SIM v6.0

Release

Jul 19, 2017

Contents

Note: Documentation is still in development process. Please do not hesitate to contact us on support@silvermonkey.net for further information.

CHAPTER 1

Welcome

This document is meant to be a source for all information regarding the administration and installation of the Silver Monkey v6.1 engine.

Contents:

Requirements

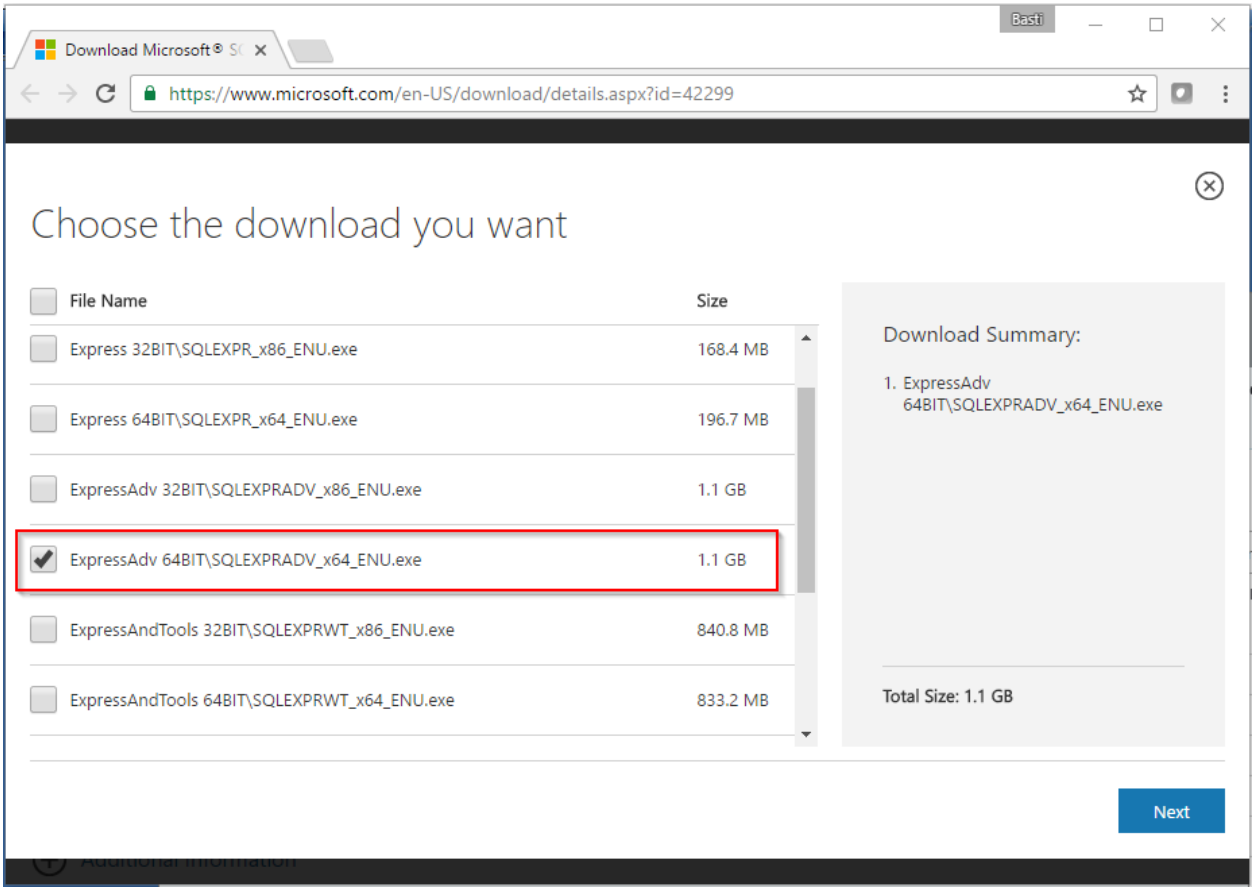
Application Server (IIS)

- Microsoft Windows Server 2012 or higher
- Internet Information Server
- Microsoft .NET Framework 4.6.1
  - https://support.microsoft.com/en-us/kb/3102436
  - With Windows Server 2012, the installation requires Windows Update KB2919355, which may be included in your update stream
  - If not already included, please install manually as described here: https://support.microsoft.com/en-us/kb/2919355

Database Server (SQL)

- Microsoft SQL Server 2012 or higher
- Or Microsoft SQL Server Express with Advanced Services
- 2012: http://www.microsoft.com/en-us/download/details.aspx?id=29062 (ENUx86SQLEXPRADV_x86_ENU.exe)
- 2014: https://www.microsoft.com/en-US/download/details.aspx?id=42299

Important: Make sure to download “ADV” package:



Server Hardware Requirements (IIS+SQL)

The system requirements for processors, RAM and hard disk space depend on the size of the correspondig ConfigMgr environment and the number of users working at the same time. Anyway, there is always the option to easily move the application to a more powerful machine or to distribute it across several servers with load balancing.

In addition to the requirements of the operating system, the following conditions arise:

- CPU 1GHz
- RAM 2GB
- Database size 500MB
- Website/Application files 50MB

(Valid for up to 10,000 systems and 20 concurrent users on the Web Application)

Workplace Systems

- Microsoft Internet Explorer 10 or higher

- Mozilla FireFox 5 or higher (Windows SSO is not supported by browser)
- Google Chrome

Installation

In this article:

Requirements

1. For general information on system requirements see *Requirements*.
2. SQL Service Account for accessing SIM SQL DB (in this article `sim-svc-sql`)

Important: Please install all requirements before beginning with this guide!

IIS Features

Execute the following command to enable IIS features on the application server:

```
CMD.EXE /C DISM.EXE /enable-feature /all /online /featureName:IIS-WebServerRole /
↪featureName:IIS-WebServer /featureName:IIS-CommonHttpFeatures /featureName:IIS-
↪StaticContent /featureName:IIS-DefaultDocument /featureName:IIS-DirectoryBrowsing /
↪featureName:IIS-HttpErrors /featureName:IIS-HttpRedirect /featureName:IIS-
↪ApplicationDevelopment /featureName:IIS-ASPNET /featureName:IIS-NetFxExtensibility /
↪featureName:IIS-ASPNET45 /featureName:IIS-NetFxExtensibility45 /featureName:IIS-ASP␣
↪/featureName:IIS-CGI /featureName:IIS-ISAPIExtensions /featureName:IIS-ISAPIFilter /
↪featureName:IIS-ServerSideIncludes /featureName:IIS-HealthAndDiagnostics /
↪featureName:IIS-HttpLogging /featureName:IIS-LoggingLibraries /featureName:IIS-
↪RequestMonitor /featureName:IIS-HttpTracing /featureName:IIS-CustomLogging /
↪featureName:IIS-ODBCLogging /featureName:IIS-Security /featureName:IIS-
↪BasicAuthentication /featureName:IIS-WindowsAuthentication /featureName:IIS-
↪DigestAuthentication /featureName:IIS-ClientCertificateMappingAuthentication /
↪featureName:IIS-IISCertificateMappingAuthentication /featureName:IIS-
↪URLAuthorization /featureName:IIS-RequestFiltering /featureName:IIS-IPSecurity /
↪featureName:IIS-Performance /featureName:IIS-HttpCompressionStatic /featureName:IIS-
↪HttpCompressionDynamic /featureName:IIS-WebDAV /featureName:IIS-
↪WebServerManagementTools /featureName:IIS-ManagementScriptingTools /featureName:IIS-
```

For easy deploymnet: `Download the script.`

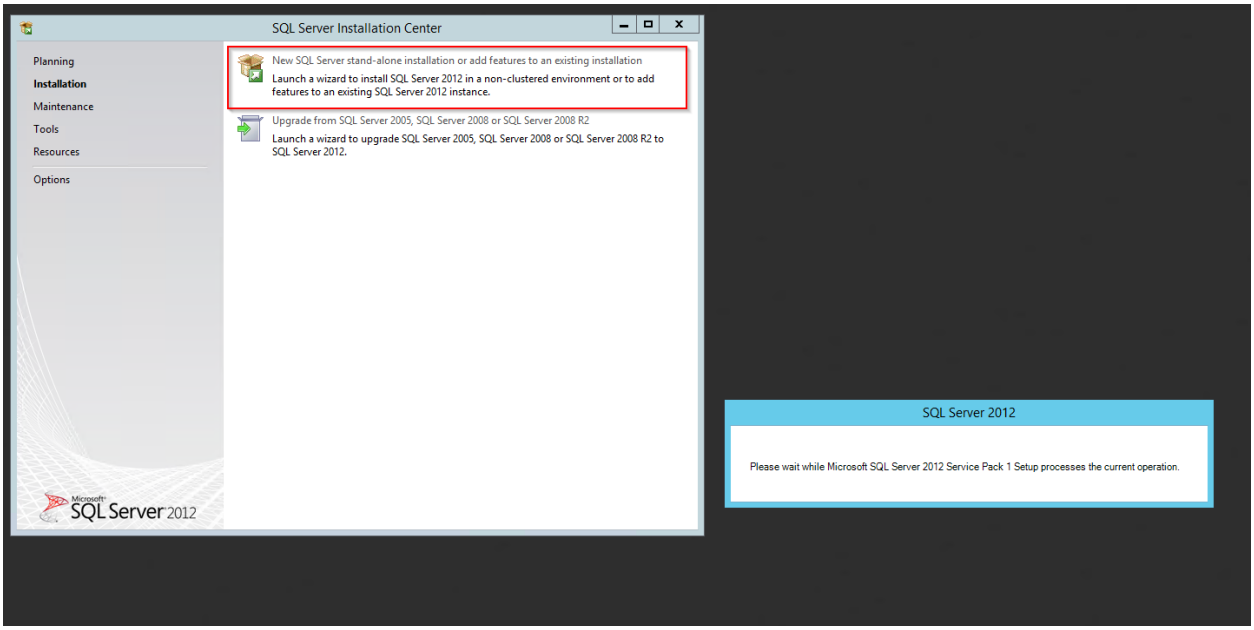
Microsoft SQL Server

For information about supported SQL Server versions see *Supported configurations*

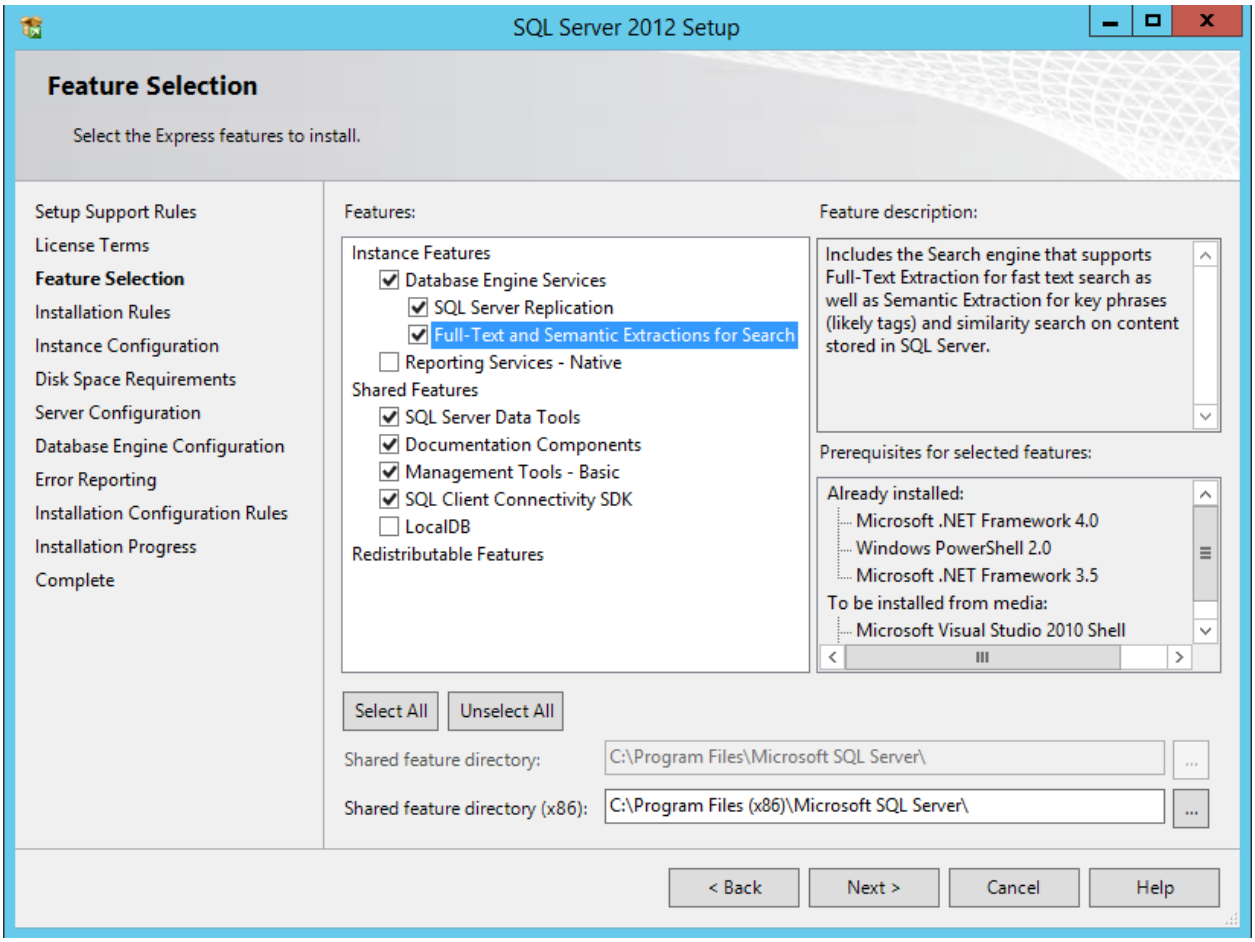
The installation of the SQL Server will be described in the following steps.

Installation Setup

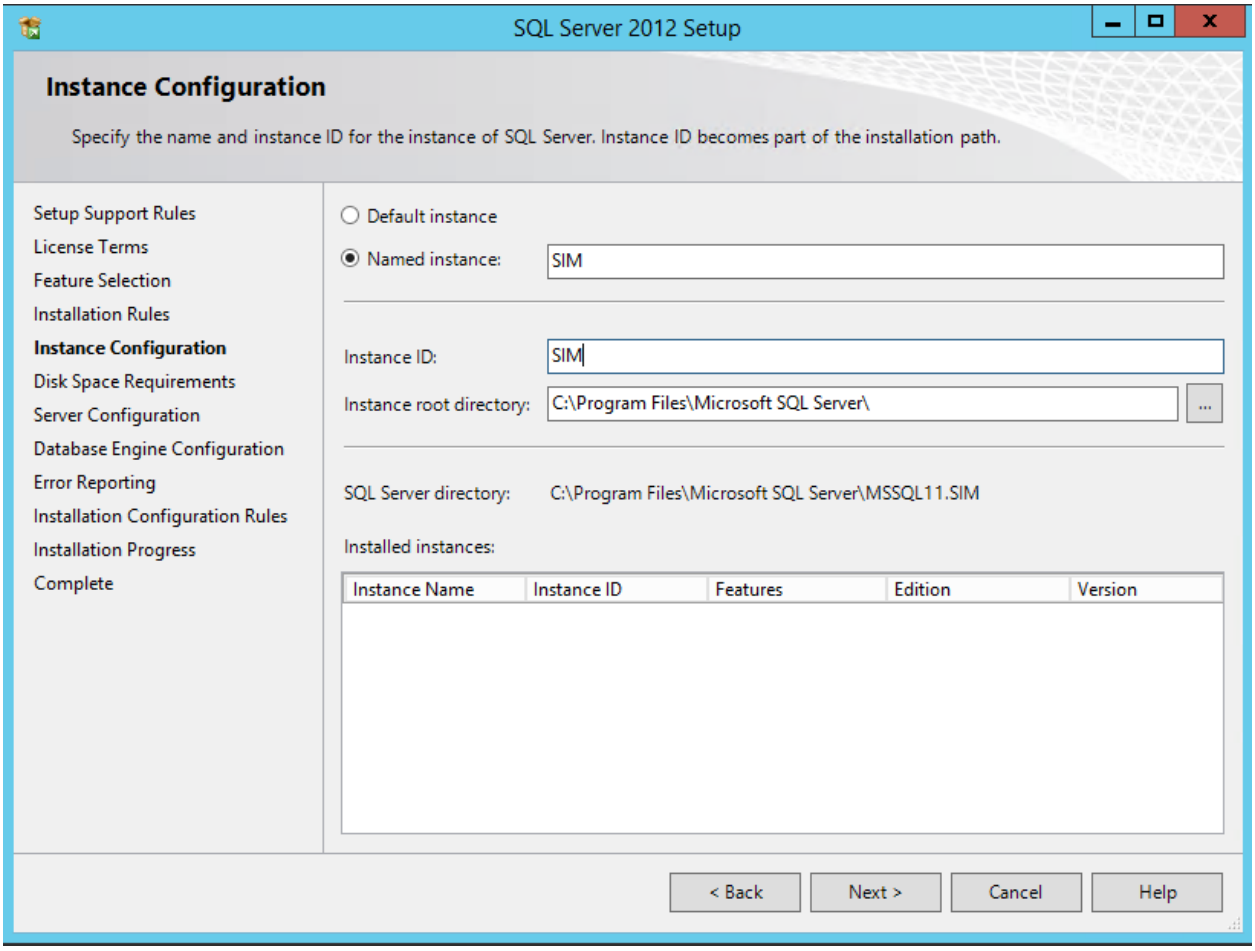
Start the SQL Server installation setup. Choose the “New SQL Server stand-alone installation...”-Option in the following Window:



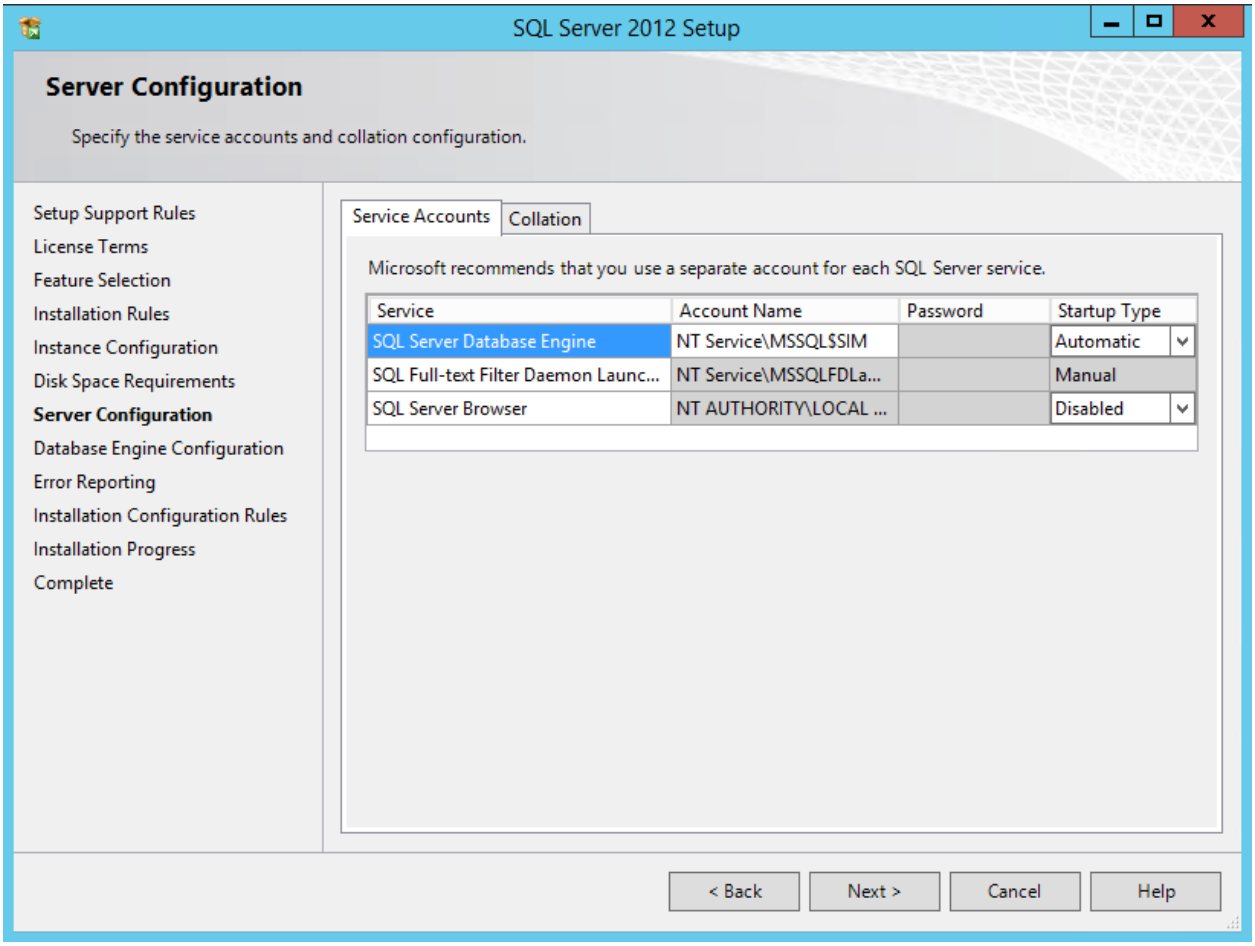
Throughout the installation, please choose the same features as shown below:



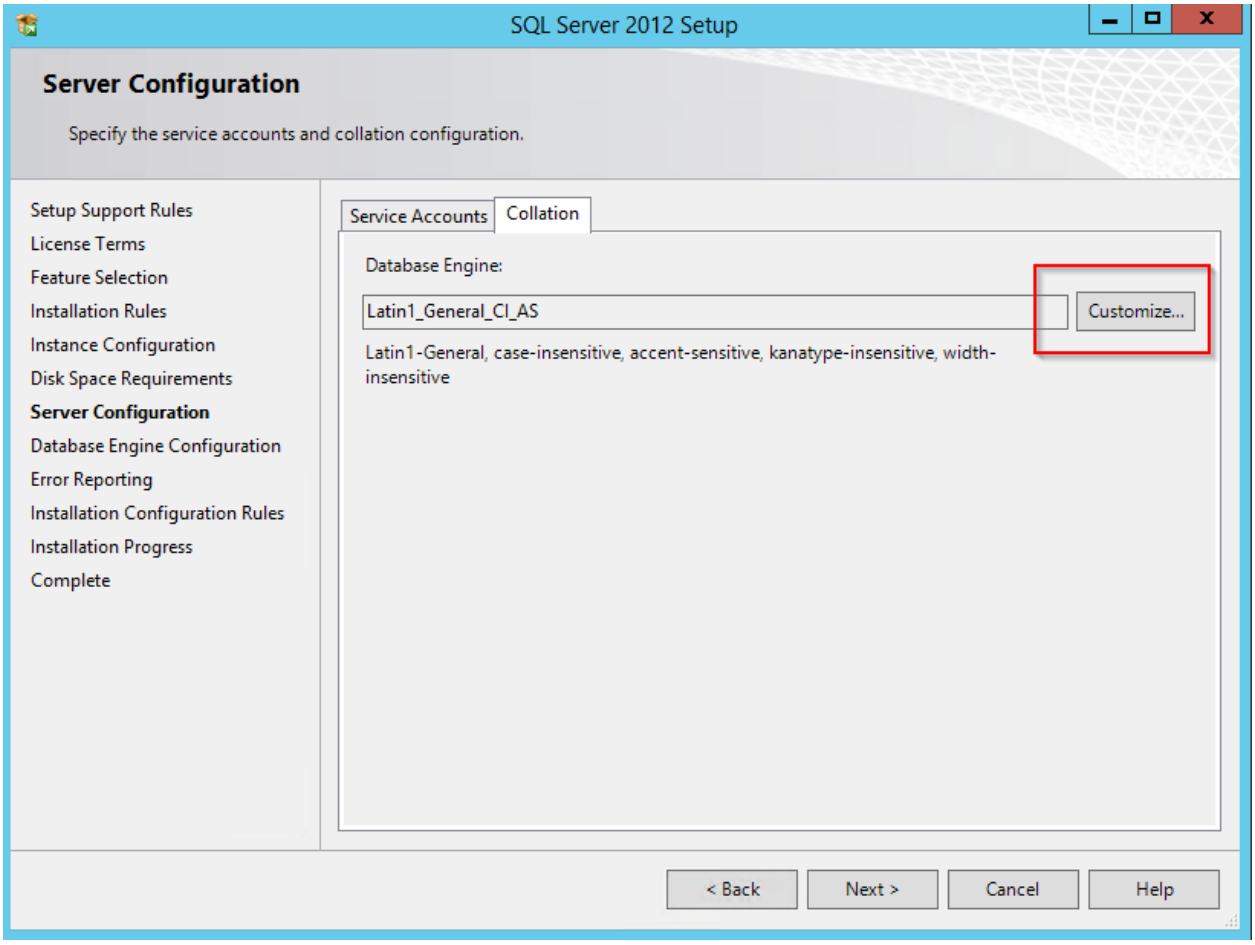
Name the instance SIM or choose another name:



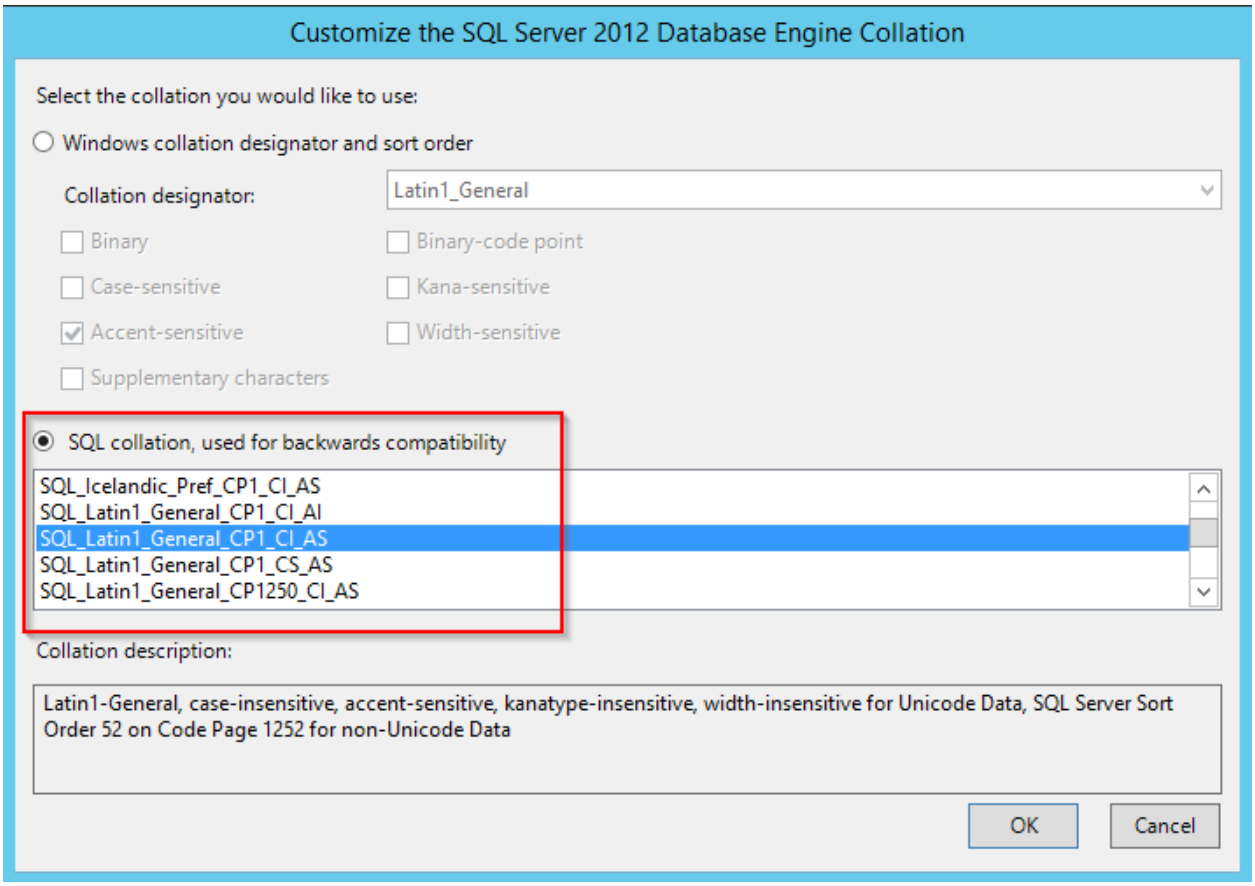
Configure the server as follows:



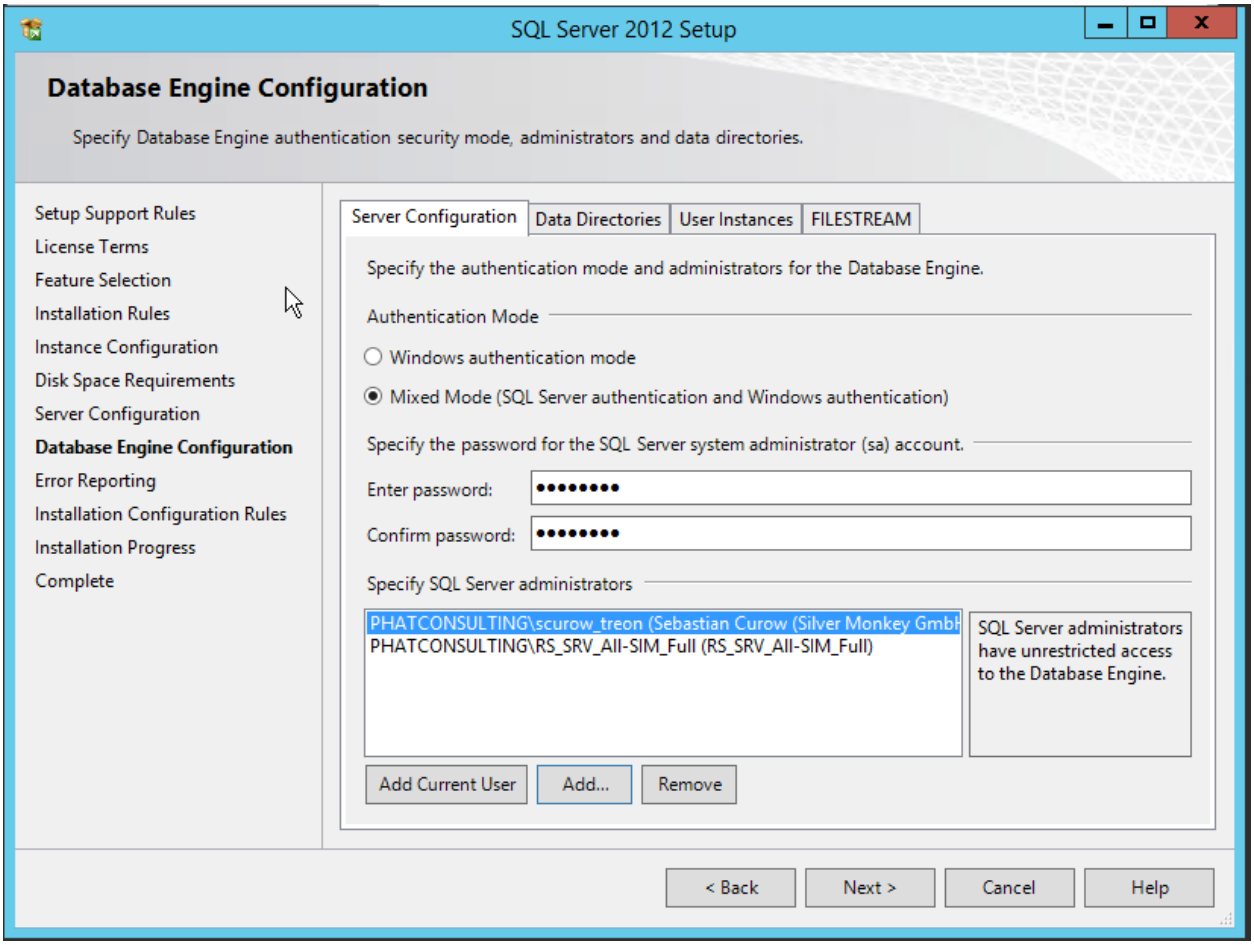
Customize the Database Engine



Choose the Database Engine called 'SQL_Latin_General_CP1_CI_AS':



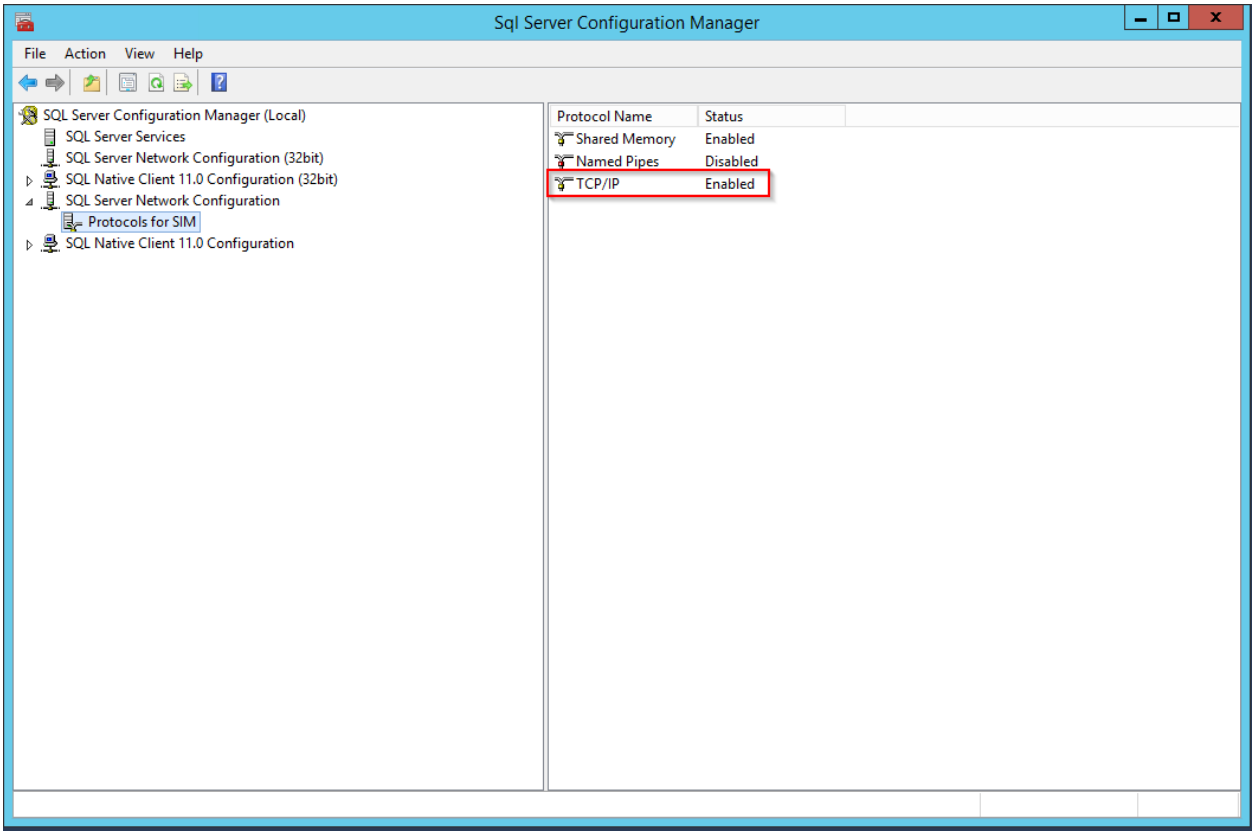
Select the 'mixed mode'-authentification and add your AD service account for SQL (`sim-svc-sql`) as SQL Server administrator:



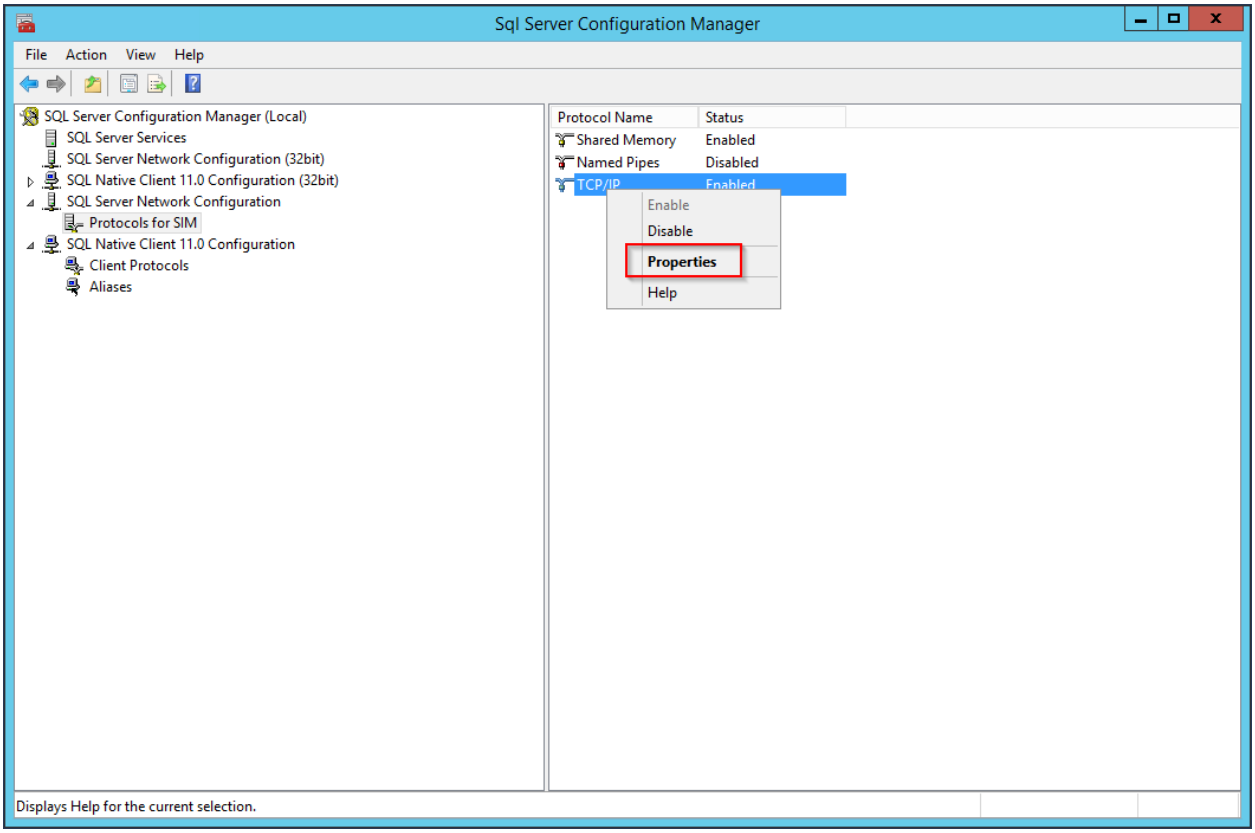
You have completed the setup!

SQL Server TCP/IP Configuration

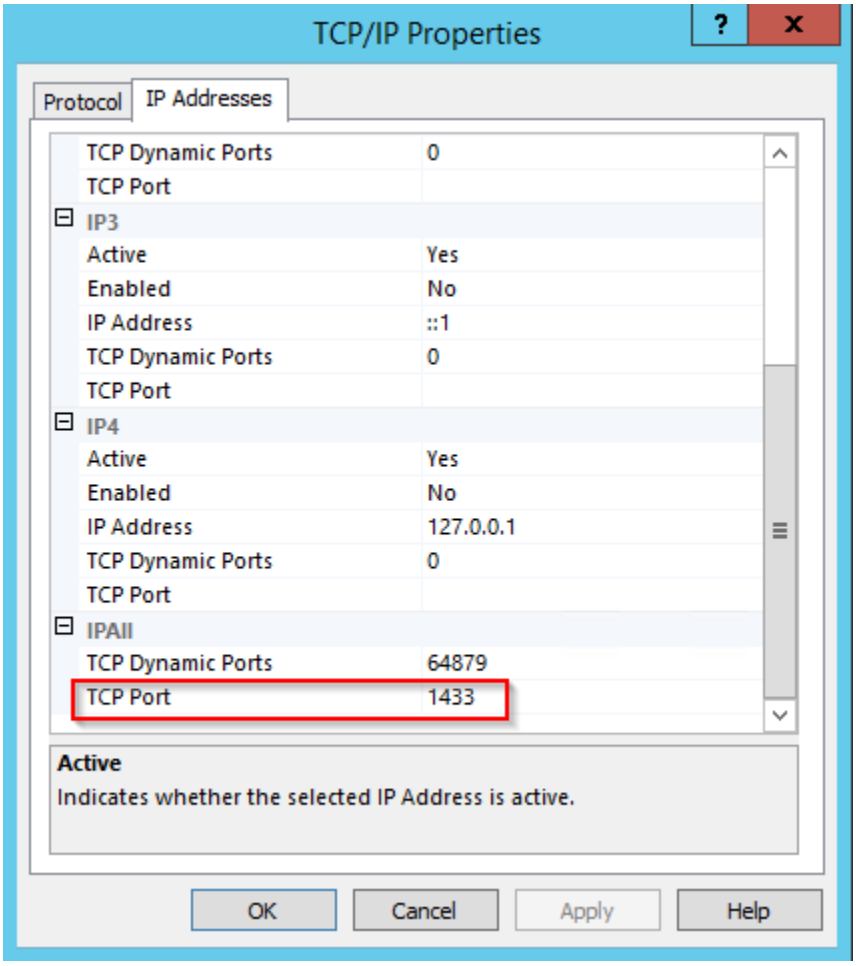
Open the SQL Server Configuration Manager, choose 'SQL Server Network Configuration' and then 'Protocols for [Database Name]'. Change the TCP/IP Status to *Enabled*:



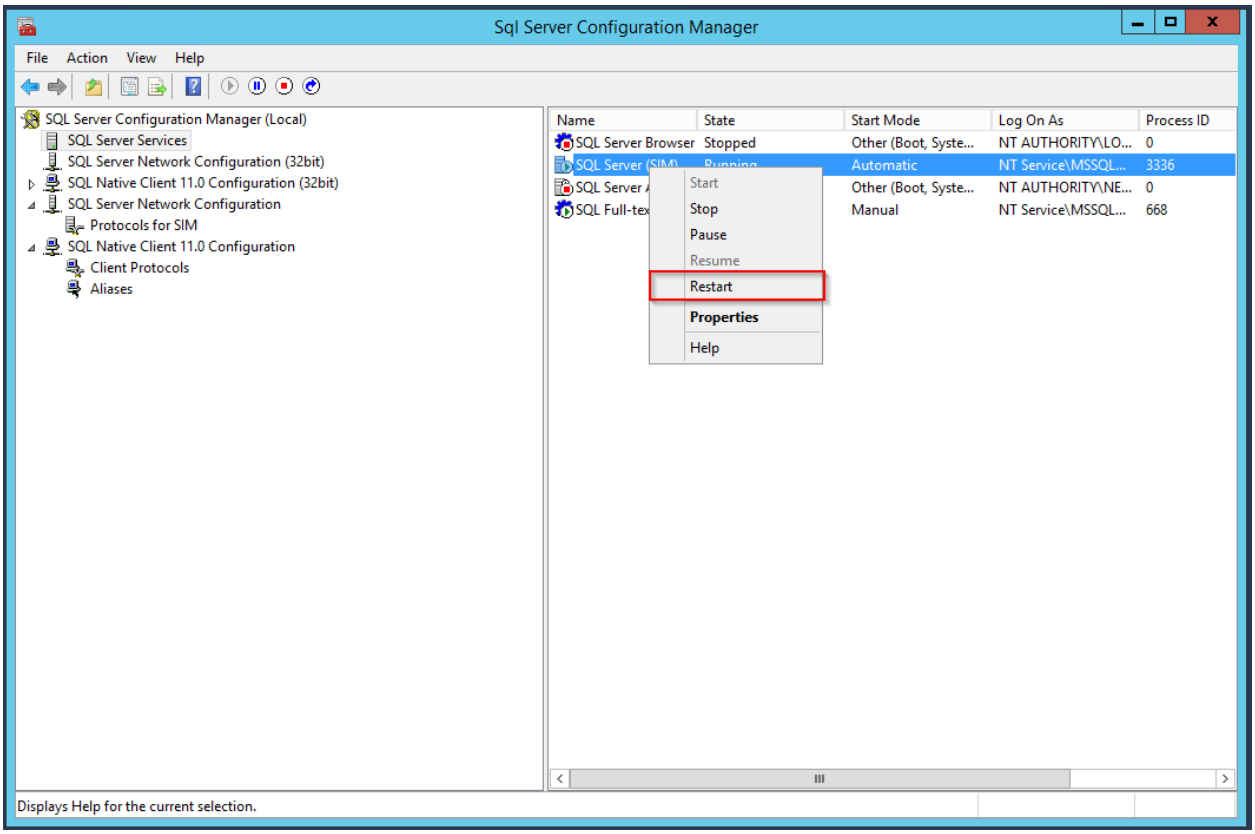
Right-click the TCP/IP line and choose 'Properties':



Choose the tab “IP Adresses” and change the ‘TCP Port’-entry to 1433:

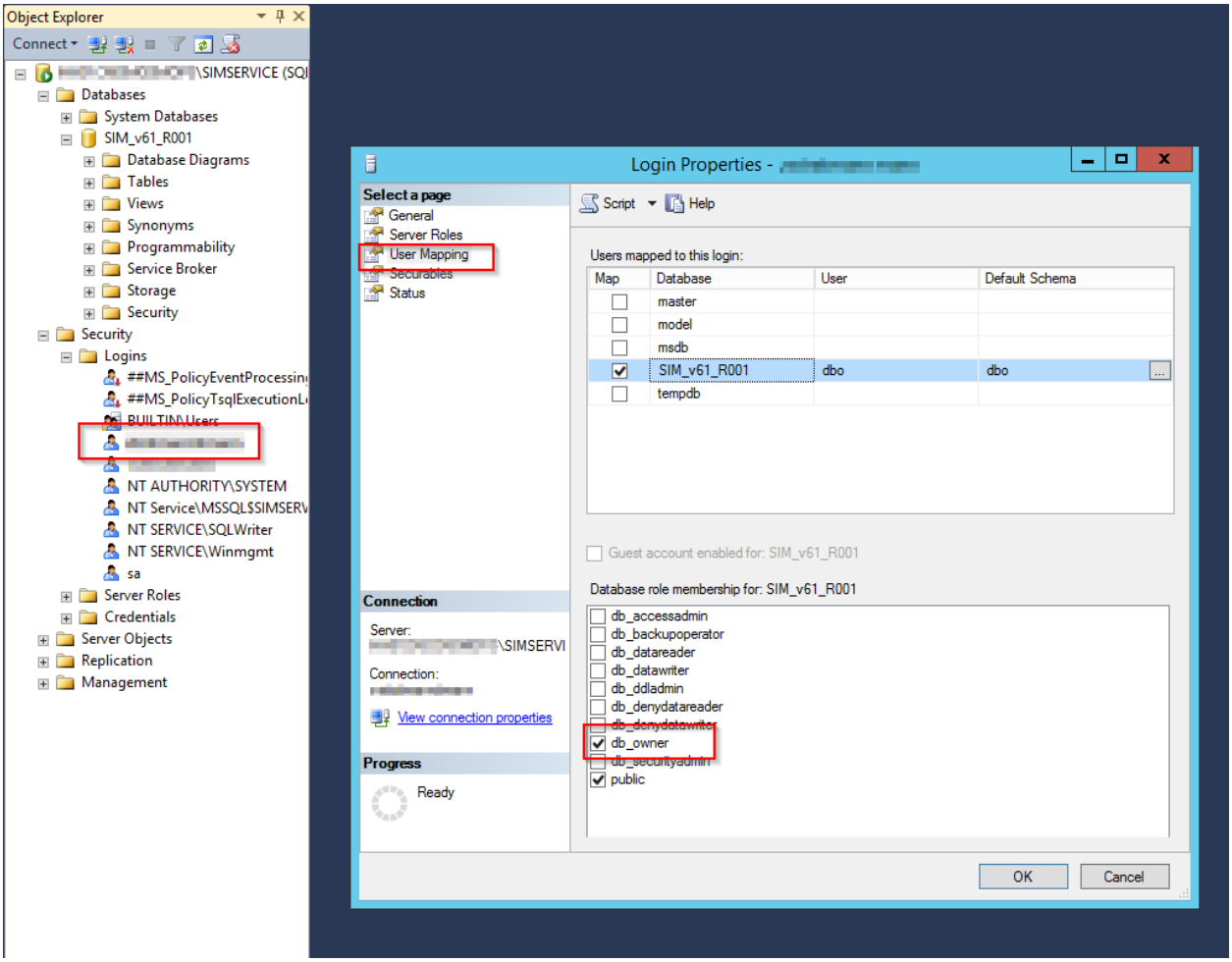


Afterwards, navigate to the SQL Server Services and restart the ‘SQL Server ([Database Name]):



SIM SQL DB Installation

1. Create database `SIM_v60_R001`
2. Grant SilverMonkey Service Account (`sim-svc-sql`) “db_owner” rights for the corresponding database

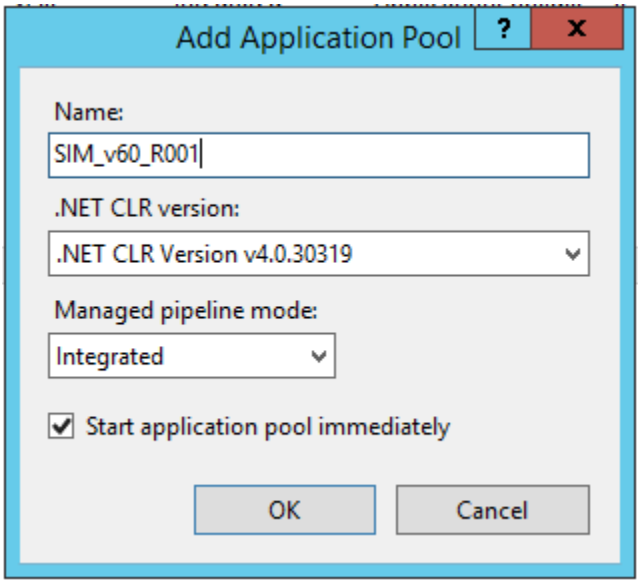


3. Import .SQL file from installation media (.\Database) into SQL Management Studio
4. Make sure the **USE** command aims to the correct database created above and execute script

Configure IIS

Create IIS App Pool

1. Go to IIS Manager and create an AppPool with .NET CLR version set to *v4.0*‘* :



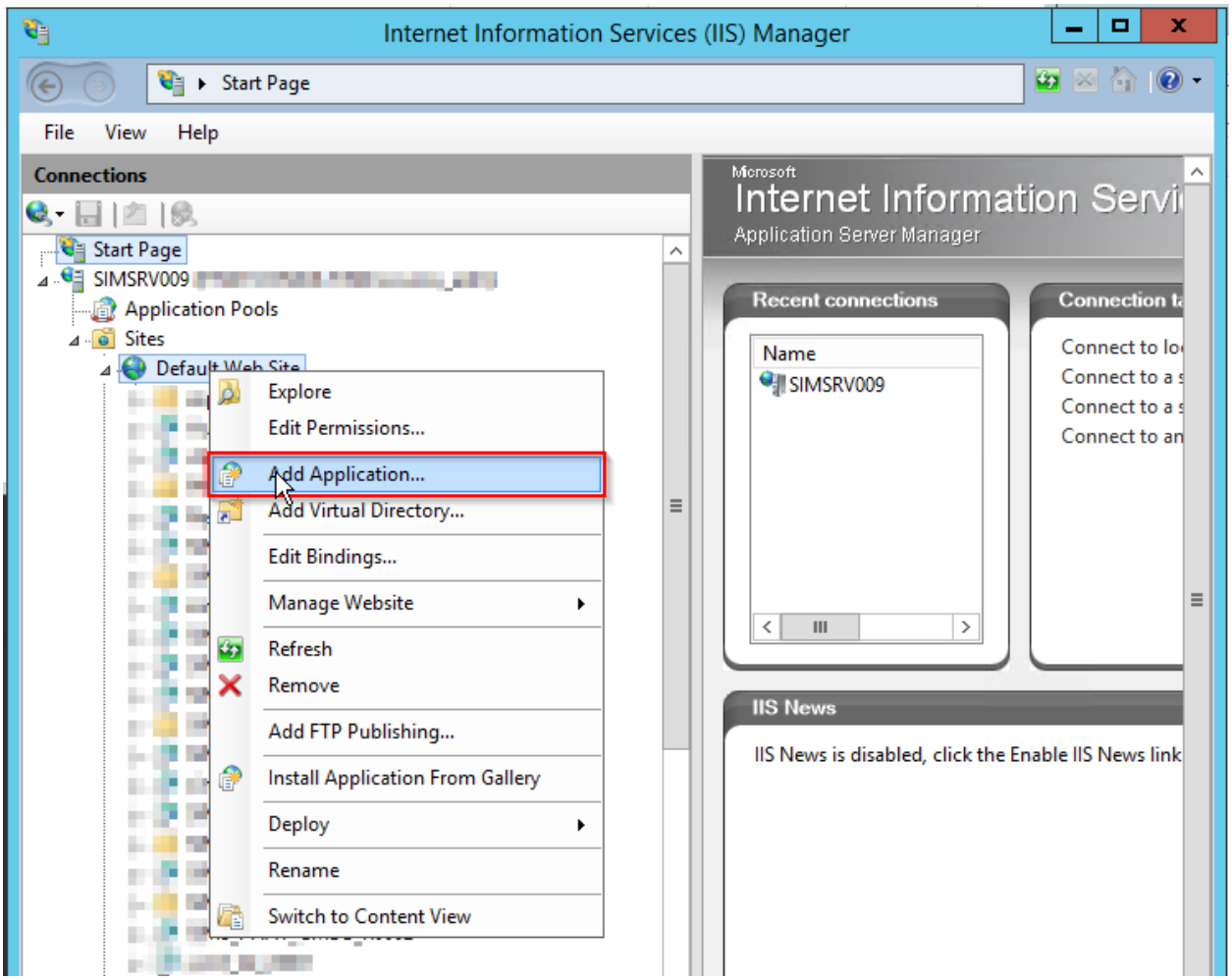
Create SilverMonkey folder

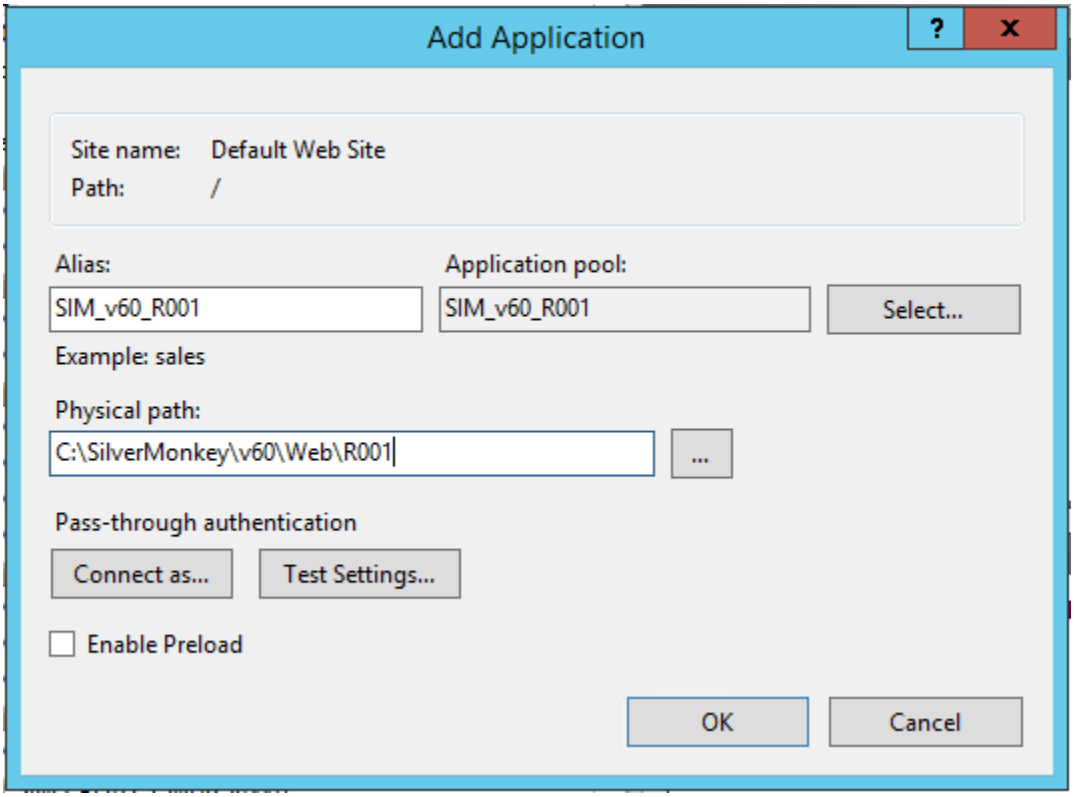
1. Create C:\SilverMonkey
2. Copy files from installation media to **C:\SilverMonkey\v60**

Create IIS Application

1. Add application (e.g. to DefaultWebSite), choose SIM AppPool (created above) and target to C:\SilverMonkey\v60\Web\R001.

Hint: The alias defines the later URL: http://HOSTNAME/ALIAS

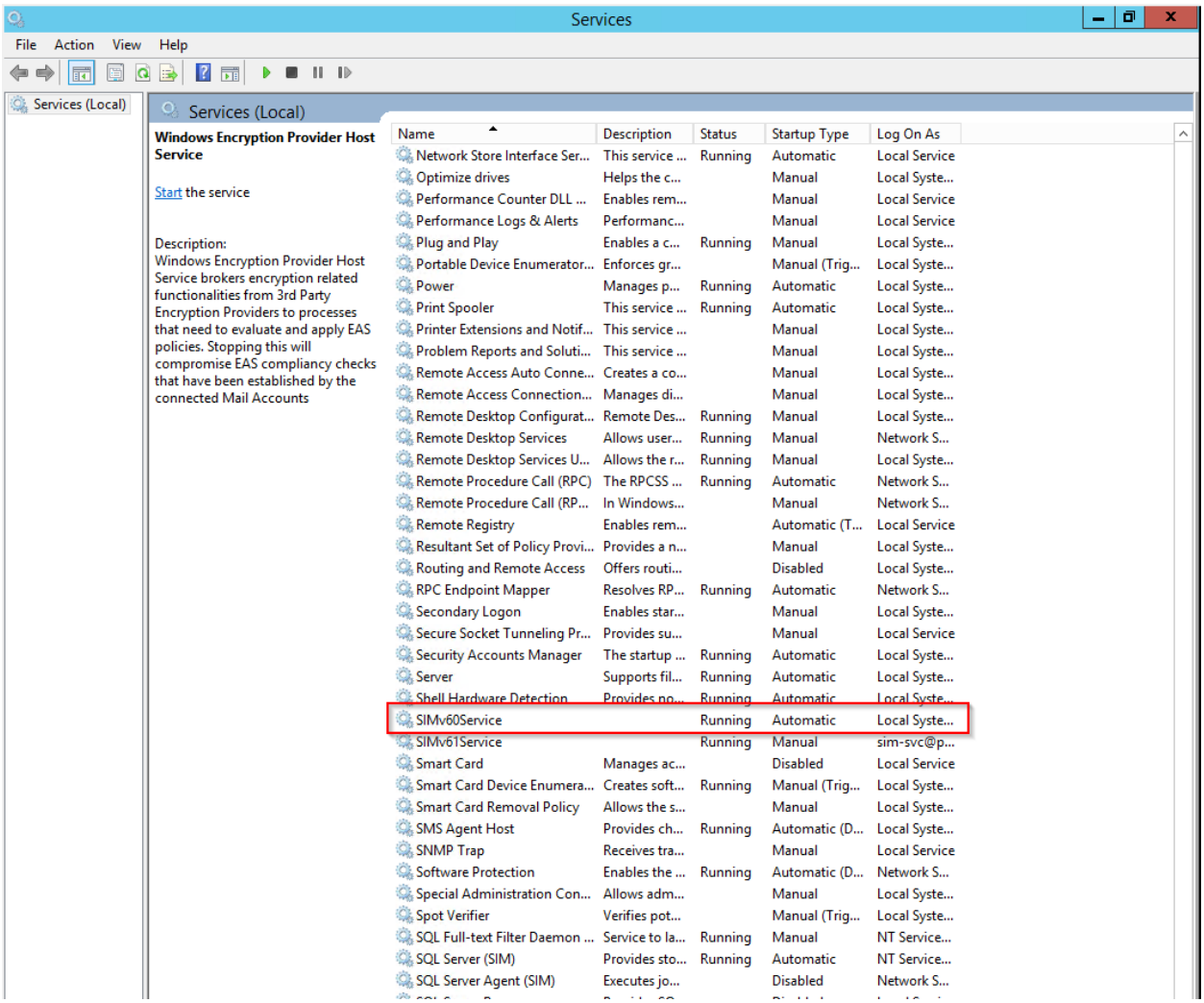

Internet Information Services (IIS) Manager
Start Page
File
View
Help
Connections
Start Page
SIMSRV009
Application Pools
Sites
Default Web Site
Explore
Edit Permissions...
Add Application...
Add Virtual Directory...
Edit Bindings...
Manage Website
Refresh
Remove
Add FTP Publishing...
Install Application From Gallery
Deploy
Rename
Switch to Content View
Microsoft
Internet Information Servi
Application Server Manager
Recent connections
Name
SIMSRV009
IIS News
IIS News is disabled, click the Enable IIS News link



2. Change value `databaseConnectionString` to SIM v60 DB in file **C:\SilverMonkey\v60\R001\Web.Config**

Install Windows Service

1. Go to C:\SilverMonkey\v60\WinService
2. Change value `databaseConnectionString` to SIM v60 DB in file **C:\SilverMonkey\v60\WinService\SilverMonkeyService.exe.config**
3. Execute **Install.cmd** with administrative rights
4. Open services.msc and make sure that the Windows Service **SIMv60Service** is installed



Test Installation

Manual

Modules:

Manual for module “Webservice”

In this article:

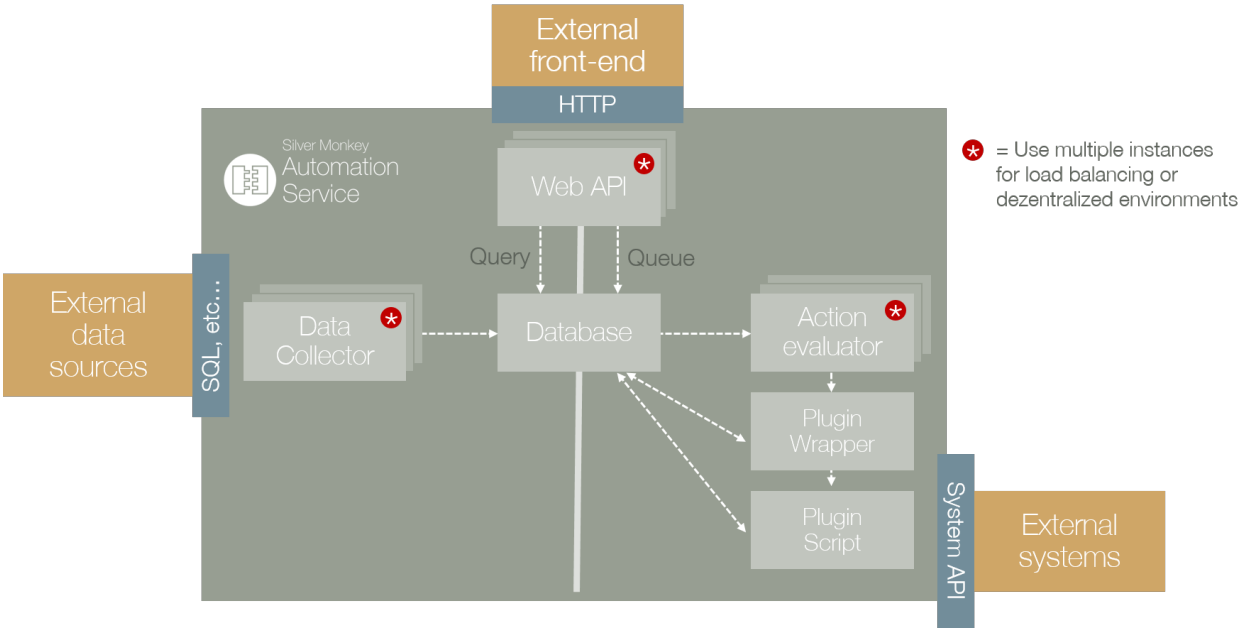
Warning: This article is under construction! Please DO NOT use any of the instructions below, yet! You may cause damage to your system. This article will be finished soon.

Authentication

Depending on the setting of the IIS application there are two possible authentication methods

1. Windows Authentication (recommended)
2. Authentication via firewall exception in IP base

Concept



The Webservice module consists of two main function: /queue and /query. Everything is accessable through a web api based on JSON format.

Queue For triggering and getting infos from (such as status) actions the /queue namespace have to be used.

All actions are created as planned actions in the SQL database table "queue". The "Action Evaluator" asks for planned actions. If an action is found, the Wrapper.ps1 is started with the information from `definition` XML and passes this data to the corresponding plugin PS1.

Query For retreiving dynamic data lists /query have to be used.

Queue

Adding a queue element for executing a powershell addon script

Creating Queue element via powershell

```
param(
[string]$definition,
[string]$url
)

Invoke-RestMethod -Uri "$url/api/queue?definition=$definition"
```

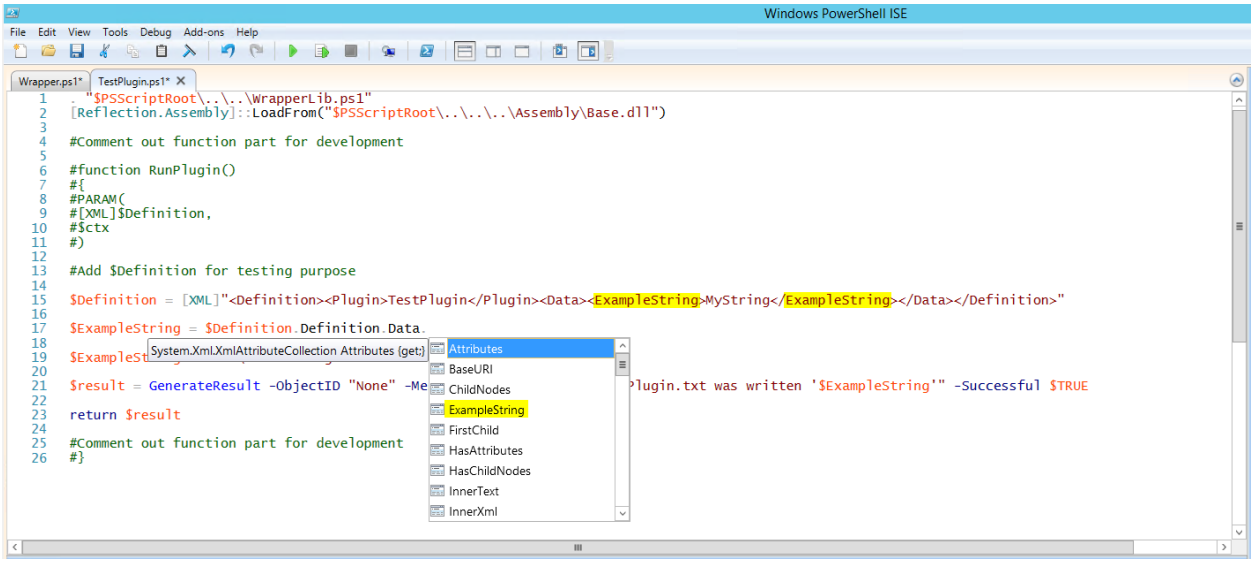
Creating a plugin

For creating plugins there are several rules:

1. Every plugin must consist of a main function (with specific parameters) which will be executed by the wrapper.ps1
2. Every plugin must return a specific class, which will be created by `GenerateResult`

```
. "$PSScriptRoot\..\..\WrapperLib.ps1"


function RunPlugin()
{
PARAM(
[XML]$Definition,
$ctx
)

    Try
    {

        #Place here general plugin functions


        #Generate a plugin result:
        $result = GenerateResult -ObjectID "None" -Message "Some return description␣
↪for queue result..." -Successful $TRUE

    }
    Catch
    {

        $ErrorMessage = $_.Exception.Message
        $result = GenerateResult -ObjectID "None" -Message "Unhandled exception␣
↪thrown while running plugin: $ErrorMessage" -Successful $FALSE

    }

    return $result

}
```

For development the wrapper behaviour can be simulated by commenting out the function part, and adding the XML string directly. Once the Powershell ISE run the XML variable declaration, the XML schema is avaiable through code completion:




```
. "$PSScriptRoot\..\..\WrapperLib.ps1"

#Comment out function part for development

#function RunPlugin()
#{
#PARAM(
#[XML]$Definition,
#$ctx
#)

#Add $Definition for testing purpose

$Definition = [XML]"<Definition><Plugin>TestPlugin</Plugin><Data><ExampleString>
↪MyString</ExampleString></Data></Definition>"

$ExampleString = $Definition.Definition.Data.ExampleString

$ExampleString >> "C:\SIMTestPlugin.txt"

$result = GenerateResult -ObjectID "None" -Message "To file C:\SIMTestPlugin.txt was␣
↪written '$ExampleString'" -Successful $TRUE

return $result

#Comment out function part for development
#}
```

Query

Getting information from the web service.

– Information will be handed in later –

Forms:

Manual for views

Note: Documentation is still in development process. Please do not hesitate to contact us on support@silvermonkey.net for further information.

This document is meant to be a source for all information regarding the administration and installation of Silver Monkey v6 Views.

In this article:

Views

Data

All data which is used on the page is inserted via sql-statement.

Form

HTML-Elements

Atributes

Atribute	Description
Type	Static. If no other type is defined, textbox is going to be used as standart
Data	SQL Queries used to generate data for this particular configuration
ID	ID of the controlling element. If no ID is set while configuring, one will automatically be generated
Title	Name of the view
Format	Design options to format the views, typo, color, fontsize, etc
Class	
Style	css sheet for general style options
Watermark	digitally watermark against copyright infringement

Button

Button are triggers in order to execute certain scripts or commands with interaction from the user. The table below shows different configuration elements.

Atribute	Description
ID	Unique Name used for referencing
Function	

Lists

Display entrys of another class (e.g. table). It is possible to use placeholders for attributes in this statement.

Atribute	Description
ID	Unique name, used for referencing and identifying.
Command	

An example of how to use the different attributes.

```
<List Id="List" Command="LoadFrame('EditItem', 'View.aspx?ViewId=60104&amp;Id='+strId,
↪ strDirection);" Add="LoadFrame('EditItem', 'View.aspx?ViewId=60104&amp;Id=-2',
↪'down');">
          <ListItem>
            <div class="Content">
              <div class="Title">{DisplayName}</div>
              <div class="Text">-</div>
            </div>
          </ListItem>
        </List>
```

WorkflowControl

ButtonMore

Is used to initiate furher actions for this button. For example postPushButton scripts as seem in the example below.

WorkflowChart

Visualizes workflows through bpmn or epk diagramm.

Insert jpeg here of visualized workflow.

Explanation how it works to implement one of these workflows

Chart

A visualization of numbers in a diagram. Used mostly in dashboards to show peaks of downloads or orders in this tool.

insert jpeg here

insert how it works below

Data comes from an sql?

Search

A function to iterate through the whole database comparing the search item with it. Can be implemented through a search bar or used in the configuration as seen below.

Rating

An attribute used for items such as hardware or softwware rated by the users who ordered them in self service. Based on this rating filtering and sorting items in different views. The shop area for self service is an example for. every item has an additional field for a rating from 0-5.

Insert shop picture unsorted and sorted.

Tile

Tiles are design elements for webdesign. A tile contains branding and color is easy to reproduce. Different color- and fontsets can be used design a constant look for the website.

NavTree

A NavTree uses an already existing table to build a navigation element out of it. Every column represents the first level of navigation and it contains all elements as a second level navigation in that column.

```
<NavTree Table="ShopCategory" OrderBy="SortOrder" Filter="" SelectedId="{?CategoryId}
↪" Script="document.location.href='View.aspx?ViewId=20030&amp;NoHeaderAnimation=1&
↪amp;CategoryId={Id}';" />
```

Repeater

A funcion to repeat certain commands. Refreshing a list in a certain view for example.

```
<Repeater Id="Products" Source="Products" PageSize="16">
         <div class="ShopProduct" Style="cursor:default; color:#404040;">
           <div class="ShopProductImage" Style="opacity:1;">
             <img src="{EnsureFile:.\Cache\Table\{GUID}_Image1.png;Image1;./Resource/
↪Images/MissingProductImage.png}" />
           </div>
           <div class="ShopProductTitle">
             <div class="Title">{DisplayName}</div>
             <Span>{ItemName}</Span>
           </div>
         </div>
       </Repeater>
```

DataTable

A DataSet is made up of a collection of tables, relationships, and constraints. In ADO.NET, DataTable objects are used to represent the tables in a DataSet. A DataTable represents one table of in-memory relational data; the data is local to the .NET-based application in which it resides, but can be populated from a data source such as Microsoft SQL Server using a DataAdapter. The DataTable class is a member of the System.Data namespace within the .NET Framework class library. You can create and use a DataTable independently or as a member of a DataSet, and DataTable objects can also be used in conjunction with other .NET Framework objects, including the DataView. You access the collection of tables in a DataSet through the Tables property of the DataSet object. The schema, or structure of a table is represented by columns and constraints. You define the schema of a DataTable using DataColumn objects as well as ForeignKeyConstraint and UniqueConstraint objects. The columns in a table can map to columns in a data source, contain calculated values from expressions, automatically increment their values, or contain primary key values. In addition to a schema, a DataTable must also have rows to contain and order data. The DataRow class represents the actual data contained in a table. You use the DataRow and its properties and methods to retrieve, evaluate, and manipulate the data in a table. As you access and change the data within a row, the DataRow object maintains both its current and original state. You can create parent-child relationships between tables using one or more related columns in the tables. You create a relationship between DataTable objects using a DataRelation. DataRelation objects can then be used to return the related child or parent rows of a particular row. For more information, see Hinzufügen von 'DataRelations'.

List

Represents a strongly typed list of objects that can be accessed by index. Provides methods to search, sort, and manipulate lists.

Splitter

Represents a splitter control that enables the user to resize docked controls.

Functionbar

The bar simply lists all the function definitions inside the file. The pattern matching used to generate the function list.

Frame

A frame is used to build a website, to make it more navigateable. Certain elements of the website are put into single frames to make resizing more manageable.

Header

Menubar to navigate trough a certain page/view. It is built up like a navigation-element.

Script

Scripts are interpreted programms to automate processes. They can be implemented through a variety of triggers and actions.

Style

A style is used to implement a general configuration of style elements like color, font, fontsize and branding.

General configuration

How to generally configurate your own views is descriped here. The items are linked to the configuration items in order to give further explanation. ..Link every item to its own site

..code-block:: <View Icon="**Place designated icon here**">

<Name Lang="DE">**Name of the view used for referencing**</Name> <Data>

SQL-Query to get the needed data

</Data> <Form> **specify form of the view here**

<Header> <HeaderMenuItem Title="**First Level menu item**" Link="**Link to the specified view**" /> <HeaderMenuMore Title="**First Level menu item**">

<HeaderMenuItem Title="**Second level menu item**" Link="**Link to the specified view**" />

</HeaderMenuMore>

</Header> <Splitter>

<Left>

<List Id="List" Command="LoadFrame('EditItem', 'View.aspx?ViewId=40050&Id='+strId, strDirect

<ListItem>

** ** {DisplayName} {Count} Installationen

</ListItem>

</List>

</Left> <Right>

```
                <Frame Id="EditItem" />
            </Right>
        </Splitter>
    </Form>
</View>
```

Changelog

Version	TicketId	Product	Description
6.0.0	None	Initial Version	

Supported configurations

Supported Microsoft SQL Server Versions

Product	Version	Supported
SQL Server 2012	11.0	Yes
SQL Server 2014	12.0	Yes
SQL Server 2016	13.0	Yes

Supported Microsoft Windows Server Versions

Product	Version	Supported
Windows Server 2012	NT 6.2	Yes
Windows Server 2012 R2	NT 6.3	Yes
Windows Server 2016	NT 10.0	Yes

Supported .Net Framework Versions

- Hence the code was written in .Net Core 1.0 only this version is supported

Support

If you have further questions regarding out products or the documentation contact us:

- Tel. : +49 40 - 226 383 160
- E-Mail : Support@SilverMonkey.net

If you need general Information about our Products visit: http://www.SilverMonkey.net