TELIA ACE

INSTALLATION MANUAL

ACE CLIENTS

Note that the information in this document is subject to change.

Contents

1 Introduction

This function design describes how the installation programs for ACE clients works at installation, upgrade and uninstallation.

1.1 Terms and abbreviations

Term	Description
Client	Refers to the ACE Admin, ACE Agent, ACE Pulse, and ACE Report client applications.
ACE Configuration Utility	Configuration tool that can be installed with a client.
ACE Connector for Skype	Application used when Lync or Skype for Business are used to handle calls in ACE
ACE Interact	Web based agent client for handling contacts in ACE
Registered protocol handler	Allows web applications to send data to desktop applications
TeliaEdgeAgentPop	A registered protocol handler that allows ACE Interact to communicate with ACE Screenpop
ACEConnectorForSkype	A registered protocol handler that allows ACE Interact to communicate with ACE Connector for Skype
InstallShield (IS)	Package tool. Packs files and creates a self-extracting program that will distribute the files according to the configuration.
MSI	Microsoft Installer file.
UAC	User Account Control – a Microsoft security infrastructure

1.2 Appendixes

–

1.3 References

Ref [1]   Telia ACE Installed Components

Ref [2]   Operation Manual Telia ACE

Ref [3]   Microsoft Developer Network – http://msdn.microsoft.com

Ref [4]   Telia ACE Interface Specification Screen Pop Interface

1.4 Revisions

Rev	Date	Signature	Comment
1.0	2018-06-21	MaTh	Approved for ACE 12.0 - Rebranded for ACE
2.0	2018-10-23	MaTh	Approved for ACE 13.0 - Updated with installation info for ACE Connector for Skype for use with ACE Interact

2 MSI installation

Note: Special instructions apply to the installation of ACE Report.
See section 3.

You can install an ACE client in two ways:

1. Run the MSI installation program in its basic design. Customer unique configurations are made afterwards using Configuration Utility.

2. An MSI transform is constructed based on the original MSI installation program, including all configurations. In order to have all the configurations transferred to an MSI transform, you configure all relevant and customer unique information in Configuration Utility and then export what you have configured to a registry file, that you can later attach when constructing an MSI transform. Which keys that shall be exported per client is described in *Telia ACE Installed Components*.

You find an example of registry files in section 0.

2.1 Run MSI installation

By right-clicking on the MSI file for the client and selecting **Properties**, you can see which version of the client that will be installed.
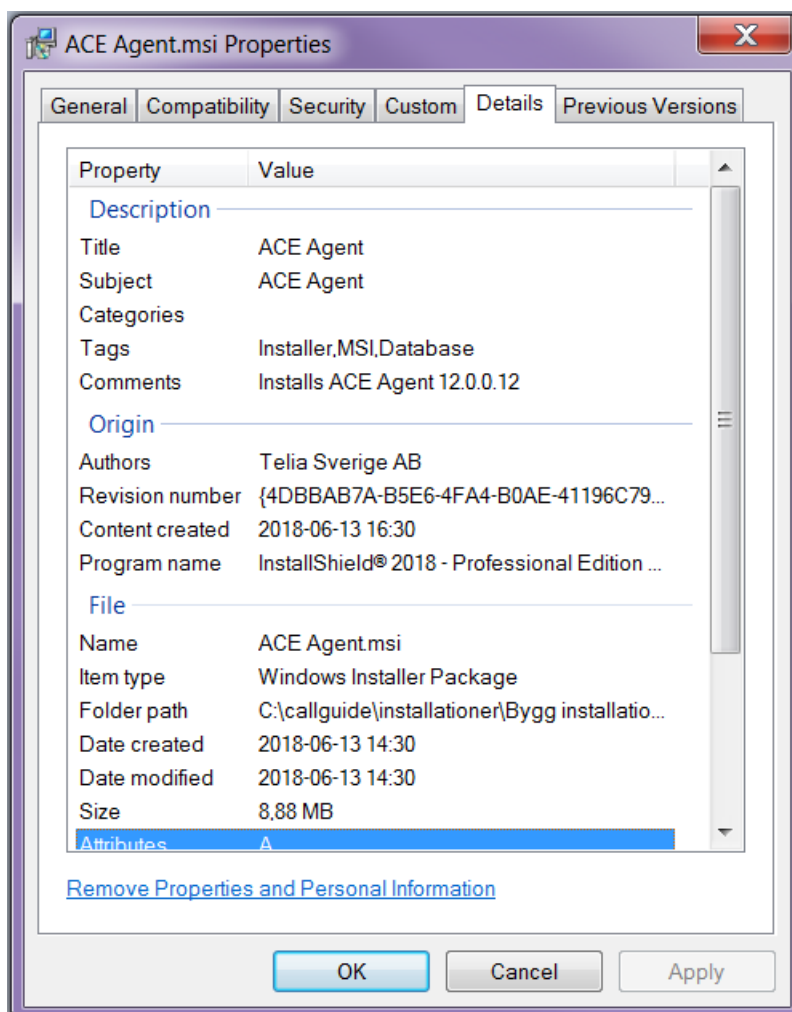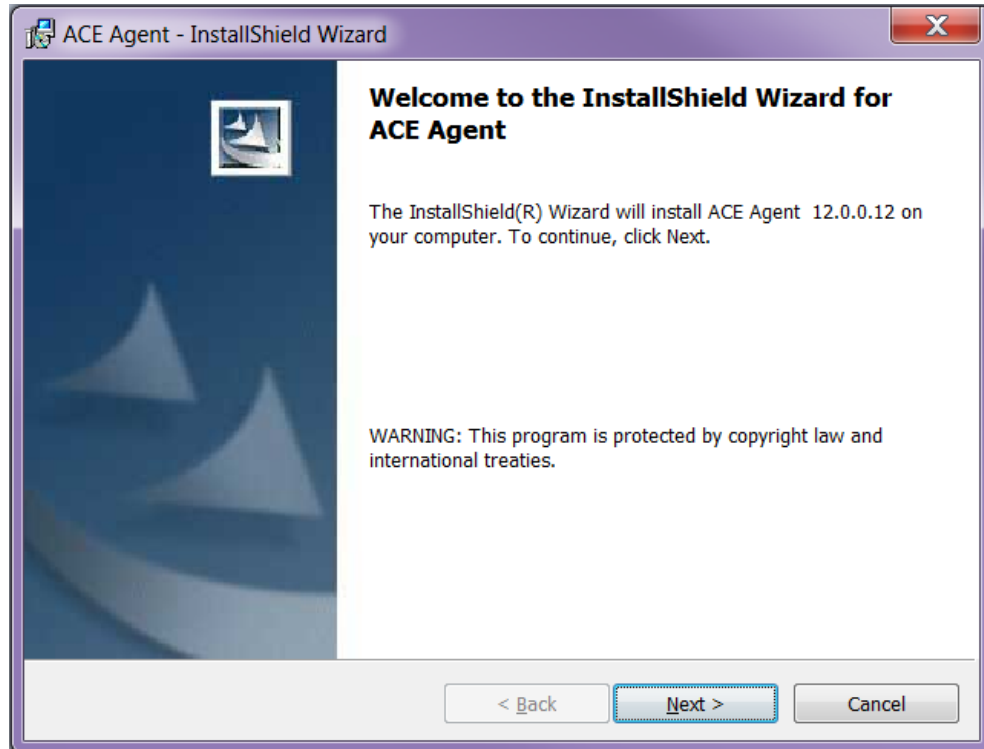

In the release notes for each respective client and version you find a description of which versions of dll:s, if any, that come along in the installation.
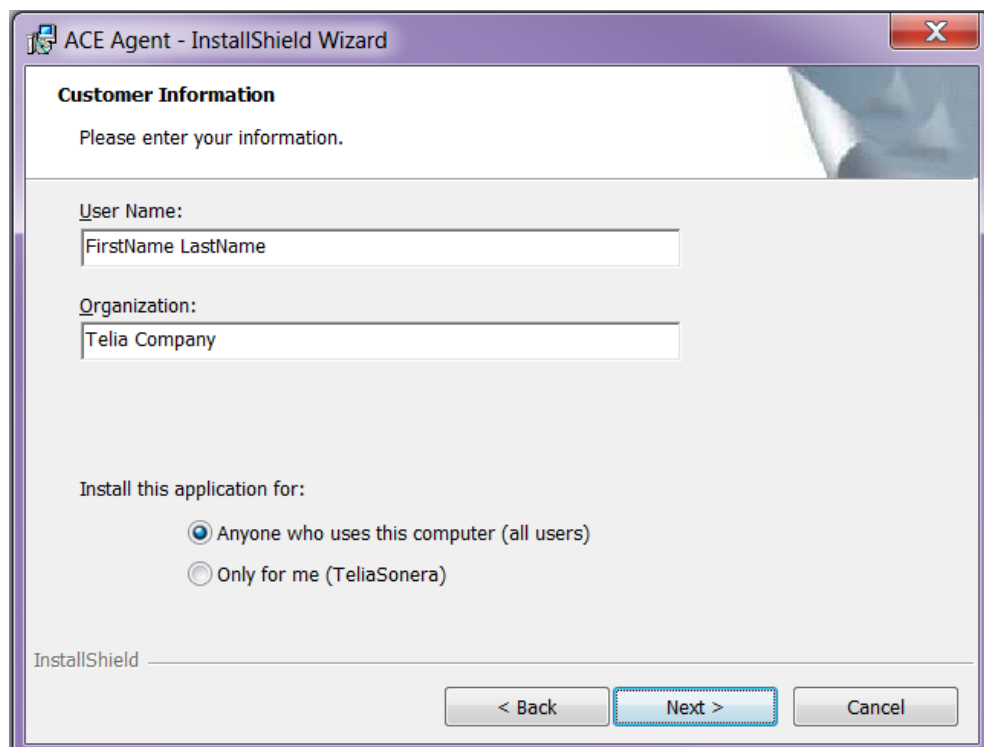
You run the installation program (in English) by double-clicking on the MSI file created for the installation.
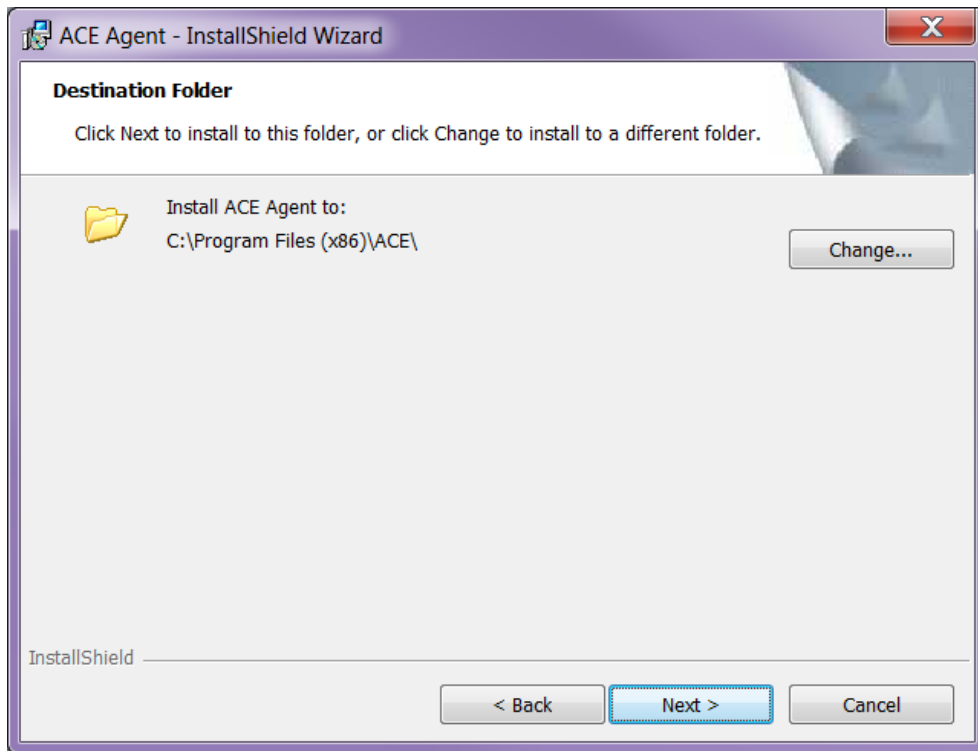
2.1.1 New installation

This is where you see which version of the client that will be installed.

Select whether the client shall be installed for all using this computer or just for the one installing.

As default the client is installed in Windows program folder. However, you have the option of changing to any folder.

If you choose "Complete" setup, the client is installed with all available languages, ACE Configuration Utility and a component for single sign-on. The latter is available provided that you have .NET Framework 4.5 (or newer) installed.

If you choose "Custom" setup, you can choose what to install, you can e.g. skip languages that will not be needed. For further information about settings for which language to be used by the application, see *Operation Manual Telia ACE*
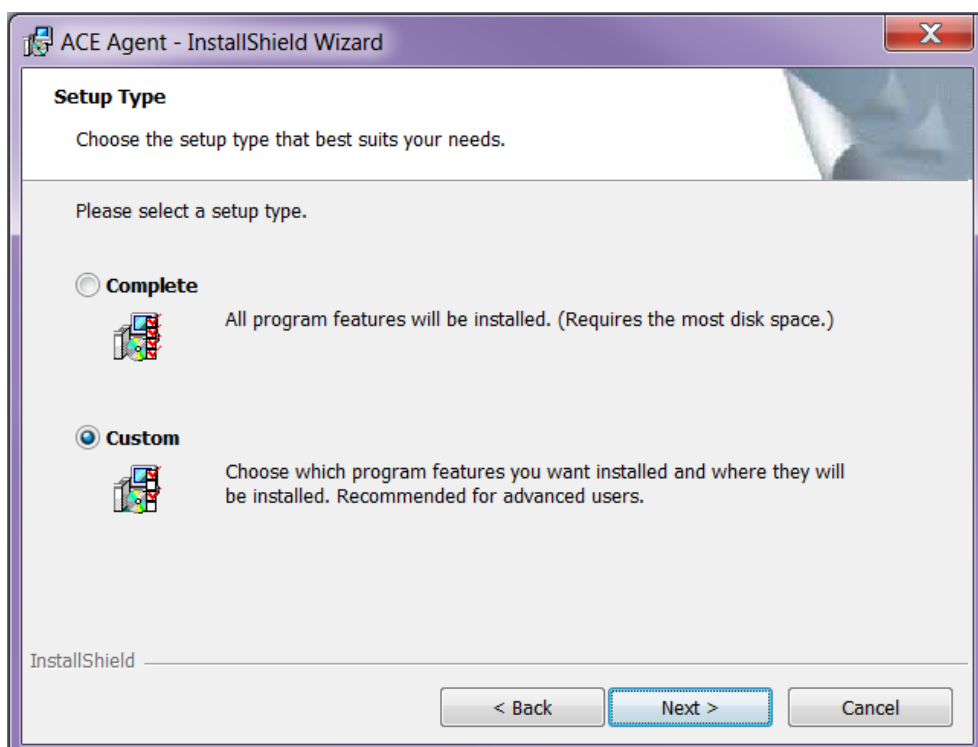
When a component has been selected, you see a description together with information about the disk space required for this component.
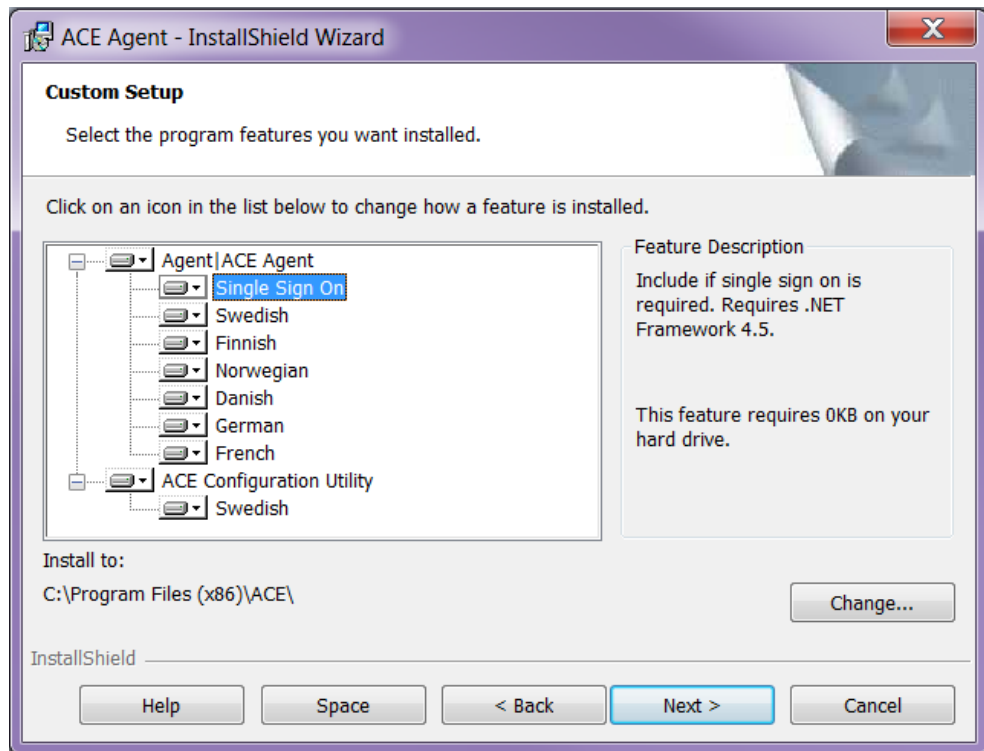
By clicking on the arrow next to a component, you see a menu where you can choose not to install the component, unless it is a mandatory component.
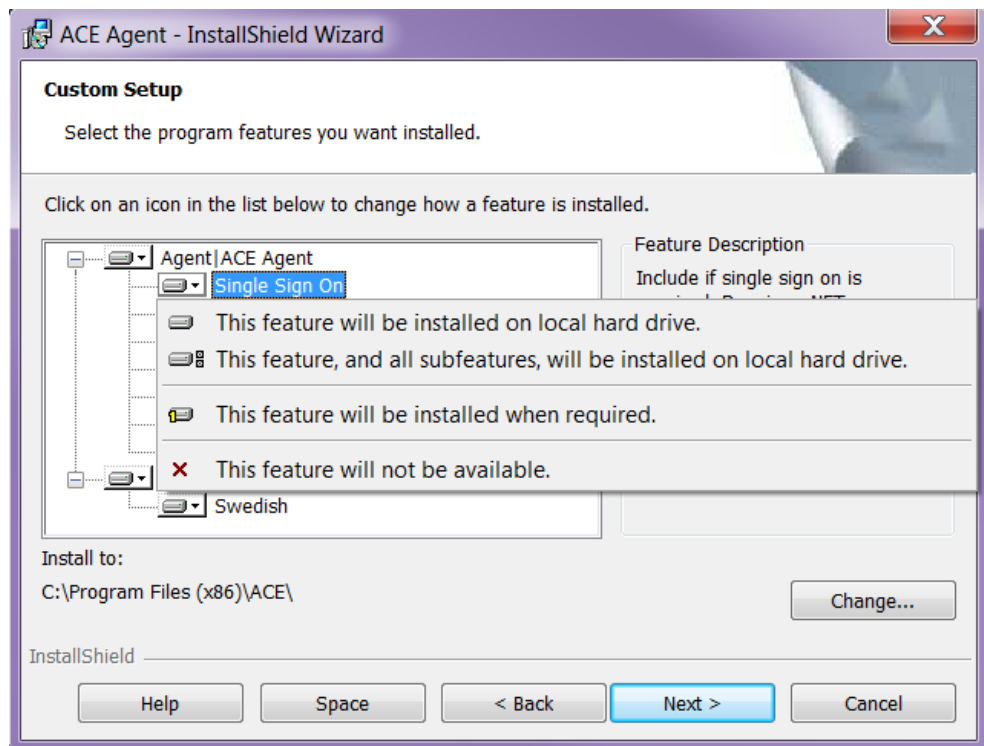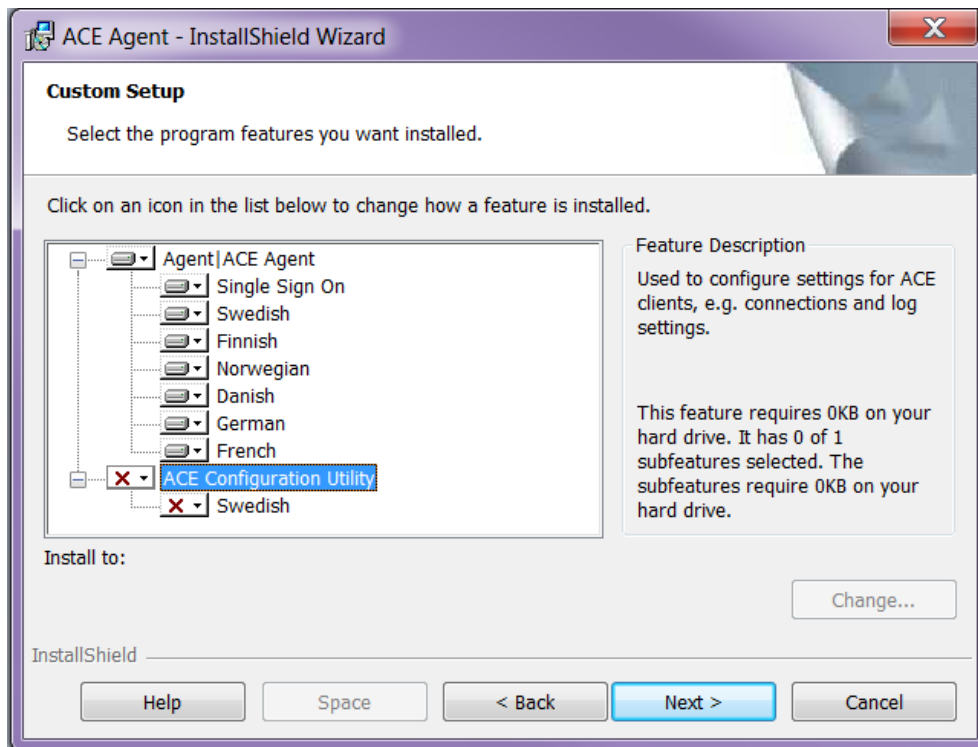
When all settings are complete, all files required for each respective component are installed.
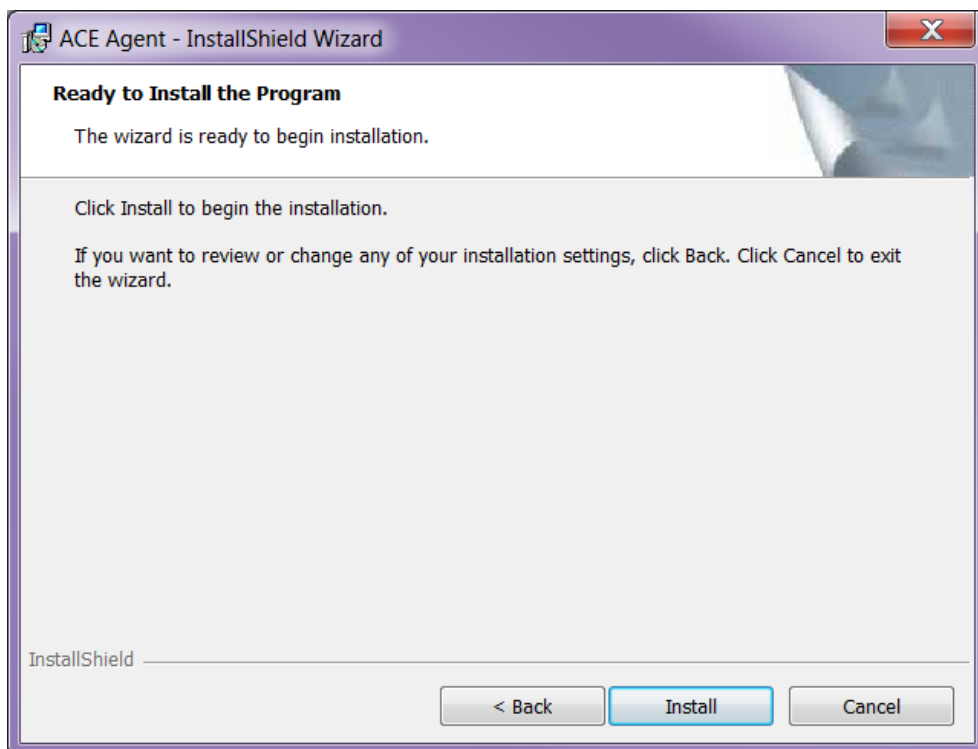
Clicking **Install** will bring up the progress bar window and probably a User Account Control confirmation window as well, depending on computer settings.

When the installation is finished, configurations must be made in ACE Configuration Utility or directly in Windows Registry to be able to run the client.

2.1.2 Modify, repair, uninstall

By running the same installation file (.msi) once more you have the option of

- modifying the existing installation, e.g. add a language you skipped at the original installation,

- repairing a corrupt installation,
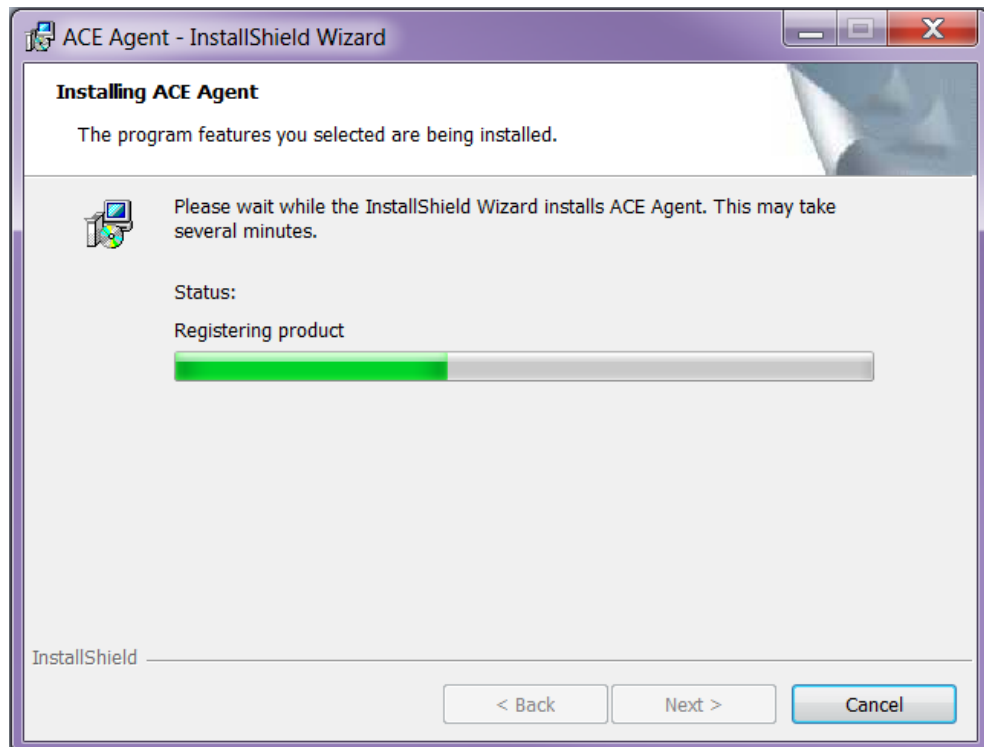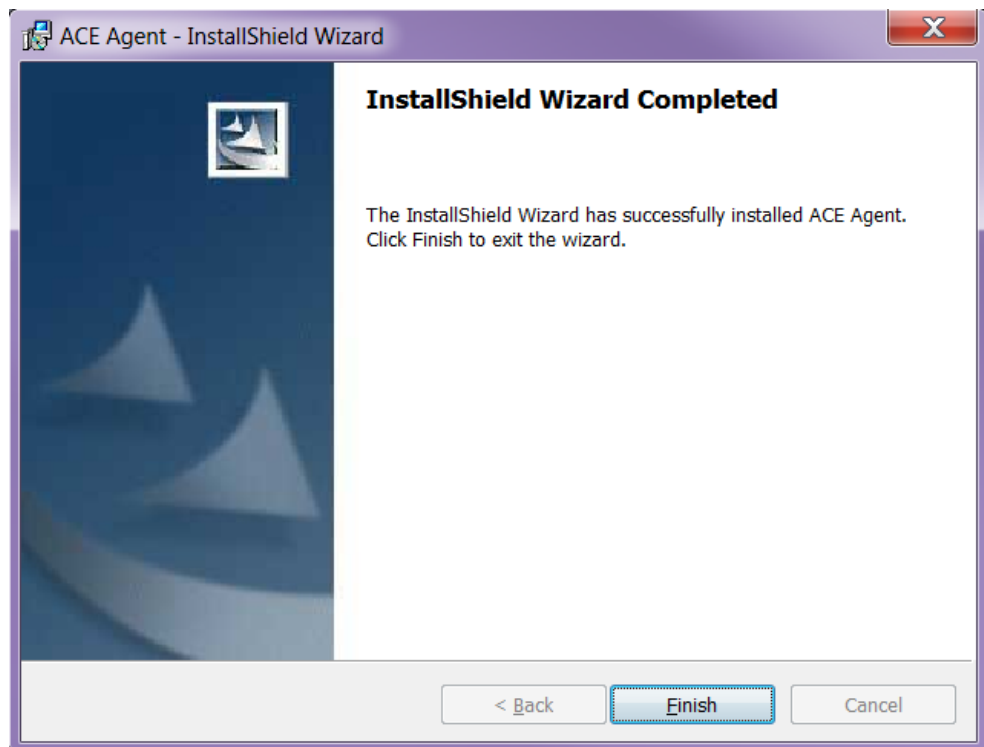
- uninstalling the client. Note that files that have been installed by the installation program and that are shared by other programs will not be uninstalled in case they are still required by the other programs.

2.1.3 Upgrade

A new version of a client comes in a new .msi file. If you double-click on this file, the previous installation will be removed automatically during the install.

2.2 MSI transform

A transform is in itself an .msi file, replacing an existing msi installation. In its simplest form an msi installation is an .msi file, e.g. ACE Agent.msi. When you create a transform from ACE Agent.msi, you can replace ACE Agent.msi with the transform file, which is e.g. named SpecialAgent.msi.

Via the transform you can add unique configurations, for example, the transform can create extra registry keys and copy extra files to a specified search path.

2.2.1   Example - Building an MSI transform

In this section you find an example of how you, in InstallShield 2018 Spring, create an MSI transform that modifies an existing installation file of the .msi type.

1. In InstallShield, select **File/New**…
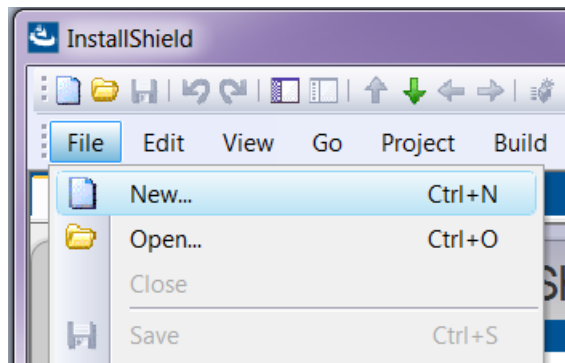
2. In the opened dialog, select the Windows Installer tab and select **Transform**. Give the transform a name and location on the computer for the transform.
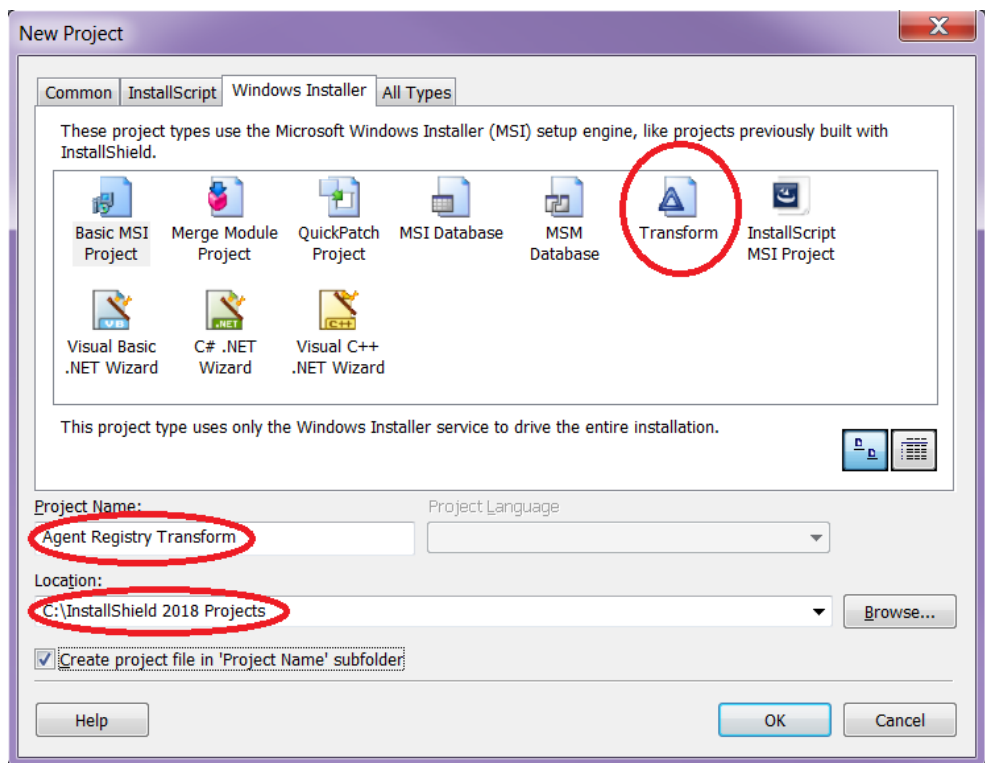
3. A dialog sequence is initiated. Click on **Next** in the first pane.
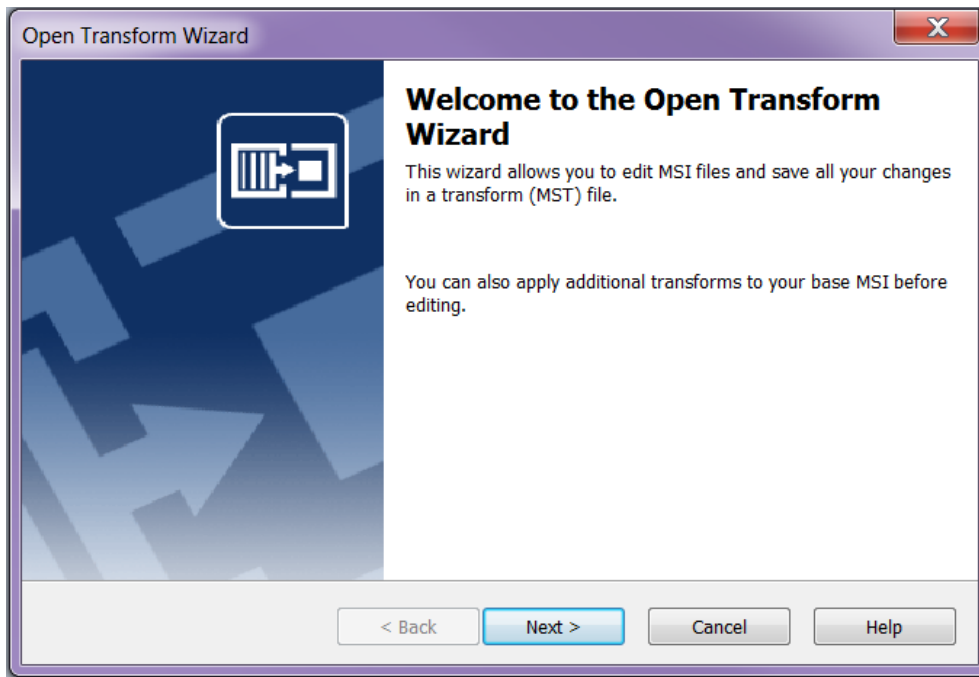
4. In the next pane you have to specify which MSI installation you want to transform. Pick an earlier created .msi file. Click on **Next**.
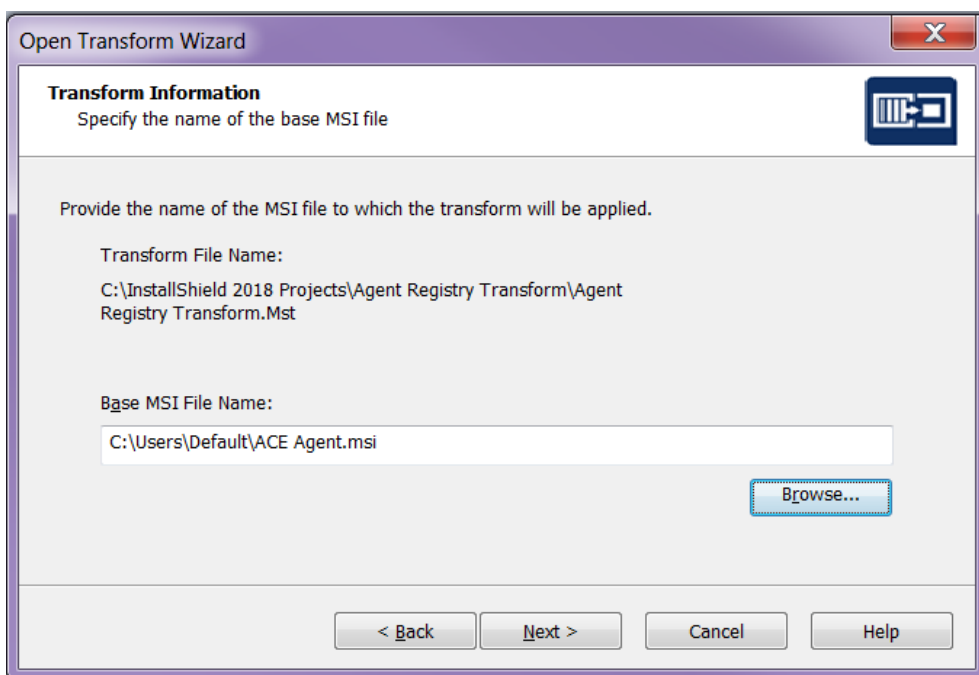
5. In the following pane you must specify any further transforms that shall apply. Click on **Next**.
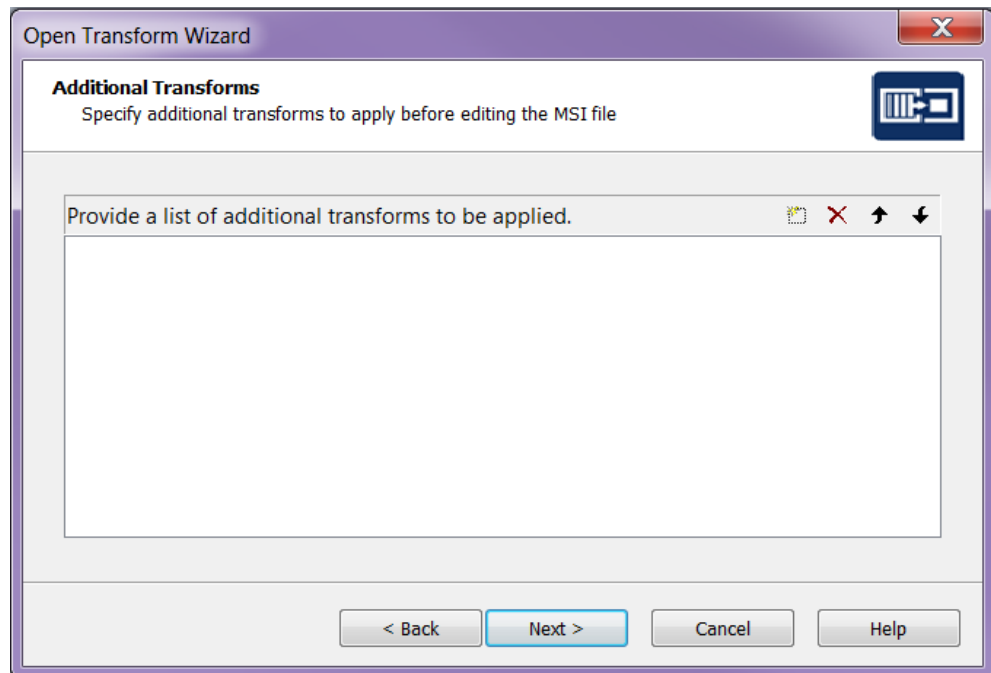
6. In the next pane you can specify if you want to create a so-called Response Transform. If you do so, a simulation of the installation is run. All the values you specify in the dialogs in this simulation will be saved and these values will be specified when you run the transform. You can also specify command line parameters to be sent to the .msi file when it is run.
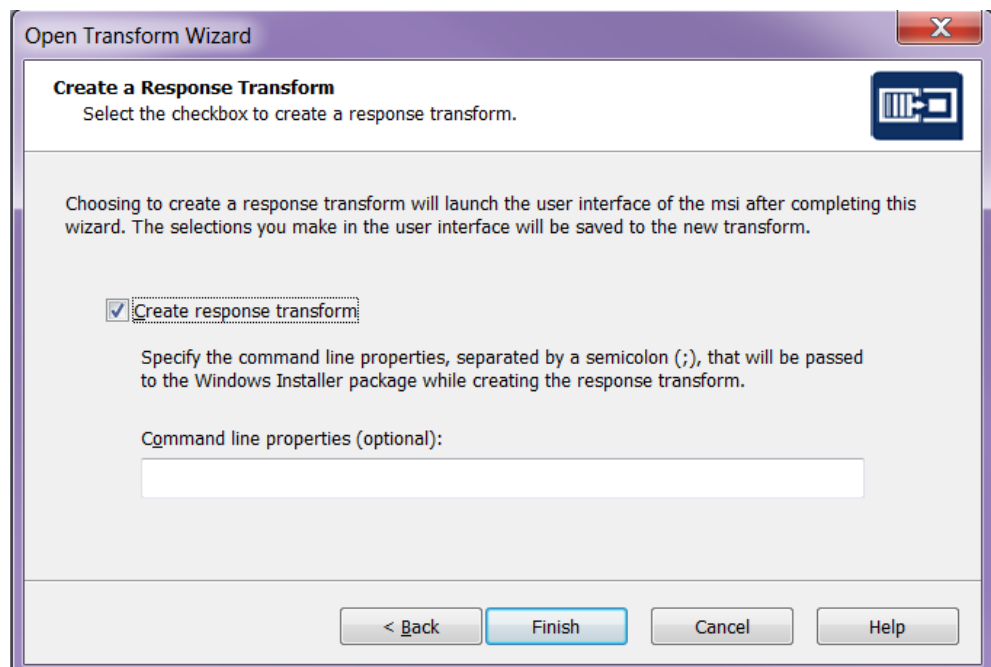
7. Click on **Finish** (when the simulated installation is done). A transform is now created. Below you see Project Assistant:
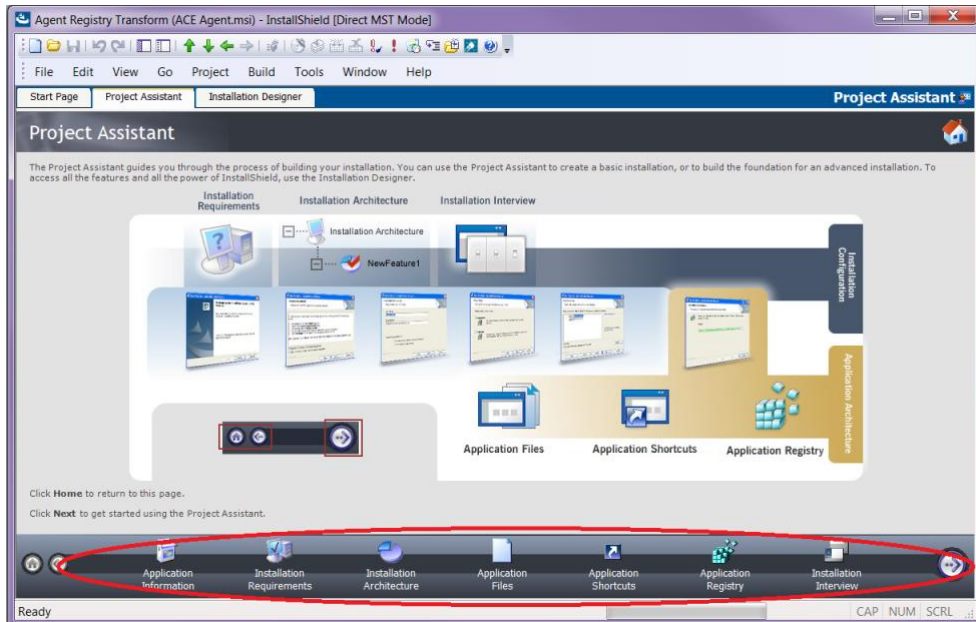
8. In Project Assistant a number of properties for the transform can be specified. During the Installation Architecture you specify the parts of the original installation that shall be included in the transform.

9. In case you want to modify what is to be installed, you tick **Yes** under "Do you want to customize your Installation Architecture?". If you e.g. do not want to install ACE Configuration Utility, you select the corresponding feature and click on the **Delete** button.

- Installation Manual ACE Clients - 2018-10-23
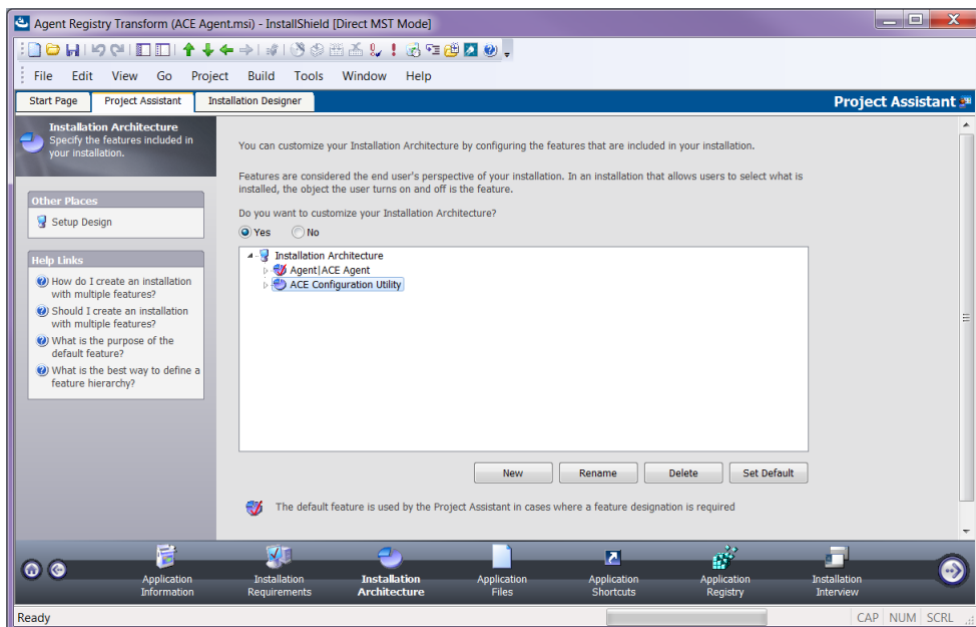
10. Under **Application Files** you specify the files to be included in the transform. Here, as well, we are going to remove ACE Configuration Utility (unless it shall be installed). You do this by right-clicking on the folder with the files and selecting **Delete**:
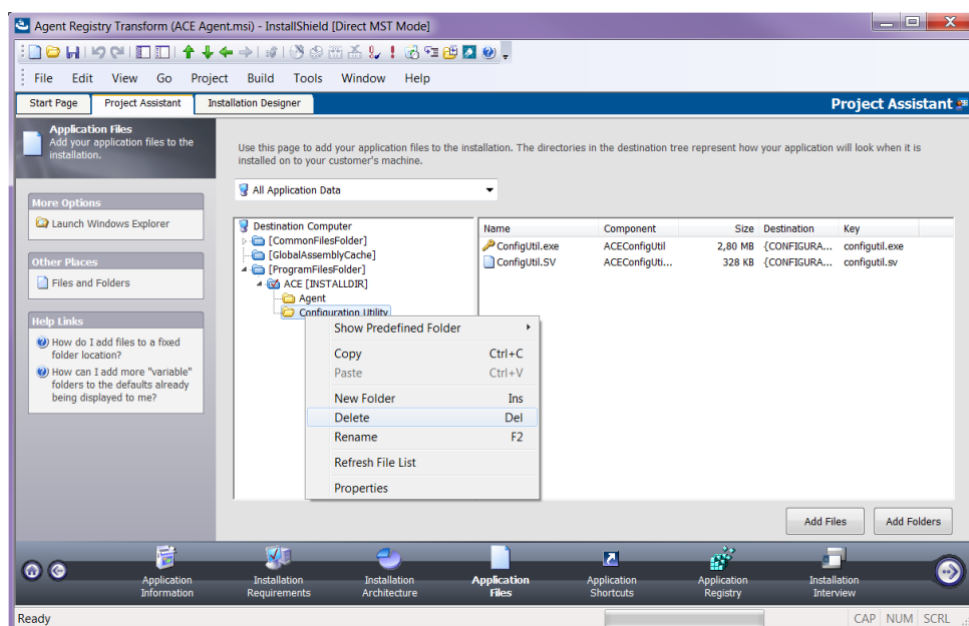
11. To add a new file, you use the **Add Files** button. A dialog is opened, where you can point to a file:
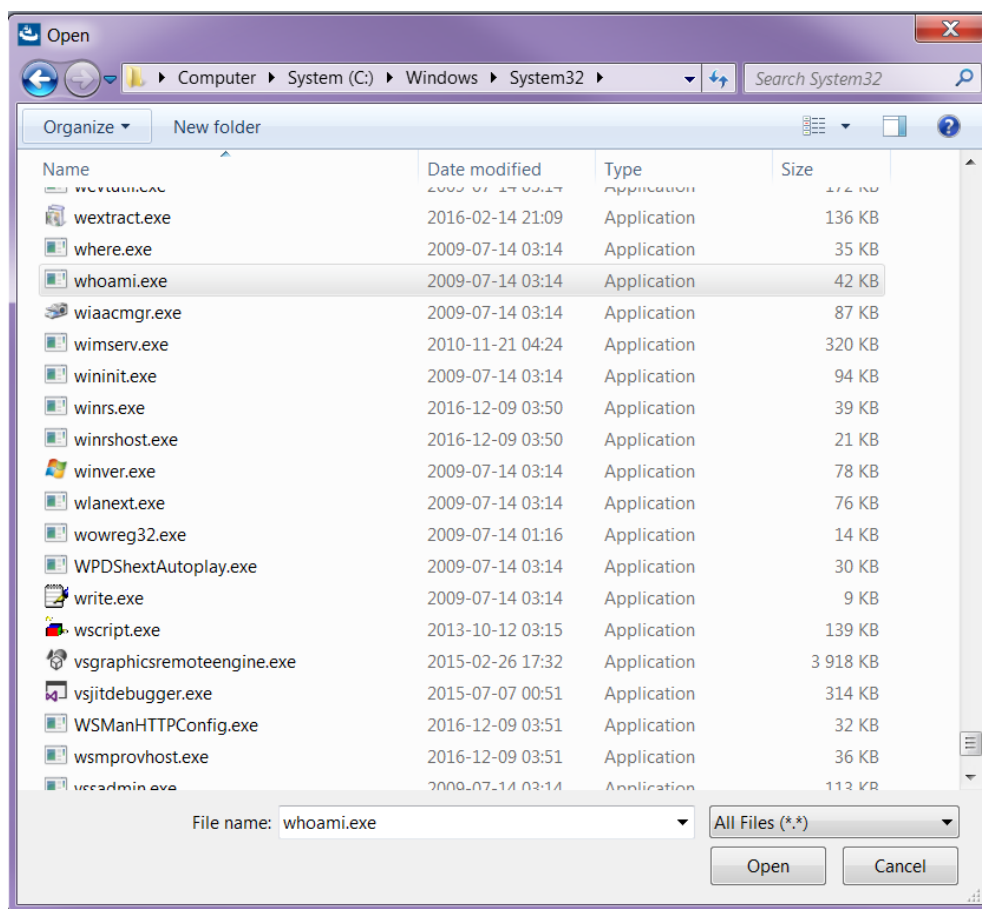
12. When you have selected the file you must choose the way to send it along in the transform. Specify "In a new CAB file" and tick "Stream the new CAB file into the Windows Installer package" – the new file will be incorporated in the transform file.

Note**:** It seems that this choice only available after having created the msi package once, i.e., if this checkbox is grey, try executing steps 19+20. Then you should be able to tick the choice.
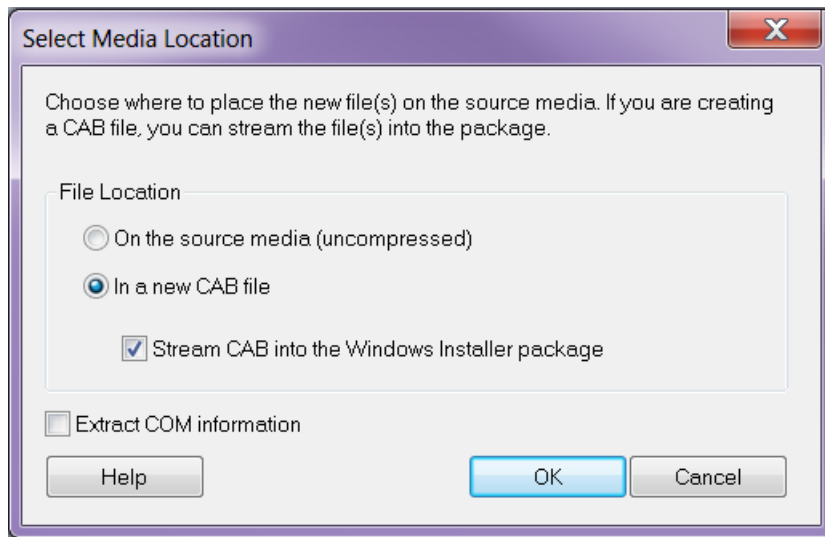
13. Under the Application Registry title, you can add registry keys that shall be created by the transform:
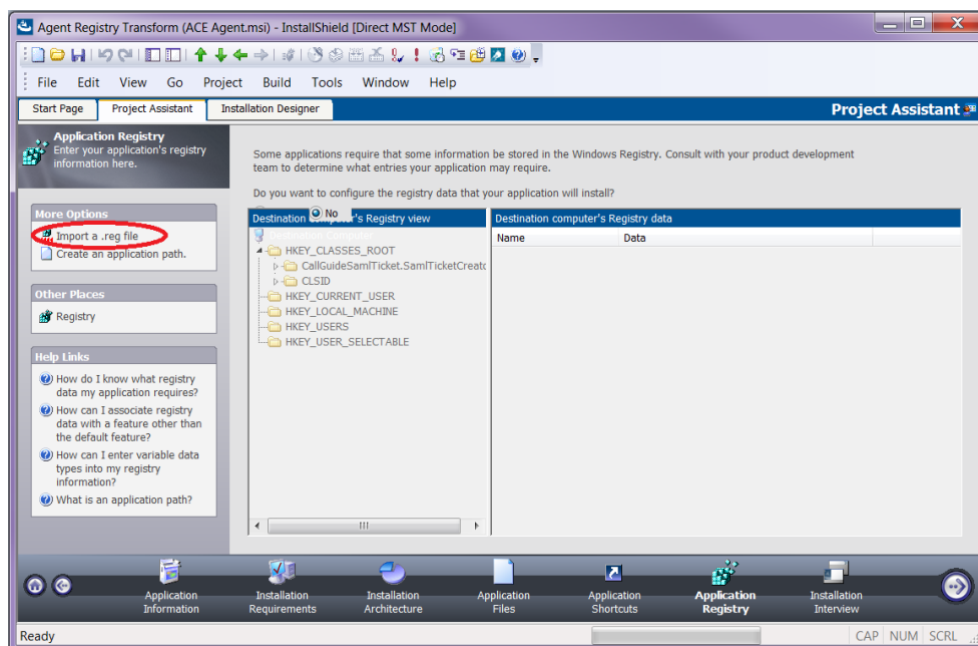
14. To record a file with registry entries (.reg), click on "Import a reg file". Then a dialog sequence is started, where you are allowed to point out a registry file. In the first dialog, select **Browse** and point at the file to be imported. Then click on **Next**:
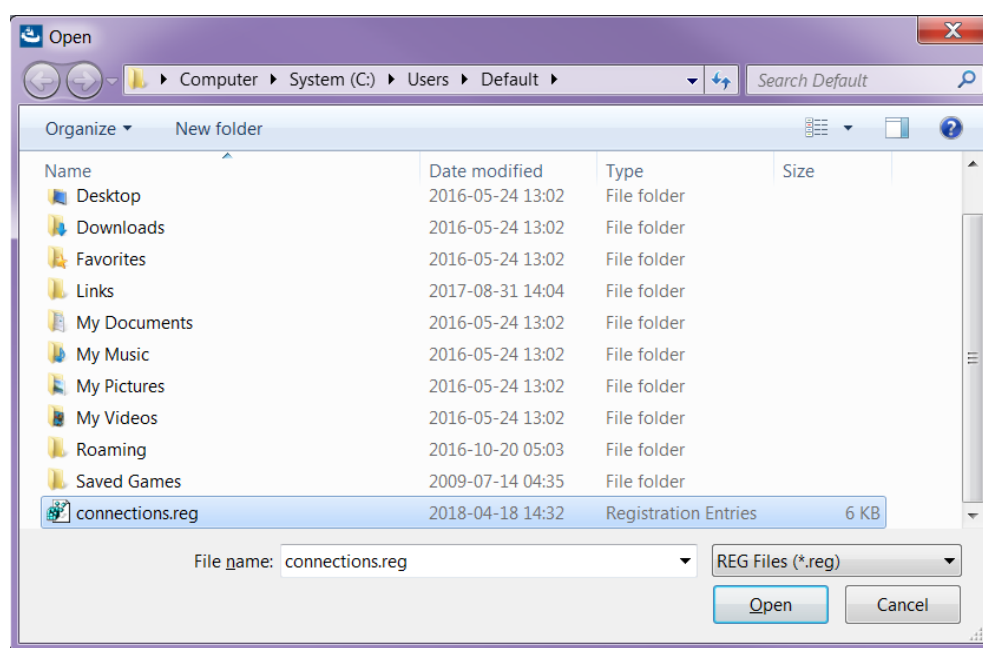
15. In the next dialog you specify whether the imported registry keys shall be allowed to replace already existing keys:
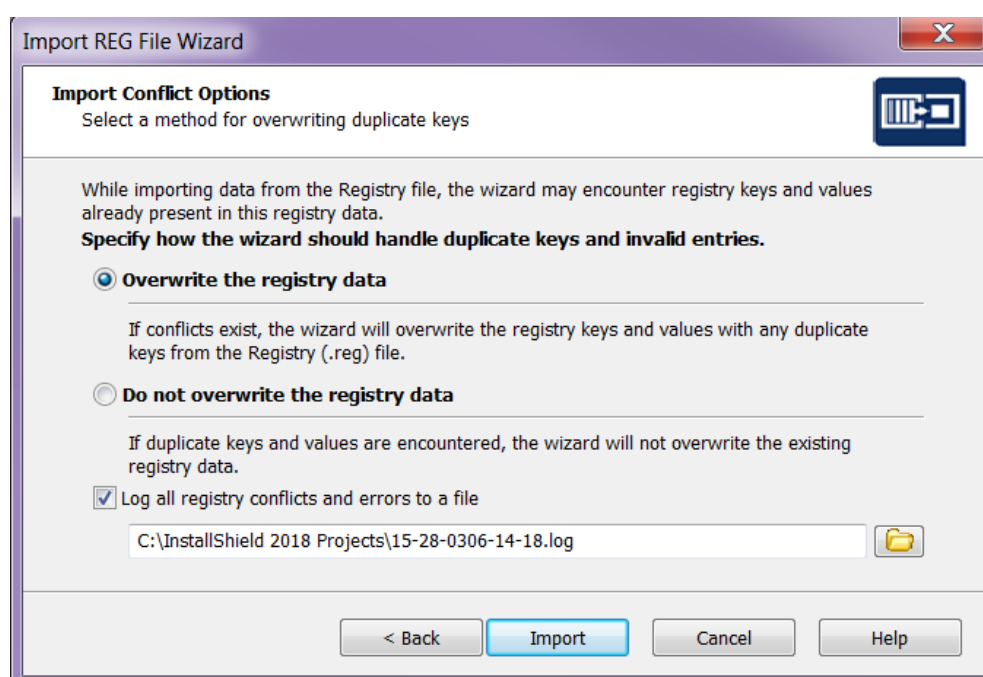
16. Click on the **Import** button. The registry file is imported. In the next dialog, click on **Finish**:
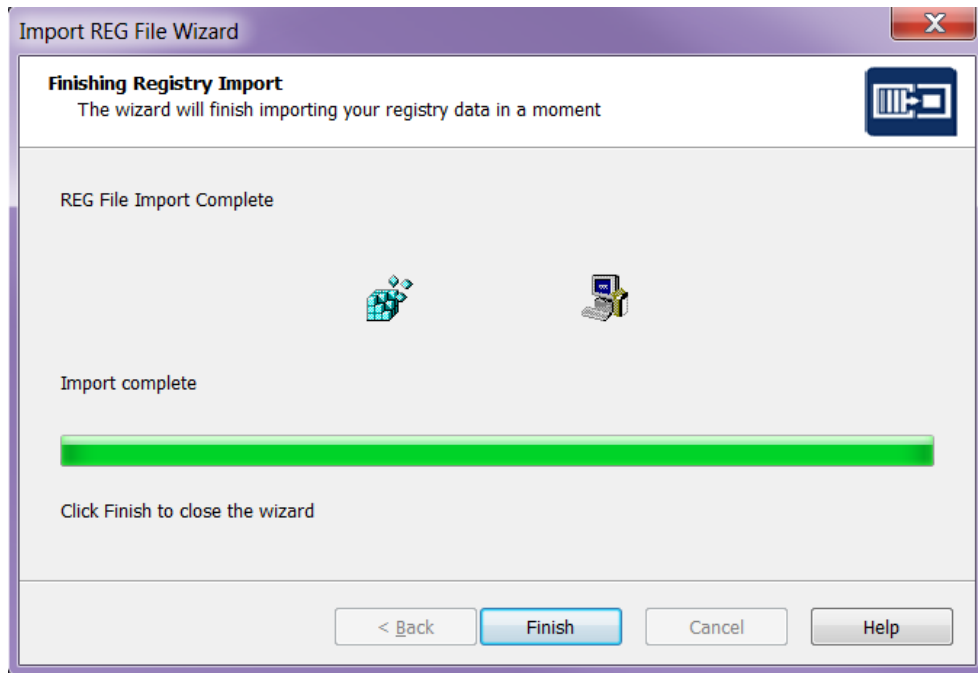
17. To verify that the registry keys have been correctly imported, you can tick the **Yes** radio button and inspect the content of the tree view with registry keys:
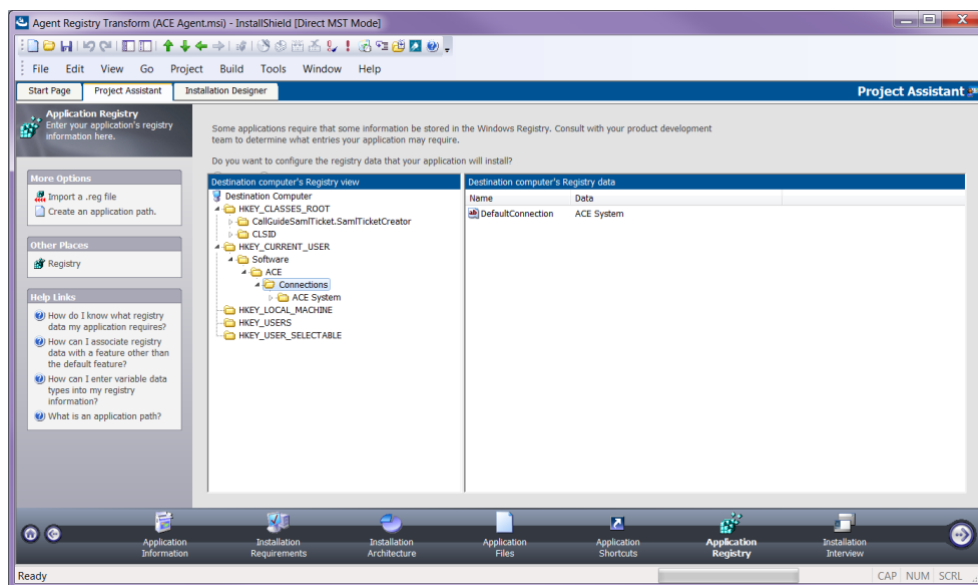
18. Under the Installation Interview header you can specify the dialogs that shall be seen by the user. To reduce the number of choices, you tick **No** for all dialogs:
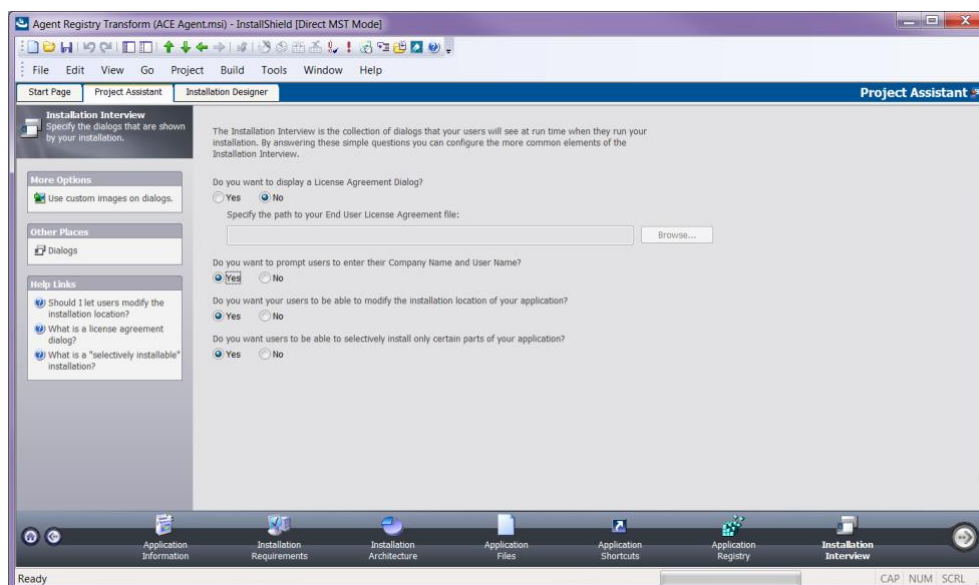
19. To create the very transform file, you select **File/Save As…**
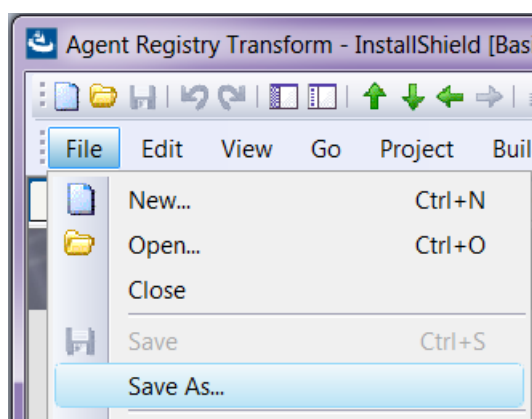
20. In the opened dialog you select the .msi file type and specify a name for the transform:
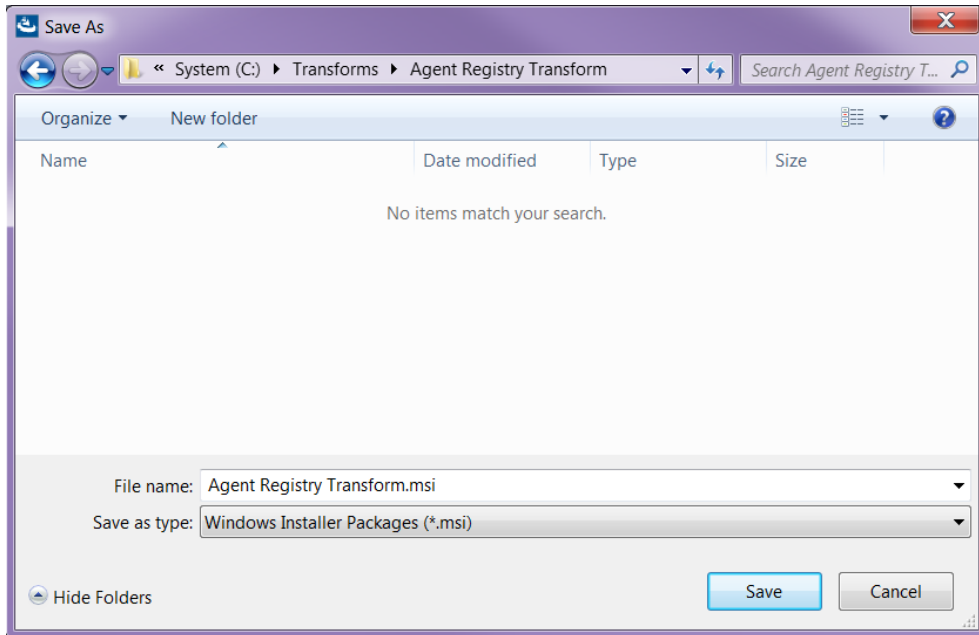
21. Click on **Save** and the transform is created! To install via the transform you can now double-click on the .msi file:
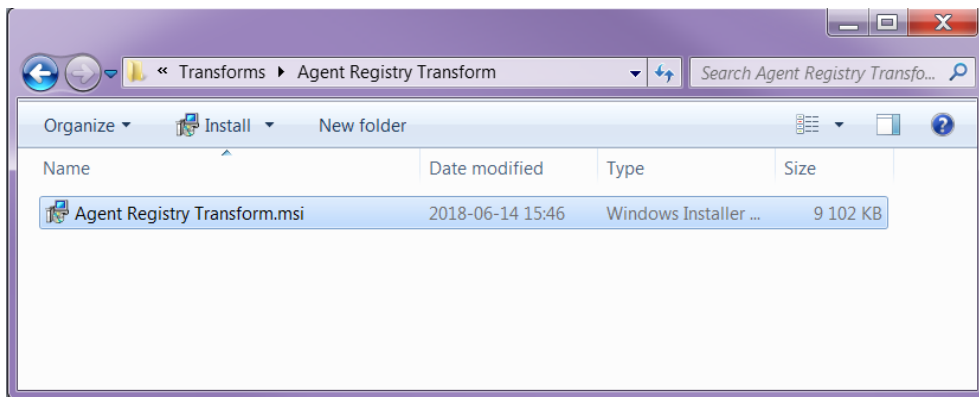
2.2.2 Example of a registry file for 32-bit Windows for desktop clients

The following is an example of a registry file where configurations for ACE Admin, ACE Agent, ACE Pulse are included. Note that text after "--" are comments. For more information on what registry keys to include see *Telia ACE Installed Components.*

```
Windows Registry Editor Version 5.00

-- Key to hard code language in ACE Agent
[HKEY_LOCAL_MACHINE\SOFTWARE\Borland\Locales]
"C:\Program\ACE\Agent\Agent.exe"="SV"

[HKEY_LOCAL_MACHINE\SOFTWARE\ACE]
"LatestUpdate"="20100323 10:31:22"

[HKEY_LOCAL_MACHINE\SOFTWARE\ACE\Connections]
"DefaultConnection"="Connection"

[HKEY_LOCAL_MACHINE\SOFTWARE\ACE\Connections\Connection
"TAPIDriver"="dAnswer - TAPI service"
"TAPIProvider"="dApc"
"DnStorage"="local_machine"
"AutoUpdateEndpoints"=dword:00000001
"LocationName"=""
"ScreenPopConfigFile"="Rules.vbs"

[HKEY_LOCAL_MACHINE\SOFTWARE\ACE\Connections\Connection\xxx.x
xx.xxx.xxx]
"Port"=dword:00003977

[HKEY_LOCAL_MACHINE\SOFTWARE\ACE\Connections\Connection\yyy.y
yy.yyy.yyy]
"Port"=dword:00003977

-- Log settings for ACE clients
[HKEY_LOCAL_MACHINE\SOFTWARE\ACE\Logging]
"LogSize"=dword:000001f4
"LogFilePath"="%ALLUSERSPROFILE%\\Application Data\\ACE"
"NoOfLogFiles"=dword:00000004
"LogLevel"="Debug"
```

2.2.3 Example of a registry file for 64-bit Windows for desktop clients

The following is an example of a registry file where configurations for ACE Admin, ACE Agent, ACE Pulse, are included. Note that text after "--" are comments. For more information on what registry keys to include see *Telia ACE Installed Components.*

```
Windows Registry Editor Version 5.00

-- Key to hard code language in ACE Agent
[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Borland\Locales]
"C:\Program\ACE\Agent\Agent.exe"="SV"

[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\ACE]
"LatestUpdate"="20100323 10:31:22"

[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\ACE\Connections]
```

```
"DefaultConnection"="Connection"

[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\ACE\Connections\Conn
ection
"TAPIDriver"="dAnswer - TAPI service"
"TAPIProvider"="dApc"
"DnStorage"="local_machine"
"AutoUpdateEndpoints"=dword:00000001
"LocationName"=""
"ScreenPopConfigFile"="Rules.vbs"

[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\ACE\Connections\Conn
ection\xxx.xxx.xxx.xxx]
"Port"=dword:00003977

[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\ACE\Connections\Conn
ection\yyy.yyy.yyy.yyy]
"Port"=dword:00003977

-- Log settings for ACE clients
[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\ACE\Logging]
"LogSize"=dword:000001f4
"LogFilePath"="%ALLUSERSPROFILE%\\Application Data\\ACE"
"NoOfLogFiles"=dword:00000009

"LogLevel"="Debug"
```

3   ACE Report

A number of differences and limitations apply to the installation of ACE Report.

The installation program for ACE Report is not a single MSI file but in fact a folder structure with the following components:

- A folder with an installation program for the prerequisite component .NET 4

- A folder with an installation program for the prerequisite component SAP Crystal Reports 32-bit runtime for Microsoft .NET Framework 4.0

- An MSI file for the installation of the ACE Report client

- A setup.exe installing prerequisites and ACE Report

To install correctly, use setup.exe, not the MSI file for ACE Report.

The MSI installation program for ACE Report differs from the installation programs for the other ACE clients:

The installation is built with Microsoft Visual Studio (as opposed to InstallShield) resulting in different style and behaviour in the installation dialogue.

It does not allow for any customization of the features installed. It is thus not possible to de-select installation of language support files or de-select installation of ACE Configuration Utility.

The information in this document regarding how to create an MSI transform has not been validated for ACE Report (or any of its prerequisite components).

When the installation is finished, configurations must be made in ACE Configuration Utility or directly in Windows Registry to be able to run the client.

4 ACE Screenpop used with ACE Interact

When ACE Screenpop should be used with ACE Interact a separate MSI installation program for ACE Screenpop is used.

In addition to installation, configurations for ACE Screenpop and TeliaEdgeAgentPop, the Registered Protocol Handler used by ACE Interact to communicate with ACE Screenpop, is needed in Microsoft Registry.

These configurations can be done manually or via an MSI Transform.

You can for example configure all relevant keys in Microsoft Registry and export your configurations to a registry file that you later attach when constructing an MSI transform. Which keys that shall be exported is described in *Telia ACE Installed Components*.

You find an example of registry files in section 4.1.

In addition to Microsoft Registry settings for ACE Screenpop and TeliaEdgeAgentPop web browser configurations can be made to supress the question asking the users of ACE Interact to approve the use of the Registered Protocol Handler TeliaEdgeAgentPop the first time a screen pop takes place. These configurations are browser specific and are described in *Telia ACE Installed Components*.

Note: When performing a screenpop from ACE Interact only a subset of contact data keys is included in screenpop. For a list of contact data keys that are included see Ref [4].

4.1 Example of registry file for 64-bit Windows for ACE Screenpop with ACE Interact

The following is an example of a registry file where configurations for ACE Screenpop when used with ACE Interact are included. Note that text after "--" are comments. For more information on what registry keys to include see *Telia ACE Installed Components*.

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\ACE\EdgeAgent]
"ScreenPopConfigFile"="C:\\Program Files
(x86)\\ACE\\ScreenPop\\Screenpop.vbs"
"ScreenPopPort"=dword:00000521
"LatestUpdate"="20170207 09:00:00"



[HKEY_CLASSES_ROOT\TeliaEdgeAgentPop]
@="URL:TeliaEdgeAgentPop Protocol"
"URL Protocol"="\"\""

[HKEY_CLASSES_ROOT\TeliaEdgeAgentPop\DefaultIcon]
@="ScreenPop.exe,1"

[HKEY_CLASSES_ROOT\TeliaEdgeAgentPop\shell]

[HKEY_CLASSES_ROOT\TeliaEdgeAgentPop\shell\open]

[HKEY_CLASSES_ROOT\TeliaEdgeAgentPop\shell\open\command]
@="\"C:\\Program Files (x86)\\ACE\\ScreenPop\\ScreenPop.exe\"
\"%1\""
```

5 ACE Connector for Skype

ACE Connector for Skype is installed by running an installation (setup) program. The installation program installs the pre-required Microsoft Lync 2010 SDK Runtime and .Net if not already installed on the computer. ACE Connector for Skype can be run with .NET 3.5 or.Net 4[1].

The installation program adds a shortcut to ACE Connector for Skype in the computer Start-menu.

The installation program will not automatically start ACE Connector for Skype after completed installation unless the computer is restarted. On subsequent user logins ACE Connector for Skype is automatically started since the installation program has added a shortcut to ACE Connector for Skype in the startup folder on the computer.

5.1 ACE Connector for Skype used with ACE Interact

The Installation program for ACE Connector for Skype will, in addition to installation of required files and shortcuts, add Microsoft Registry settings needed when using ACE Connector for Skype with ACE Interact.

The registry keys needed, including the Registered Protocol Handler ACEConnectorForSkype, are described in *Telia ACE Installed Components*.

These configurations can also be done manually or via an MSI Transform.

You can for example configure all relevant keys in Microsoft Registry and export your configurations to a registry file that you later attach when constructing an MSI transform.

You find an example of registry files in section 5.1.1.

In addition to Microsoft Registry settings for ACE Connector for Skype, web browser configurations can be made to supress the question asking the users of ACE Interact to approve the use of the Registered Protocol Handler ACEConnectorForSkype the first time a Skype call should be handled by ACE Interact. These configurations are browser specific and are described in *Telia ACE Installed Components*.

5.1.1 Example of registry file for 64-bit Windows for ACE Connector for Skype with ACE Interact

The following is an example of a registry file where configurations for ACE Connector for Skype when used with ACE Interact are included. For more information on what registry keys to include see *Telia ACE Installed Components*.

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\ACE\ACEConnectorForS
kype]
"ACEConnectorForSkypePort"=dword:000024ee

[HKEY_CLASSES_ROOT\ACEConnectorForSkype]
"URL Protocol"="\"\""
@="URL: ACEConnectorForSkype Protocol"

[HKEY_CLASSES_ROOT\ACEConnectorForSkype\DefaultIcon]
@="\"ACEConnectorForSkype.exe,1\""

[HKEY_CLASSES_ROOT\ACEConnectorForSkype\shell]
```

---

[1] Later versions of .Net 4 are compatible, in-place updates on top of .NET 4 This means that applications built to target.NET 4 will continue running on .NET 4.x.y without change.

```
[HKEY_CLASSES_ROOT\ACEConnectorForSkype\shell\open]

[HKEY_CLASSES_ROOT\ACEConnectorForSkype\shell\open\command]
@="\"C:\\Program Files
(x86)\\ACE\\ACEConnectorForSkype\\ACEConnectorForSkype.exe\"
\"%1\""
```