

Data Exchange Framework Tenant Service 4.0 Installation Guide

A guide to installing the Data Exchange Framework Tenant Service

December 3, 2019



Table of Contents

1. Introduction	3
1.1. Requirements	3
1.2. Install Tenant Service on a Sitecore server	3
1.2.1. Configure Sitecore and Tenant Service on different servers	4
1.3. Install the web-component of Tenant Service on IIS	4
2. Configure Tenant service	7
2.1. Caching	10
3. Use Application Insights Logging for Tenant Service	11

1. Introduction

The Data Exchange Framework Tenant Service gives you the ability to trigger data synchronization processes via web-based push notifications.

1.1. Requirements

This product requires:

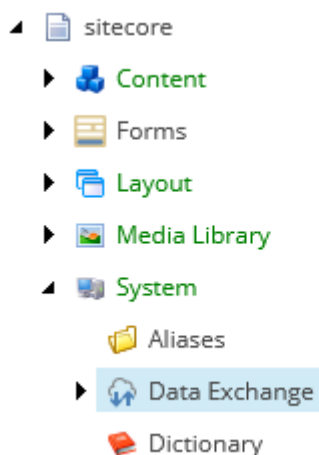
- Sitecore 9.3.0.
- configuration of the service on IIS.

1.2. Install Tenant Service on a Sitecore server

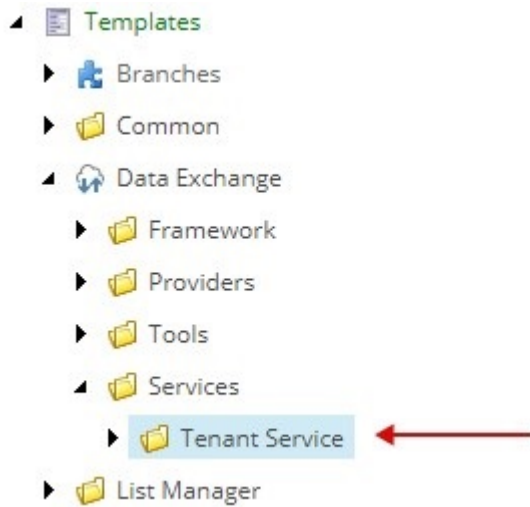
This section describes how to install Tenant Service (TS) on your Sitecore server.

You can run data synchronization on a content management (CM) server. You must install the entire Data Exchange Framework (DEF) product on the CM server.

1. Download the following packages from the [Sitecore Downloads page](#):
 - Data Exchange Framework 4.0.0
 - Sitecore Data Exchange Tenant Service 4.0.0
2. In Sitecore, in the Control Panel, click on **Install a Package**.
3. Install the *Data Exchange Framework* package.
4. Install the *Sitecore Data Exchange Tenant Service* Sitecore package.
5. You can confirm that the Data Exchange Framework is installed by opening Content Editor and locating the item `/sitecore/system/Data Exchange`.



- You can confirm that the Tenant Service is installed by opening the Content Editor and locating the item `/sitecore/templates/Data Exchange/Services/Tenant Service`.



1.2.1. Configure Sitecore and Tenant Service on different servers

If you have the Tenant Service installed on a different server than your Sitecore content management server, you must enable communication between the two servers:

- On your Sitecore Content Management server, in the `\App_Config\Sitecore\Services.Client\` folder, open the `Sitecore.Services.Client.config` file.
- Locate the following setting:

```
<setting name="Sitecore.Services.SecurityPolicy" value="Sitecore.Services.Infrastructure.Web.Http.Security.ServicesLocalOnlyPolicy,Sitecore.Services.Infrastructure"/>
```

Change the setting to:

```
<setting name="Sitecore.Services.SecurityPolicy" value="Sitecore.Services.Infrastructure.Web.Http.Security.ServicesOnPolicy,Sitecore.Services.Infrastructure"/>
```

The [Sitecore ItemService service](#) provides you with an API to read Sitecore items remotely. For example, the Tenant Service can use ItemService to read the configuration.

1.3. Install the web-component of Tenant Service on IIS

Tenant Service (TS) is a web-based service and must be configured as a web site via IIS. This chapter guides you through the process of installing Web-Based Micro TS. There are two alternative ways of installing TS:

- Manually
- Using PowerShell

Install manually

NOTE

For information on how to create certificates, please refer to the Set Up Client Certificates section of the Sitecore Experience Platform Installation Guide, available from the [Sitecore Downloads](#) page.

To install TS manually:

1. Download the Sitecore Data Exchange Framework Tenant Web Service package from the [Sitecore Downloads](#) page. It contains the Sitecore Data Exchange Framework Tenant Service 4.0 rev. 01410.scwdp.zip file.
2. Extract the .zip file into the website IIS root folder. By default, this folder is located at C:\inetpub\wwwroot\- 3. In the C:\inetpub\wwwroot\- 4. Locate the connection string named `sitecore`. If it is not there, add it. Update the connection string with the following properties:

Property	Content
database	Name of the database containing the item configuration. The default is <code>master</code> .
tenant ID	The tenant item ID for the tenant the micro service communicates with.
host	The host name of the Sitecore server, without <code>http</code> .
user name	A user name that can access the Sitecore server.
password	The user password.

For example:

```
<add name="sitecore"
connectionString="database=master;
tenant id={2A593000-2FDA-1234-1234-092F0CA338BD};
host=my-sitecore;
password=b;
user name=sitecore\admin;
timeout=20" />
```

5. Create and start the site in IIS.

NOTE

When Tenant Service is set up as described above, you can access it by requesting the `http://[tenantservice]` URL. If the service is running successfully, you will get a response showing the date and time.

6. On the Sitecore servers that connect to the tenant service, add a connection string to the server's App_Config\ConnectionStrings.config file, in the `connectionStrings` node. For example:

```
<add name="my-tenant" connectionString="https://my-sitecore.service"/>
```

NOTE

You can test the connection between the Sitecore server and the tenant service by requesting the `http://<tenantservice>/api/tenant` URL. The response displays the available tenant service endpoints.

Install using PowerShell and Sitecore Installation Framework

To install TS using SIF installation files and PowerShell:

1. Download the Sitecore Data Exchange Framework Tenant Web Service package from the [Sitecore download page](#).
2. Download the SIF installation scripts for Tenant Service package from the [Sitecore download page](#). It contains the following files:
 - `tenantservice-xp0.json` - an installation configuration file
 - `createcert.json` - an installation configuration file
 - `deploy.ps1` - a PowerShell deployment script
3. Unzip the installation script files to a folder on your installation.

NOTE

The `deploy.ps1` script contains a `SitecoreTenantID` parameter that is used to create the necessary connection string for the tenant service. If you already have a tenant service, you can edit the `deploy.ps1` script and change the `SitecoreTenantID` parameter to the [Tenant ID](#) of the tenant service. You can also add or change the `SitecoreTenantID` in the `connectionstrings.config` file later.

4. Open a PowerShell console with administrator rights. Navigate to the folder where you unpacked the installation script files. Run the `deploy.ps1` script with the following syntax:

```
.\deploy.ps1 -scInstanceName "<sitecore instance>" -servicePackage "<path to tenantservice.scwdp.zip file>"
```

The following table explains the parameters you need to supply when running the script:

Parameter	Description	Example
<code>scInstanceName</code>	The name of your Sitecore instance. The script will automatically name the Tenant service instance <code><scInstanceName>.service</code> . If you need to give the tenant service a different name, you must use the manual installation procedure.	<code>my-sitecore</code>
<code>servicePackage</code>	The location of the Sitecore Data Exchange Framework Tenant Web Service package.	<code>c:\tenant-scwdp.zip</code>

2. Configure Tenant service

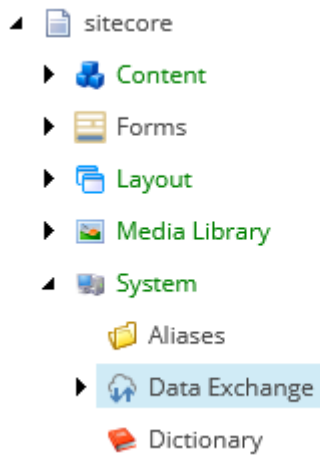
This topic guides you through the process of configuring Tenant Service (TS) on a Sitecore server.

NOTE

To enable the TS feature on a Sitecore server, you do not need to create a new tenant. You can extend an existing tenant with TS features by following points 5-9 in the procedure below.

To upgrade a tenant with TS:

1. Select the Data Exchange item **/sitecore/system/Data Exchange**.

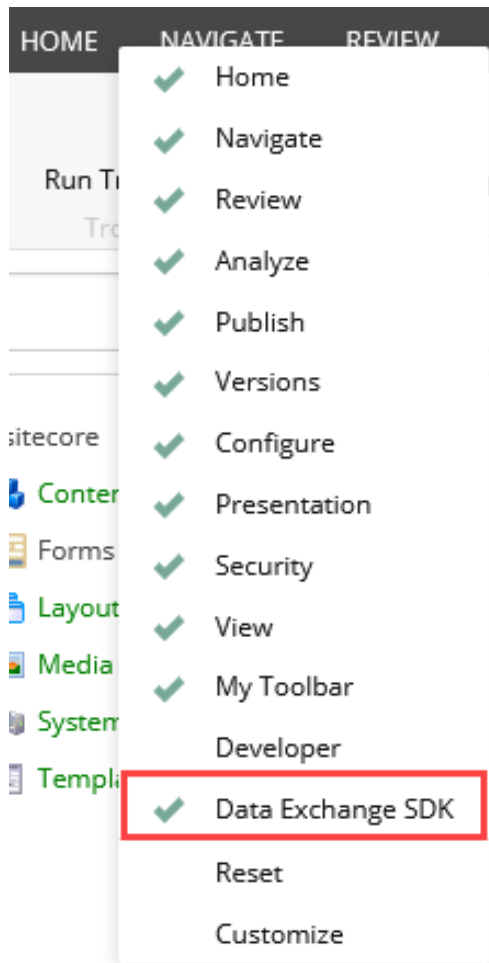


2. Create an **Empty Data Exchange tenant** using the **Insert** option.

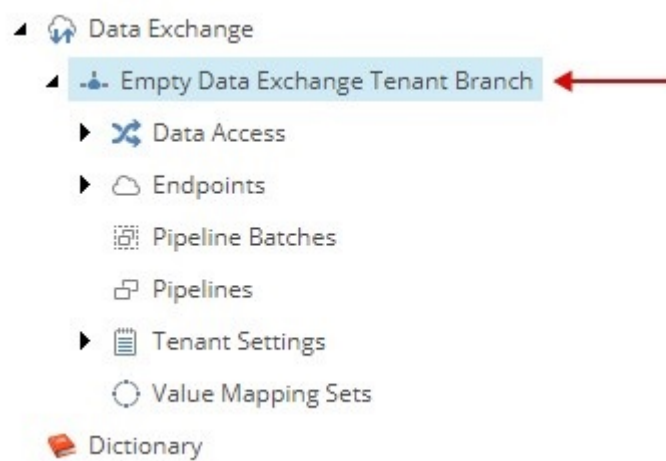
NOTE

You can also extend an already created tenant with the Tenant Service feature.

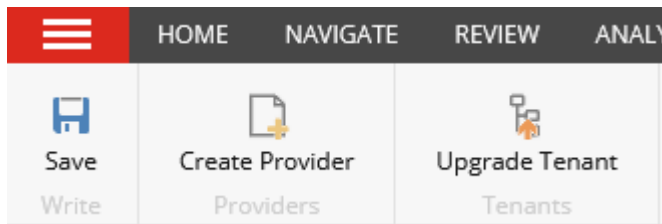
3. Right-click the menu ribbon, and verify that the **Data Exchange SDK** ribbon tab is enabled.



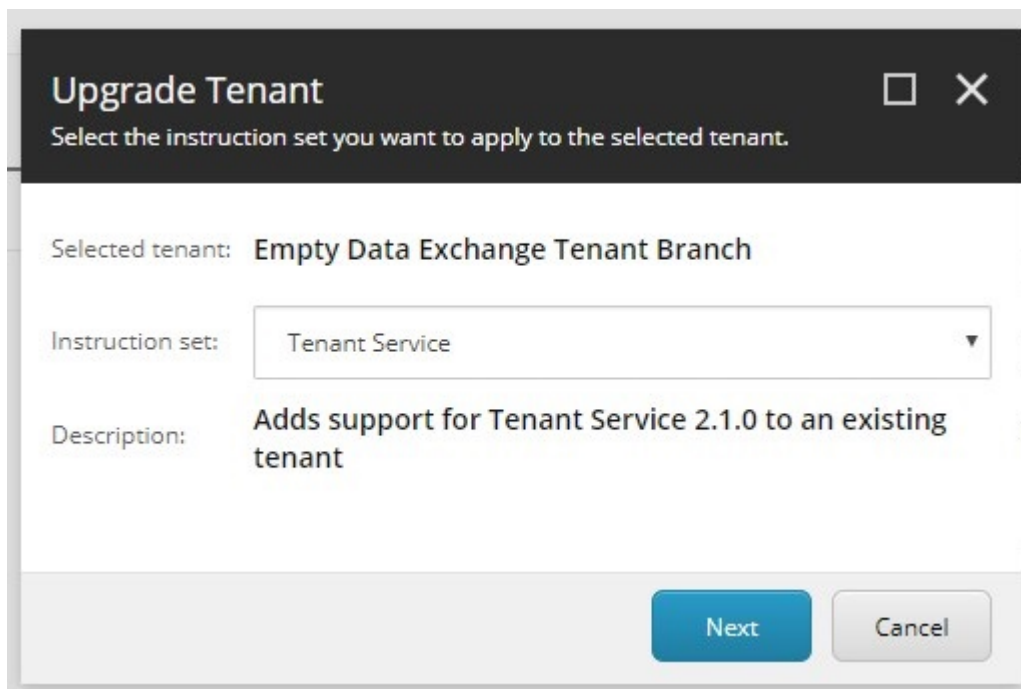
4. Select the newly created tenant item.



5. On the **Data Exchange SDK** ribbon click the **Upgrade Tenant** button.

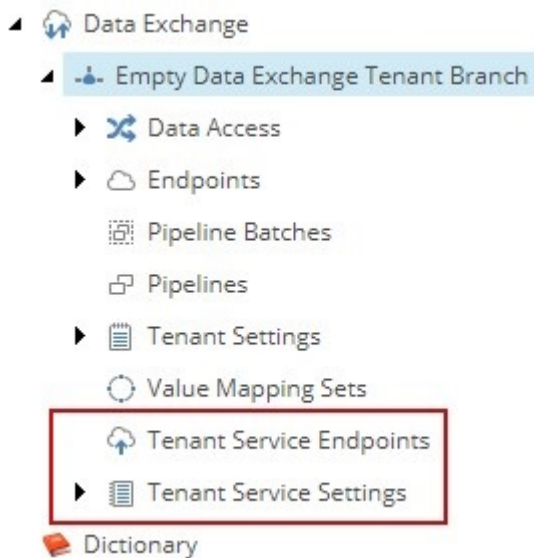


6. In the dialog window select **Tenant Service** in the **Instruction set** drop down menu.



7. Click **Next**.
8. Click **Start Upgrade**.

9. You can confirm the tenant was upgraded by finding the **Tenant Service Endpoints** and **Tenant Service Settings** items under the upgraded tenant.



10. On the server, navigate to the root folder of the Sitecore installation, open the `web.config` file, and add the following `httpProtocol` to the `system.webServer` node:

```
<system.webServer>
  ...
  <httpProtocol>
    <customHeaders>
      <add name="Access-Control-Allow-Origin" value="http://
<tenant.service.hostname>"/>
      <add name="Access-Control-Allow-Headers" value="*" />
      <add name="Access-Control-Allow-Methods" value="*" />
    </customHeaders>
  </httpProtocol>
  ...
</system.webServer>
```

NOTE

In the example, replace `<tenant.service.hostname>` with the actual hostname of your tenant service, for example, `my-sitecore.service`.

2.1. Caching

Sitecore uses caching to reduce the number of requests to resolve Tenant Service Endpoints. The duration of the caching is set in the `\App_Config\AppSettings.config` file on the xConnect server. By default, Sitecore caches values for 300 seconds.

```
<!-- cache expiration in seconds-->
<add key="cacheExpiration" value="300" />
```

3. Use Application Insights Logging for Tenant Service

By default the Tenant Service (TS) logs all information into a text log file in the `\App_Data\Logs\` folder.

You can install *AppInsights logger for Tenant Service* in order to log all TS activities and exceptions into the Azure AppInsights service.

To install the plugin:

1. Backup your current `C:\inetpub\wwwroot\<service folder>\App_Data\config\sitecore\CoreServices\sc.serilog.xml` file.

NOTE

Backing up this file enables you to roll your installation back to standard flat file logging.

2. Download the *AppInsights logger for Tenant Service* package from the [Sitecore download page](#).
3. From the downloaded package, extract the `\website\bin` folder into your `C:\inetpub\wwwroot\<service folder>\bin` folder.

To configure AppInsights logging:

1. Log in to your `portal.azure.com` subscription.
2. Create an AppInsights resource or locate an existing one, if you already have one.
3. In the **Overview** section, make a note of the instrumentation key.
4. Open the `c:\inetpub\wwwroot\<service folder>\App_Config\AppSettings.config` file. Add the instrumentation key:

```
<add key="AppInsightsKey" value="<instrumentation key>" />
```