



SyncIT

Installation and Administration GUIDE

HISTORY RESERVATION

Date	Version	Change	Name
28.03.2014	0.1	First Version of the document was prepared.	Roman Rudenko
23.05.2017	0.2	Second Version of the document was prepared.	Alexandra Petrova

CONTENTS

1 DESCRIPTION	4
2 HARDWARE REQUIREMENTS	4
3 SOFTWARE REQUIREMENTS	4
4 NETWORK REQUIREMENTS	4
5 INSTALLATION	5
6 CONFIGURATION	7
7 USING ENGINE	12

1 DESCRIPTION

This document describes how to install and configure SyncIT engine.

2 HARDWARE REQUIREMENTS

The following table lists the minimum recommended hardware requirements:

Component	Requirements
Processor	Dual core processor or 2 virtual cores
Memory	4-GB RAM or more
Hard Disk	1 GB of available hard disk space

Note: Actual hardware requirements may vary based on your specific integration requirements. The number and frequency of integrations can cause the processor and memory requirements to increase.

3 SOFTWARE REQUIREMENTS

The following table lists the software requirements:

Component	Requirements
Operating System	Windows Server 2012/2012 R2/2016 (recommended) Windows 10 (64-bit) Windows 8.1 (64-bit) Windows 7 (64-bit)
.NET	Microsoft .NET Framework 4.6
Windows Features	Windows Identity Foundation (Only for connection to CRM)

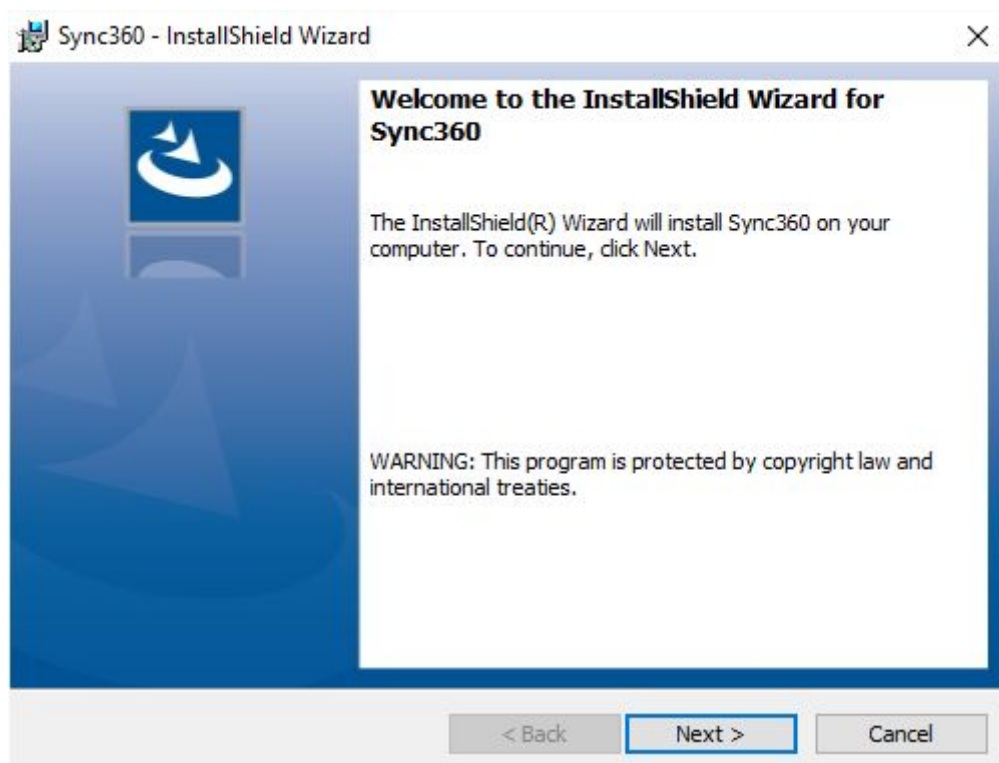
4 NETWORK REQUIREMENTS

The only network requirements are to make sure that the users who will be configuring / testing the SyncIT integration scripts and the service account used to start the SyncIT service have the ability to access any network resources / applications required for integration with the SyncIT server. This includes file shares, databases, clouds apps (Dynamics CRM Online), etc.

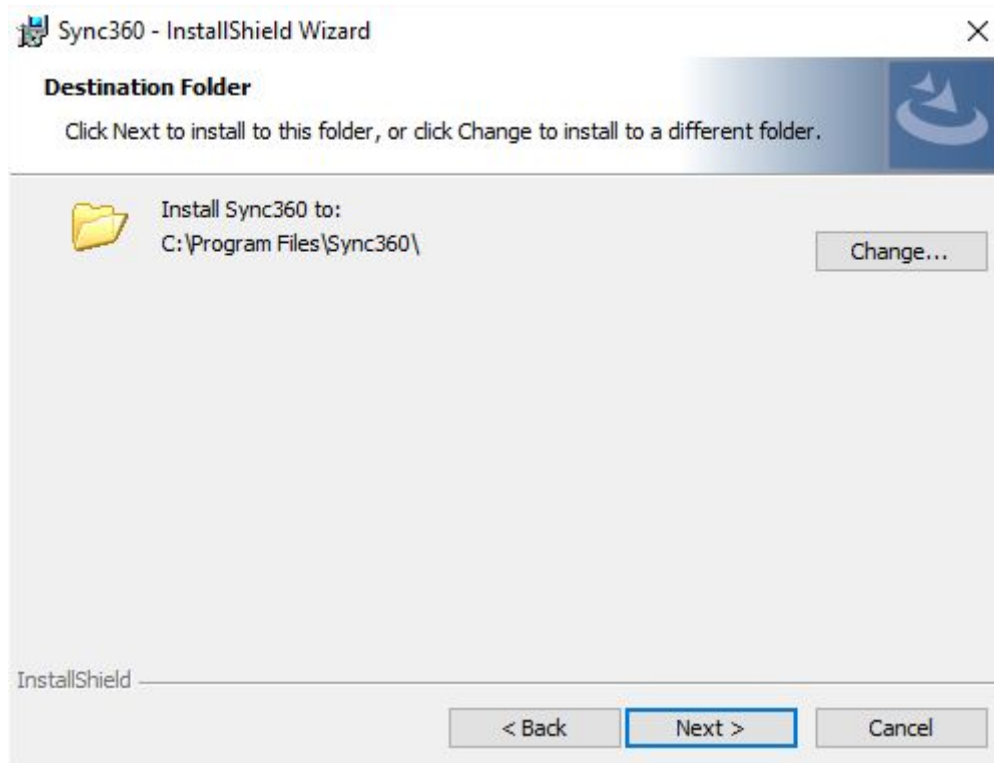
5 INSTALLATION

SyncIT engine is shipped with one exe file. The installation is very straightforward.

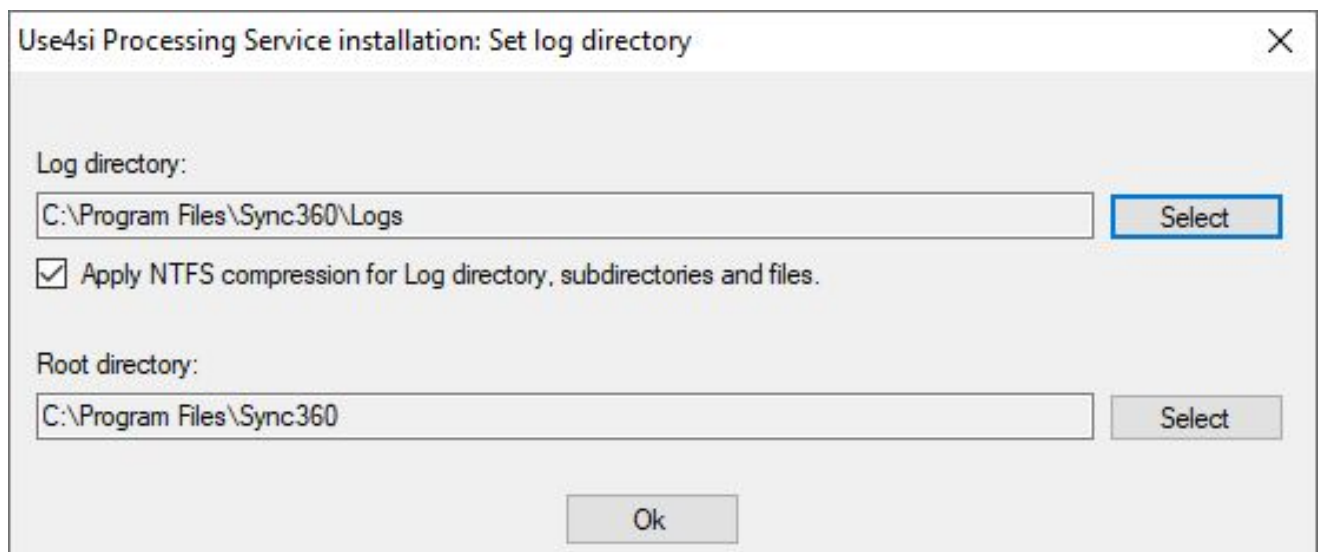
- 1) When you execute the file this screen will be displayed. Click Next



- 2) Specify installation folder or use the default one. Click Next



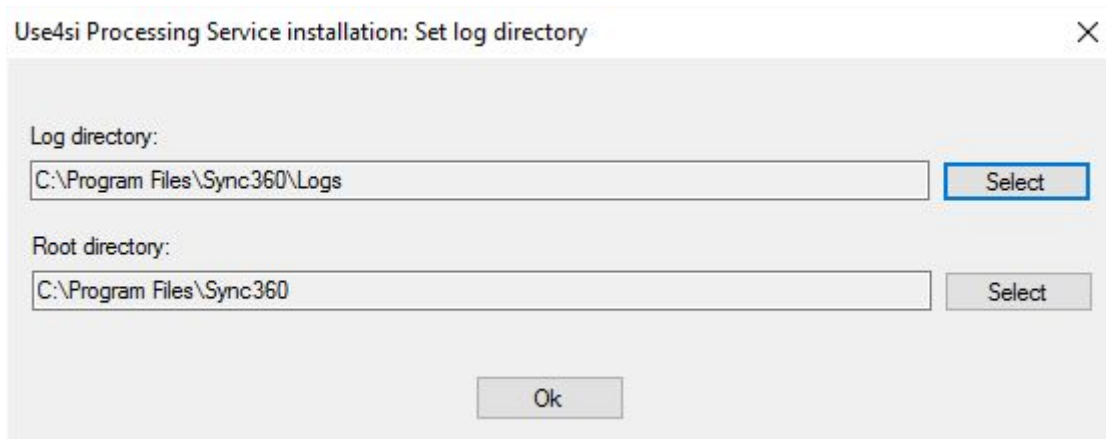
- 3) Click Install on this screen to begin installation.



- 4) During installation the dialog will be displayed asking to specify location for Log files and the Applications files root directory. Use the default or change it to any other, after that click Ok.

Note: regardless of where you specify the Root and Log directories, you will need to grant the users who need to maintain and troubleshoot the SyncIT scripts full access to these folders (read, write, create, delete).

This is especially true if you accept the default since local admin rights are required to modify files in the Program Files folder. It will be the best practice to create a group for SyncIT administrators, add users to this group, and grant the group the necessary access to these folders.



Use4si Processing Service installation: Set log directory

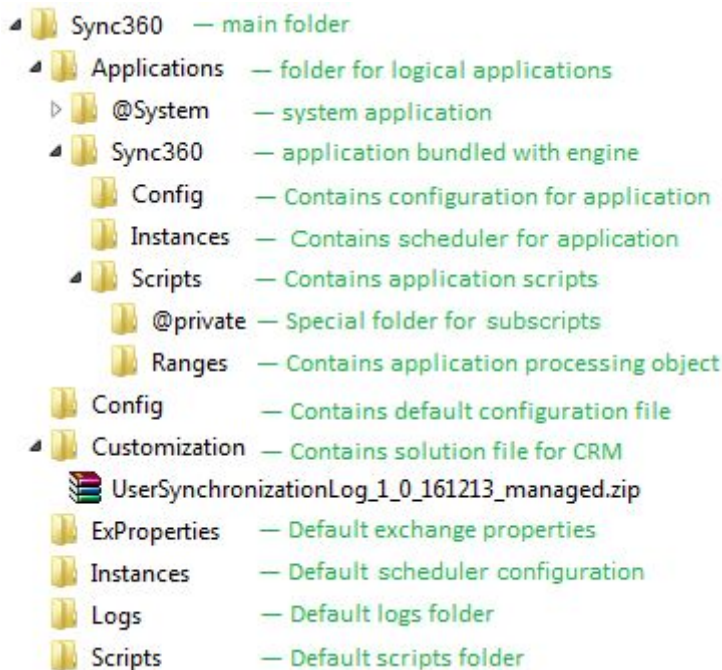
Log directory:
C:\Program Files\Sync360\Logs Select

Root directory:
C:\Program Files\Sync360 Select

Ok

6 CONFIGURATION

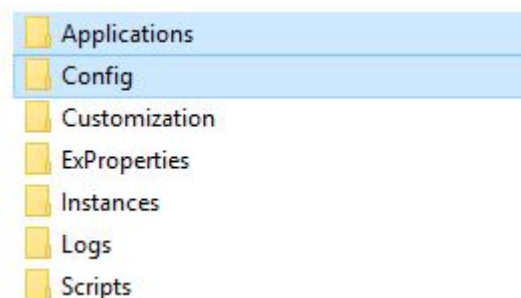
After the successful installation of SyncIT, let's take a look at the program folder structure:



The main **SyncIT** folder contains .dll and .exe files, particularly, the engine for executing the scripts – **Use4si.Develop.exe**. It will be described in the next part.

The main **SyncIT** folder also contains the following folders: **Applications**, **Config**, **Customization**, **ExProperties**, **Instances**, **Logs** and **Scripts**.

Let's consider the first two of them – **Applications** and **Config**:



1. In the **SyncIT\Config** folder there is a **default.xml** configuration file which contains connection to the engine's SQL database:

```
<connections>
  <add
    name="configdb"
    type="db"
    version="1.0"
    user="sa"
    password="uGBLYTSgbM/57flvboBZHA=="
    url="ProviderType=System.Data.SqlClient;Data
Source=192.168.1.94;Initial Catalog=use4si;uid={user};pwd={password}"
    org=""/>
  </connections>
```

To allow the engine to connect to the correct data sources, you should manually modify the configuration file. It may be **default.xml** file from the **SyncIT\Config** folder or **default.xml** from the **SyncIT\Applications\[nameofapplication]\Config** folder which will be described later.

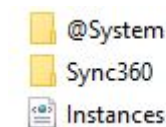
2. The **SyncIT\Applications** folder, as the name implies, contains applications. In Sync360 there is a logical concept of Applications. Every application consists of three parts, which are represented as folders – **Config**, **Instances** and **Scripts**:



- **Config** – contains information for data source connections;
- **Instances** – contains information about scheduling for each application script;
- **Scripts** – contains all script files.

By default, the **SyncIT\Applications** folder contains two predefined applications:

- **@System** – is used by engine and should not be modified;
- **SyncIT** – its main purpose is the contacts synchronization between CRM and Exchange.



The **SyncIT\Applications** folder also contains an **Instances.xml** file that defines which applications should run under service.

Instances.xml file has the following structure:

```
<?xml version="1.0" encoding="utf-8" ?>
<instances>
    <add>
        [instanceconfiguration]. [nameofapplication]@[datasourceconfiguration]
    </add>
</instances>
```

Instead of the [instanceconfiguration], [nameofapplication], [datasourceconfiguration] you should enter the real folder's and files' names.

During the script execution the engine does the following:

1. Try to find the **[instanceconfiguration].xml** file in the **SyncIT\Applications\[nameofapplication]\Instances** folder. If it's not found it will check for this file in the default **Instances** folder, and if it's not found there too, the scheduler configuration will be skipped.
2. Read the **SyncIT\Config\default.xml** file, then read the **SyncIT\Applications\[nameofapplication]\Config\default.xml** file and finally read the **Sync360\Applications\[nameofapplication]\Config\[datasourceconfiguration].xml** file.
All configuration files will be combined into one, in predefined order.

Example. By default, in the **Instances.xml** there is an example for the SyncIT application and **sample.xml** configuration file. The string adding an example application is commented:

```
<?xml version="1.0" encoding="utf-8" ?>
<instances>
    <!-- <add>main.Sync360@sample</add> -->
</instances>
```

Based on this example, let's take a look at what the engine will do at the runtime:

3. Try to find the **main.xml** file in the **SyncIT\Applications\Sync360\Instances** folder. If it's not found it will check for this file in the default **Instances** folder, and if it's not found there too, the scheduler configuration will be skipped.
4. Read the **SyncIT\Config\default.xml** file, then read the **SyncIT\Applications\SyncIT\Config\default.xml** file and finally read the **SyncIT\Applications\SyncIT\Config\sample.xml** file.
 All configuration files will be combined into one, in predefined order.

In the **SyncIT\Applications\SyncIT\Config** folder there is a **sample.xml** file that contains an example how to configure connection to CRM and Exchange.

```
<config>
  <connections>
    <add
      name="crm"
      type="crm"
      version="2011"
      user=""
      password=""
      url="http://localhost"
      org="Contoso"/>
    <add
      name="exchange"
      type="exchange"
      version="2010"
      user=""
      password=""
      domain=""
      url="https://localhost/EWS/exchange.asmx"/>
    </connections>
  </config>
```

Just specify correct values for the user, password, domain and URL to connect. For CRM also specify the organization name (org).

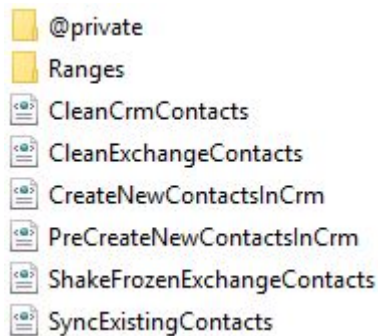
In the **SyncIT\Applications\SyncIT\Instances** folder there is a **main.xml** file that contains default scheduling for application scripts. For example, this is a part of this file:

```
<config>
  <schedule>
    <add name="SyncExistingContacts" task="ScriptScheduler" period="60">
      <param name="Criteria">SyncUsers</param>
      <param name="ScriptName">SyncExistingContacts</param>
      <param name="Interval">120</param>
    </add>
  </schedule>
```

</config>

Criteria is a script that defines objects for processing, ScriptName is an actual script name, Interval is how often script should be executed in minutes.

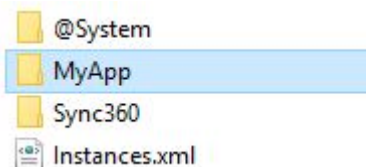
The **SyncIT\Applications\SyncIT\Scripts** folder contains all main scripts for SyncIT application and two subfolders **@Private** and **Ranges**.



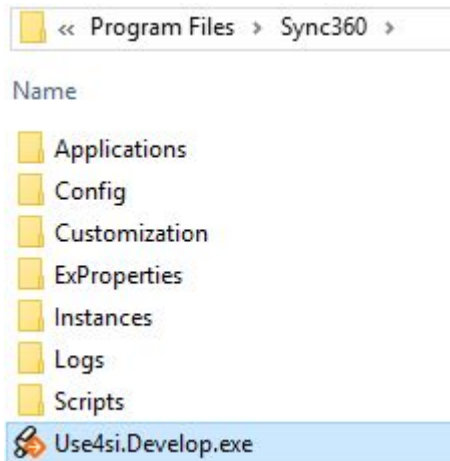
- **@Private** – this is special folder where engine tries to find scripts which are invoked by the Call operator;
- **Ranges** – this folder is used by scheduler to locate scripts which define objects for processing.

You can also create your own application.

1. Copy the **SyncIT\Applications\SyncIT** folder and paste it to the same folder.
2. Give it a name, for example, **MyApp**:

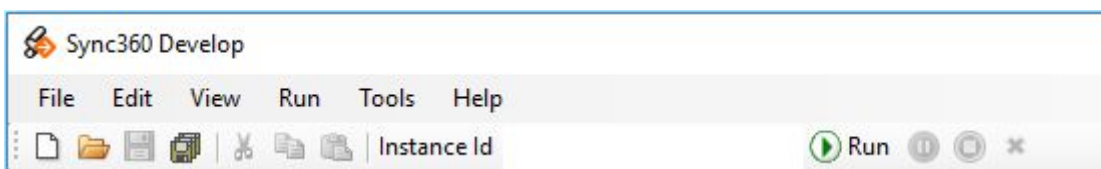


3. Return to the main **SyncIT folder**. The next step will be editing the files and running the scripts using the **Use4si.Develop.exe**:

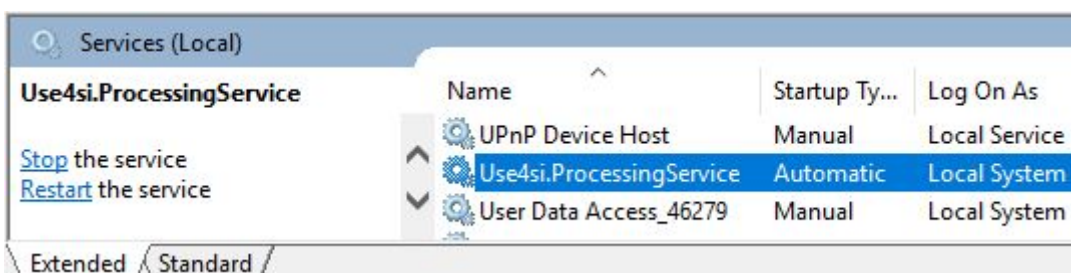


7 USING ENGINE

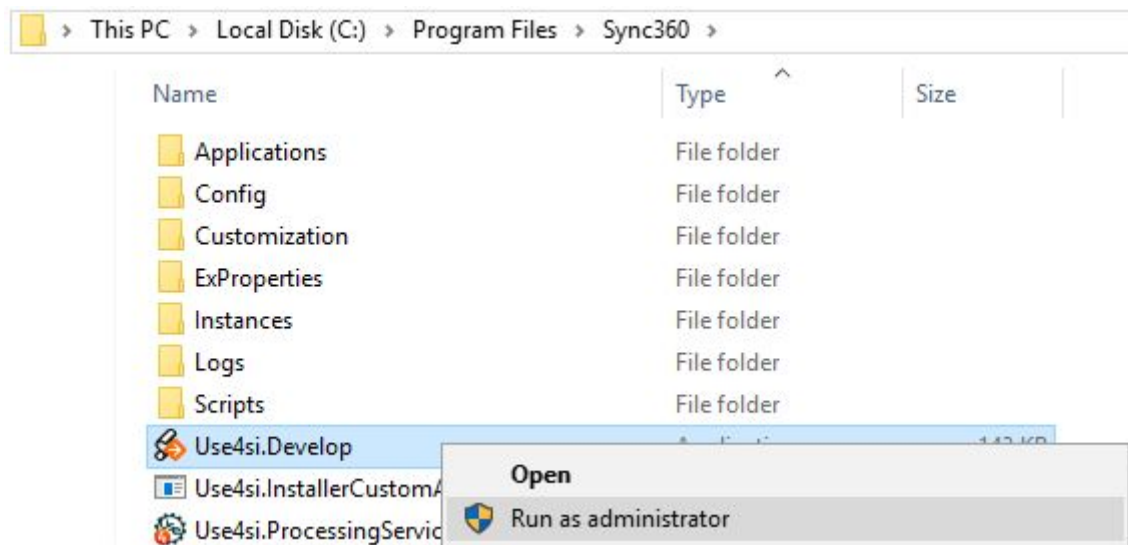
Use4si.Develop.exe is an .exe file that allows user to execute scripts through the developer interface:



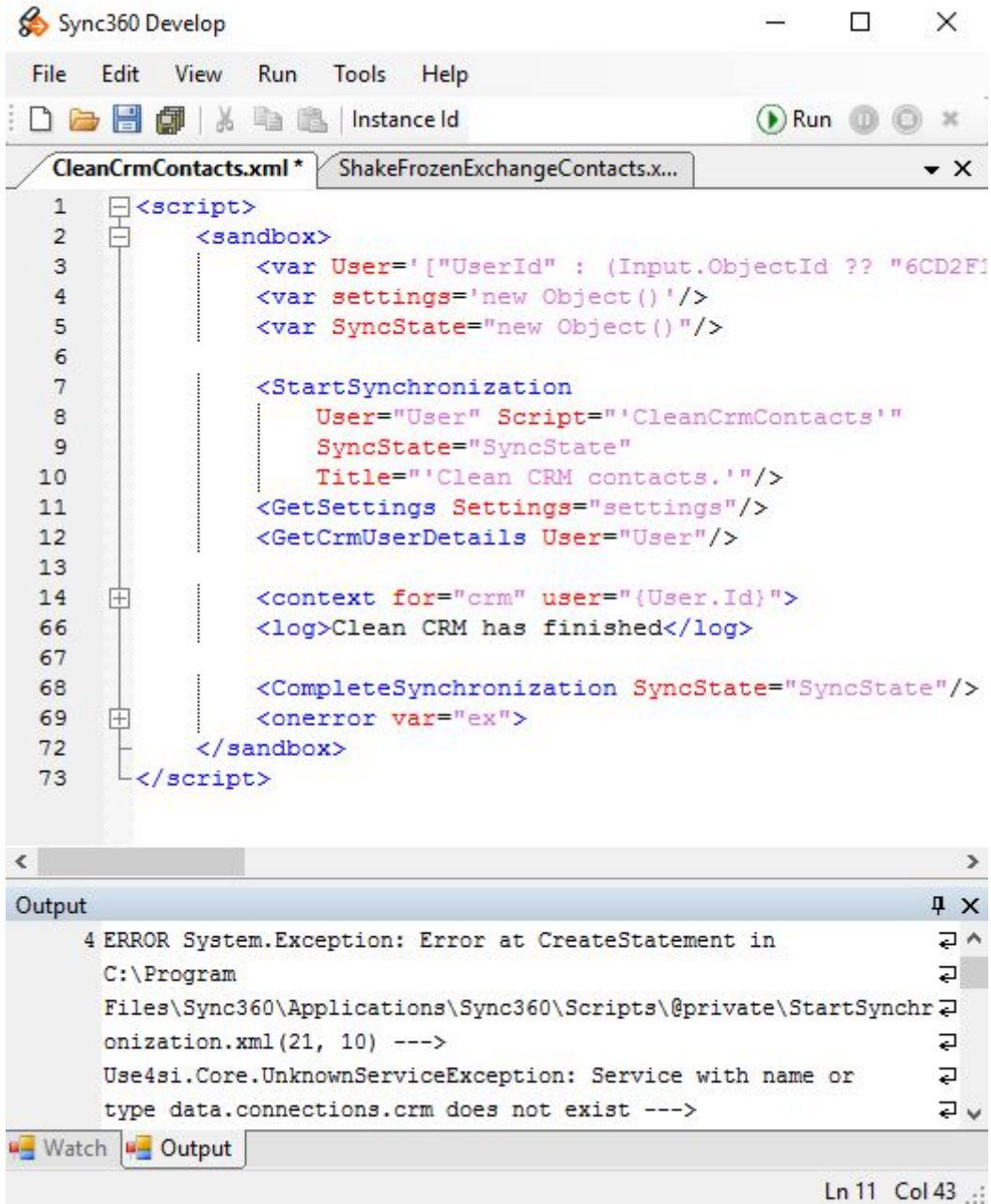
During installation a windows service is created, it's used for processing Applications as described in Configuration section. You can change the start state to automatic after all required preparation is done:



You can also find an .exe file **Use4si.Develop.exe** in the main SyncIT folder and run it as administrator:



An example of script execution:



The screenshot shows the Sync360 Develop IDE with a script named `CleanCrmContacts.xml` open. The script contains XML-like tags for a sandbox environment. The error message in the Output pane indicates a `System.Exception` at line 11, column 10, stating that a service with the name `data.connections.crm` does not exist.

```

1  <script>
2    <sandbox>
3      <var User='["UserId" : (Input.ObjectId ?? "6CD2F1
4      <var settings='new Object()' />
5      <var SyncState="new Object()" />
6
7      <StartSynchronization
8        User="User" Script="'CleanCrmContacts'"
9        SyncState="SyncState"
10       Title="'Clean CRM contacts.'" />
11      <GetSettings Settings="settings" />
12      <GetCrmUserDetails User="User" />
13
14      <context for="crm" user="{User.Id}">
66      <log>Clean CRM has finished</log>
67
68      <CompleteSynchronization SyncState="SyncState" />
69      <onerror var="ex">
72    </sandbox>
73  </script>
  
```

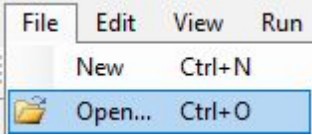





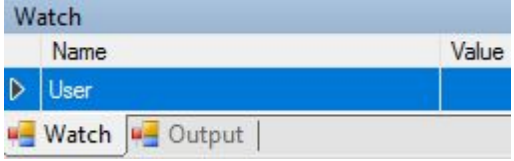
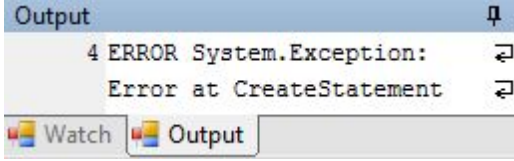
Output

```

4 ERROR System.Exception: Error at CreateStatement in
  C:\Program
  Files\Sync360\Applications\Sync360\Scripts\@private\StartSynchr
  onization.xml(21, 10) --->
  Use4si.Core.UnknownServiceException: Service with name or
  type data.connections.crm does not exist --->
  
```

Ln 11 Col 43

Let's take a look on the main features:

	<p>Open script file (for example, SyncExistingContacts.xml)</p>
	<p>You can edit multiple scripts</p>
	<p>Put a string in the format "instanceconfiguration.nameofapplication@datasourceconfiguration" (for example, sample@main) into the textbox</p>
	<p>Press to run the script</p>
	<p>Press to pause/stop the script</p>
	<p>Clear cache from last execution</p>
	<p>Here you can define variables to watch during the script execution</p>
	<p>Output window shows script execution results</p>

About

WaveAccess is a results focused software development company that provides high quality software outsourcing services to hundreds of emerging and established companies globally. We use our technical expertise to increase business efficiencies, optimize slow or unreliable systems, recover projects that have gone off track and bring ambitious ideas to life.

19

years of delivering
successful outcomes
for customers

350+

talented & passionate
professionals

4

global R&D
centers

9

industry verticals
from banking
to healthcare

300+

successful projects
delivered and counting

96%

of our customers
are repeat business

Las Vegas

headquarters

**USA, UK, Denmark
and Eastern Europe**

sales offices



Academy Award-winning
Mocha for Imagineer Systems



Silver
**Microsoft
Partner**



2011 PARTNER OF THE YEAR
Microsoft Dynamics Professional Services
CRM4Legal for Client Profiles
Winner



2017 Partner of the Year Winner
Business Analytics Award

**Microsoft
Partner**

2018 Partner of the Year
Artificial Intelligence Award

**If you need to develop
a similar project, please write us**

hello@wave-access.com

Read more at
wave-access.com