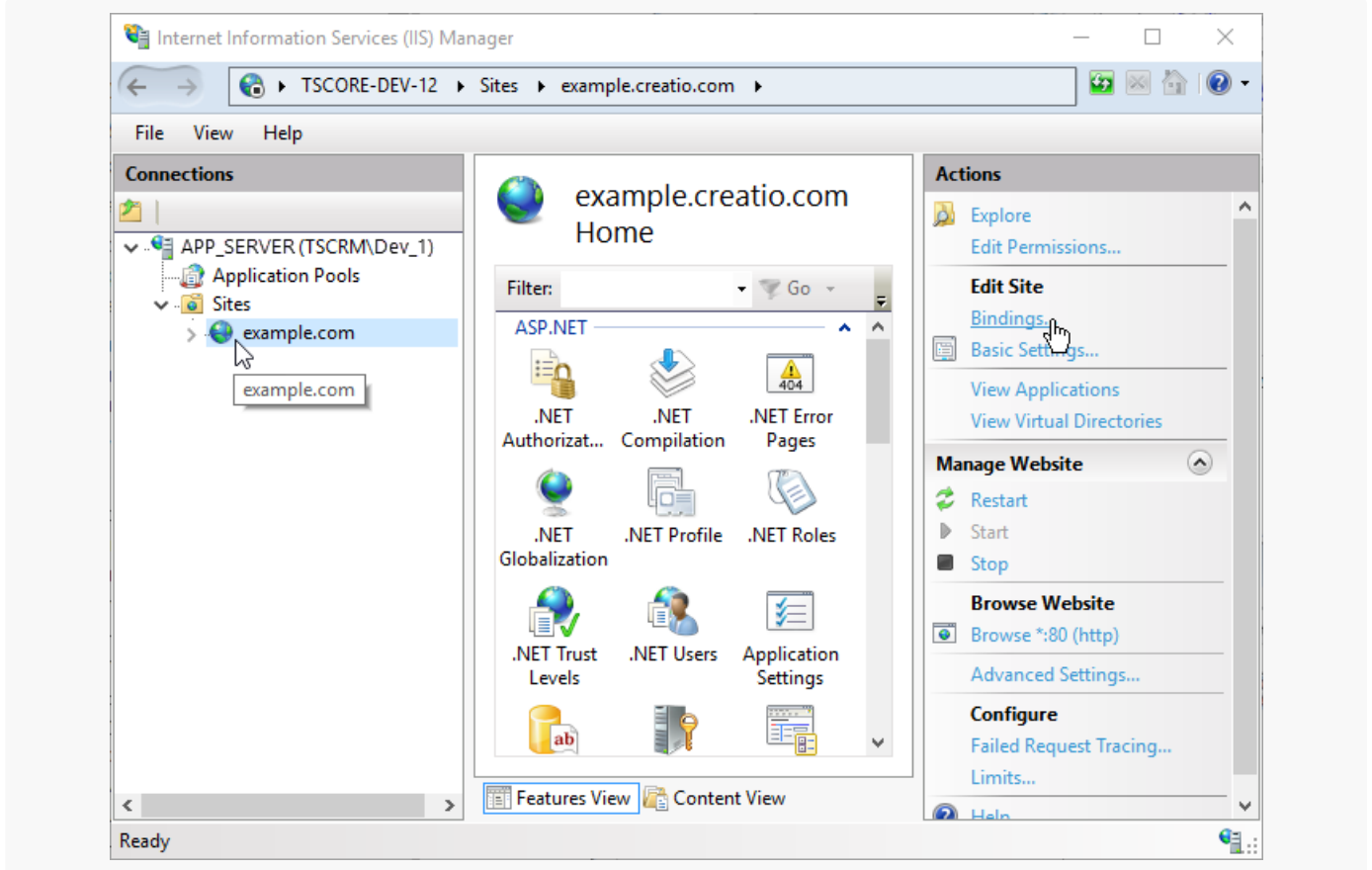


5. Click [OK] to import the certificate.

Next, connect the imported certificate to the Creatio application:

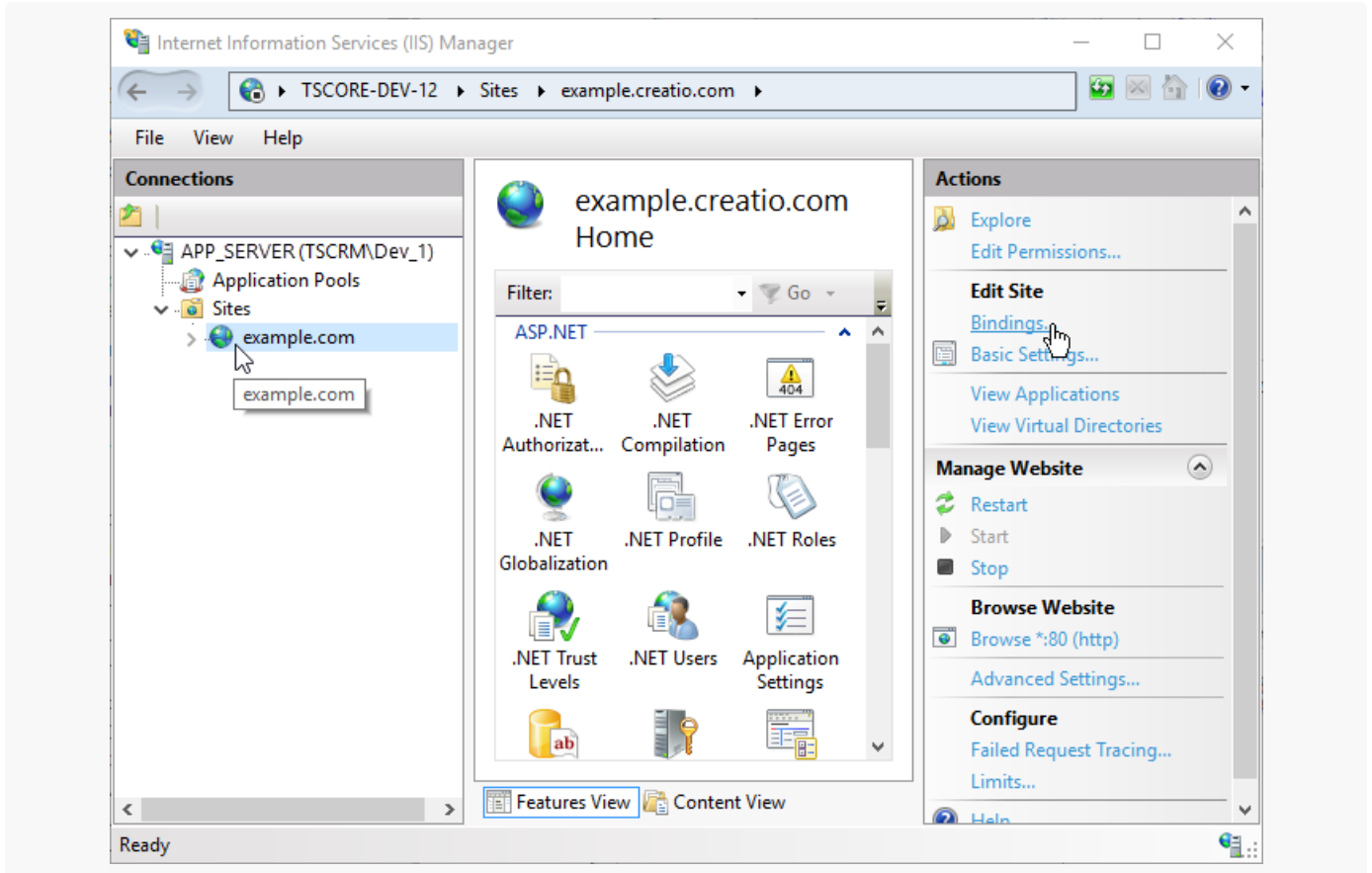
1. In the IIS window, go to the application website by clicking its name in the left [*Connections*] menu (Fig. 4).

Fig. 4 Selecting the Creatio website in the IIS window



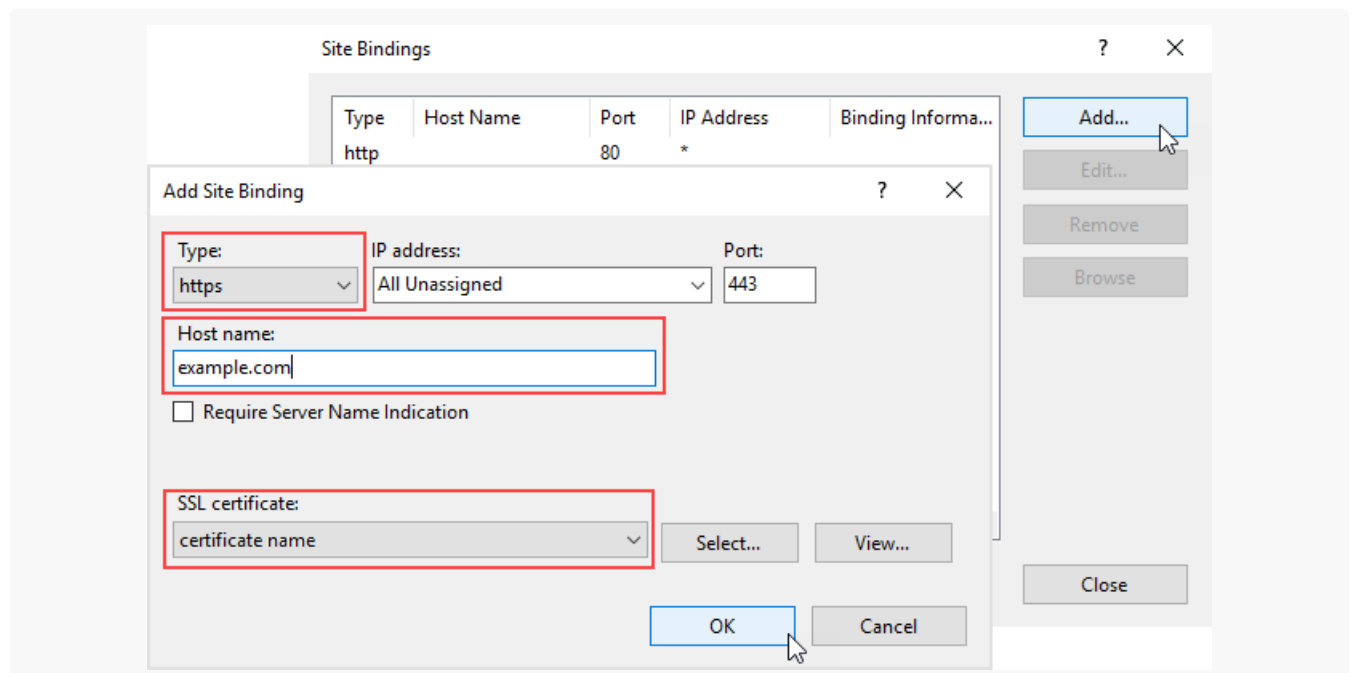
2. Click the [*Bindings*] link in the action menu (Fig. 5).

Fig. 5 Selecting website bindings



3. In the website bindings menu, click [Add] and add a new binding. In the [Add Site Binding] window, specify:
 - a. Type "https "
 - b. Website address
 - c. SSL certificate (Fig. 6).

Fig. 6 Binding a certificate to the Creatio website



- Click [OK] to confirm the settings.

Now the certificate is successfully bound to the web application.

Web.config setup

After adding the certificate, you need to make changes to the Web.config configuration file, located in the **root directory** of the Creatio website.

- At the end of the file, find:

```
<behaviors configSource="Terrasoft.WebApp\ServiceModel\http\behaviors.config" />
<bindings configSource="Terrasoft.WebApp\ServiceModel\http\bindings.config" />
```

- Change paths from "http" to "https":

```
<behaviors configSource="Terrasoft.WebApp\ServiceModel\https\behaviors.config" />
<bindings configSource="Terrasoft.WebApp\ServiceModel\https\bindings.config" />
```

Edit the Web.config file located in the **Path to the root website directory\Terrasoft.WebApp** directory.

- Set the variable value to encrypted="true". The configuration differs depending on the operating system of the server with Creatio application.

For Windows Server 2012 and higher, the configuration string should look as follows:

```
<wsService
  type="Terrasoft.Messaging.MicrosoftWSService.MicrosoftWSService,
  Terrasoft.Messaging.MicrosoftWSService" encrypted="true"
  portForClientConnection="443" maxConnectionNumber="100"
  clearIdleSession="false" clearIdleSessionInterval="120" />
```

- At the end of the file, find:

```
<services configSource="ServiceModel\http\services.config" />
```

- Change the path from "http" to "https":

```
<services configSource="ServiceModel\https\services.config" />
```

Save the configuration files.

Restart the application in the IIS and then go to your Creatio website. If all is done right, then in the address bar you will see "https://" before the web address of the application.

