

SITRAFFIC SCALA

System update, hotfixes, foreign languages
in Scala systems
(Scala V1.5 and higher)

Table of contents

1. Automatic installation of hotfixes.....	4
1.1. Tool HotfixDeployer.....	4
1.1.1. General information.....	4
1.1.2. Mode of operation	4
1.2. Deployment.....	5
1.3. Requirements.....	5
1.4. Applications to quit.....	5
1.5. Hotfixes to be installed manually.....	6
1.6. Checking the version status	6
1.7. Checking for missing hotfixes.....	7
2. Deployment of hotfixes	8
2.1. Deployment on DVD4	8
2.1.1. Directory 10_HFS.....	8
2.1.2. Directory 20_Manually.....	8
2.1.3. Directory 30_NLS.....	8
2.1.4. Other files.....	9
2.2. Deployment in the local Scala system.....	9
2.3. Deployment for Service-MA via Argus or Livelink.....	10
3. Execution of hotfix installation	11
3.1. Recommended procedure.....	11
3.2. Operating procedure of the hotfix deployer	12
4. Installation of foreign languages	21
4.1. Registration of the language packs	21
4.2. Installation and update of the language packs.....	22
4.3. Conversion of the Scala components to the new language	22
5. Automatic hotfix check.....	23
5.1. Requirements.....	23
5.2. Checking for missing hotfixes.....	23
5.2.1. Running the check.....	23
5.2.2. On the current computer.....	24
5.2.3. In the entire Scala system	25
5.3. Set-up of automatic updating	26

List of figures

Figure 1-1: Tool UtcVersionCheck.....	7
Figure 3-1: Start screen of the hotfix deployer	12
Figure 3-2: Display of the processes to quit.....	13
Figure 3-3: Display of the required preliminary and closing work	14
Figure 3-4: Display of the hotfixes that are to be installed.....	15
Figure 3-5: Display of the hotfixes that are NOT to be installed	16
Figure 3-6: Display of the installation status	17
Figure 3-7: Display of the required preliminary and closing work	18
Figure 3-8: Completion dialog.....	19
Figure 4-1: Registration of language packs	21
Figure 5-1: Starter for hotfix check.....	24
Figure 5-2: Hotfix check on current computer	25
Figure 5-3: Hotfix check in the Scala system	26
Figure 5-4: Automation of the hotfix check.....	26

Revision history

Version	Date	Change	Author
1.0	7/7/2008	Initial version	Schmidt, Feigen
1.1	8/25/2008	Suggestions on improvement regarding dialogs, manual changes and log file of the hotfix deployer (V4.3.1.12b2 or higher) incorporated	Schmidt, Feigen
1.2	10/13/2008	Additions made to installation of language packs	Feigen
1.3	10/20/2008	DVD4 directory structure changed	Feigen
1.4	11/3/2008	Deployment directory defined, renamed to SystemAktualisierung_Hotfixes_Fremdsprachen.doc	Feigen
1.5	3/23/2009	Automatic hotfix checks described	Feigen
1.6	6/24/2009	Foreign language installation specified in detail	Feigen
1.7	7/8/2009	DVD4 subdirectories changed, hotfix deployer upgraded	Feigen
1.8	11/27/2009	Scala Share corrected	Feigen

Program version described

Program	Starting with version
HotfixDeployer.exe	5.0.0.18
UtcVersionCheck.exe	1.1
RegisterLanguage.exe	1.2.0.2
Check4Hotfixes.exe	1.0.1.1
StartHotfixChecker.exe	1.0.0.17

Frei verwendbar

1. Automatic installation of hotfixes

1.1. Tool HotfixDeployer

1.1.1. General information

System updates by installing hotfixes and language packs (see chapter 4) are much easier **with SCALA 1.3.1 or higher**. For support during installation of the hotfixes, the application **HotfixDeployer.exe** is used. It automatically identifies the Scala components installed on the computer via setup, determines the hotfixes to be deployed for this and installs them.

The use of the hotfix deployer ensures that all the required hotfixes are installed for each computer in order of ascension and none are "forgotten". Moreover, the version information is automatically updated, which can then be manually requested and is also available for inventory creation using **LogInventory**.

With Scala 1.5 and higher, the tool **Check4Hotfixes.exe** is also available. In connection with a set-up Scala Installation Share this makes it possible to easily check whether hotfixes must be installed for a computer (see chapter 5).

Manual hotfix installations are, of course, still possible as before, but they should only be carried out in exceptional cases because they require substantially increased manual effort. Moreover, the manual changes may be overwritten in the next automatic hotfix installation.

1.1.2. Mode of operation

With the hotfix deployer, all Scala applications that are on a computer are updated; i.e. for every installed Scala component, all available relevant hotfixes are installed. Partial updates are neither possible nor intended. There are, however, applications in the Scala environment that cannot yet be automatically updated via the hotfix deployer (see chapter 1.5).

For IGs and ESes (RTCs), the hotfixes are provided together via the hotfix deployer (in directories IG_Software and ES_Software) and then distributed and transmitted as usual to the Linux/Lynx systems in one step using the config generator.

Already installed hotfixes are not re-installed if the hotfix deployer is run again, rather only those hotfixes whose version and hotfix number is higher than the highest currently registered version of the relevant application. It is to be noted here that only those service packs / hotfixes are installed whose major version number (the first two digits of the version number: e.g. V1.3.*.*) matches that of the installed product version.

When creating the hotfix, it is emphasized that absolutely no manual changes be connected with the installation of the hotfix. If this is, however, unavoidable due to inherent technical necessities, then a notification is to be issued during deployment by the hotfix deployer regarding the necessary steps. They are to be saved in the form of a text file in the automatically created back-up directory in order to be able to execute them after quitting the hotfix deployer. The first notification is to take place before the actual copy operation so that it is possible to abort and the required steps can be planned in. Another notification is to be issued after the copy operation regarding any later work.

Also, a back-up is created for all replaced files so that it is possible to manually restore the state before installation of the hotfix. For such manual restoration, please refer to the support center.

The hotfix deployer can also be run in a **check-only mode** in which the interface behaves as

Frei verwendbar

described but no files are updated.

1.2. Deployment

Service packs, hotfixes and language packs are deployed in a defined directory structure (for details, see chapter 2.1):

- 10_HFS
- 20_Manually
- 30_NLS

1.3. Requirements

The hotfix deployer can only function correctly if the following pre-conditions are satisfied:

- The Scala components were installed normally via their setups (version Scala V1.3.1 or higher) using the installation manual
- The hotfixes are stored in a defined directory structure (see chapter 2.1)
- The hotfixes are correctly named (this is ensured by I MO TS R&D ST). The file and directory names may no longer be renamed on site.
- The release notes on the hotfixes are present and correctly named (this is ensured by I MO TS R&D ST). The file names may no longer be renamed on site.
- No manual changes to the program environments that are evaluated by the hotfix deployer have been made (registry entries, installation directories, etc.)
- During execution of the hotfix deployer, all Scala applications are ended on the relevant computer (see chapter 1.4)
- The hotfix deployer is to be run from the root directory of the defined directory structure, i.e. from `.\\10_HFS` or from `.\\30_NLS`.

Notes:

- The contents of the complete DVD4 should be copied to a local deployment directory and the hotfix deployer should be run from there. This avoids problems with write protections and also allows later addition with hotfixes provided at a later time.
- The hotfix deployer (and the hotfix directory structure) must be located in a directory to which a drive letter is assigned. UNC paths are not supported.

1.4. Applications to quit

Before deploying hotfixes on a computer, all Scala applications (even background processes) must be ended in order to prevent conflicts when overwriting files. Currently, on the relevant computer (depending on the local situation) these are:

Computer	Program
ConfigServer	<ul style="list-style-type: none">• SITRAFFIC Supply Server Service• SITRAFFIC Server• SITRAFFIC Tomcat Service• SITRAFFIC Jini Service• Concert Client, if applicable• Supply• Office• Office ProfileManager

Frei verwendbar

Computer	Program
AS	· Monitor
CS	· Monitor
GS	· Monitor
MOTIONServer	· Monitor
WSx	· Concert (incl. Visus) · Office · Office ProfileManager · ...
Office Standalone (VIAP)	· SITRAFFIC Server · Office · Office ProfileManager

1.5. Hotfixes to be installed manually

Some Scala applications cannot yet be updated via the hotfix deployer for various technical reasons (for process, see chapter 3.1). To avoid confusion between hotfixes that can be automatically installed by the hotfix deployer and those for which this is not possible, the hotfixes of the applications affected are stored in the directory **.\\20_Manually** (in contrast to the regular fixes under in **.\\10_HFS**).

The applications stored in **.\\20_Manually** must be **installed manually** as described in the relevant release notes enclosed. Likewise, it must be ensured that the release notes files of all installed hotfixes are copied to the destination computer (for Scala applications in **e:\\sitraffic\\releasenotes**), and the script **createIndex.cmd** (also in **e:\\sitraffic\\releasenotes**) is executed.

1.6. Checking the version status

The current version status, both before and after installation of the hotfix can be checked with the tool **UtcVersionCheck.exe** (Scala V1.3.1 or higher). The tool is located on DVD4 (see chapter 2.1.4).

The tool shows the version info of all programs compatible with the hotfix deployer and the hotfixes installed via the hotfix deployer (see below). Manually installed hotfixes are not detected.

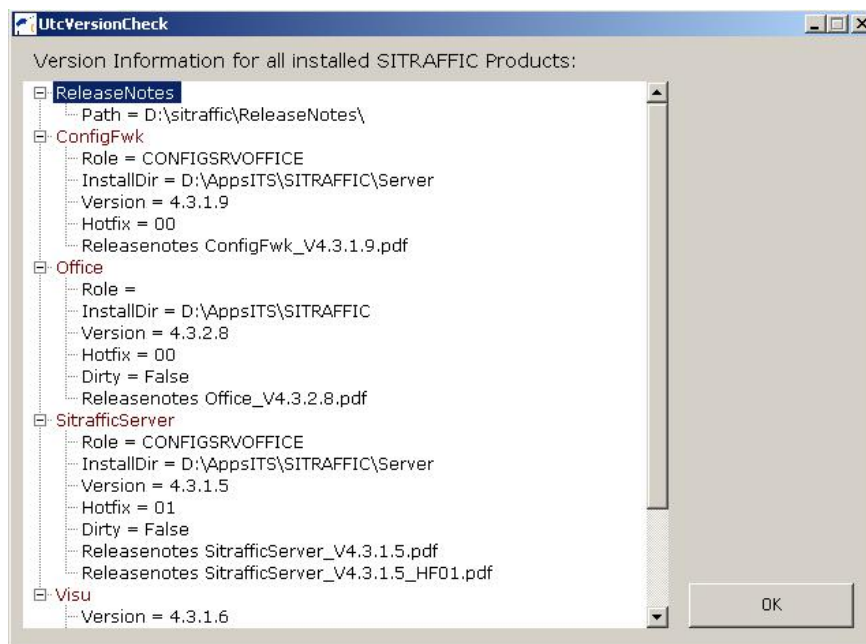


Figure 1-1: Tool UtcVersionCheck

Another option for checking is via System Control -> Software:

All installed Scala components appear here again with the prefix **Scala_.....** A requirement for this is that the release notes files have been correctly copied and the script **createIndex.cmd** has been executed. Then manual hotfixes too are detected here.

1.7. Checking for missing hotfixes

In a Scala system, the hotfixes must be installed on all the computers affected. The **Check4Hotfixes.exe** tool can be used (with Scala V1.5 or higher) to check whether hotfixes already provided on a computer still have to be installed. The tool is located on DVD4 (see chapter 2.1.4).

The tool shows for the current computer (or the entire Scala system) which hotfixes still have to be installed. A description of procedure can be found in chapter 5.

2. Deployment of hotfixes

2.1. Deployment on DVD4

The hotfixes are to be deployed with the Scala installation storage media on DVD4. They contain the relevant hotfixes that were current at the time the DVDs were put together. New hotfixes after this point in time are to be distributed via the known channels by CS&L (see chapter 2.2).

DVD4 is organized into the following directory structure:

- 10_HFS (previously: 10_Hotfixes)
- 20_Manually (previously: 20_ManualUpdates)
- 30_NLS (previously: 30_LanguagePacks)

The long directory names used in previous versions (see above: previously...) are still supported but should no longer be used if possible.

! Warning !:

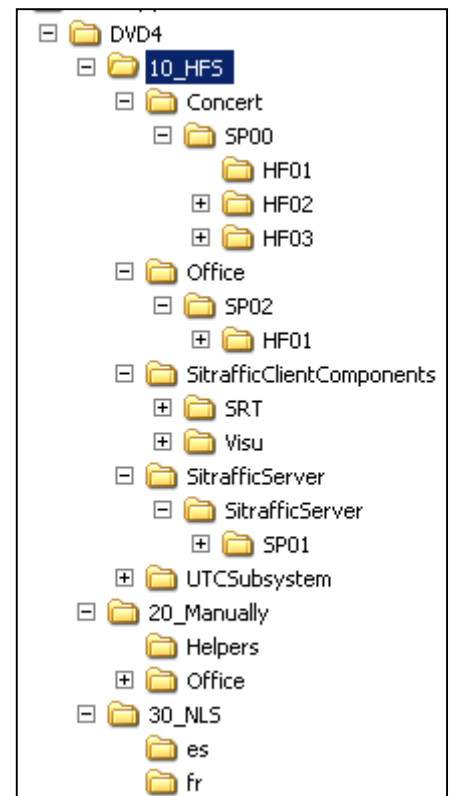
If there is still an old directory structure without directories with leading numbers, it must be completely changed or created from scratch. A mixed structure is not supported.

2.1.1. Directory 10_HFS

The directory **10_HFS** contains the hotfixes that can be automatically installed using the hotfix deployer. The hotfix deployer is therefore stored in this directory and can be run from it.

The directory structure in which the hotfixes are stored largely corresponds to that of the application installation on DVD2, i.e.

- Concert
- Motion
- Office
- SitrafficClientComponents
- SitrafficServer
- Supply
- TrafficTools
- UTCSubsystem



The directory structure including the subdirectories may not be changed because the hotfixes would otherwise no longer be found by the hotfix deployer and therefore could not be installed.

2.1.2. Directory 20_Manually

The directory **20_Manually** contains the hotfixes of the applications that do not support the hotfix deployment mechanism (see chapter 1.5). The updates are to be manually introduced according to the name of the relevant subdirectory (for procedure, see chapter 3.1).

The directory 20_Manually is not installed by the hotfix deployer.

2.1.3. Directory 30_NLS

The directory **30_NLS** contains the language packs for the SITRAFFIC components for the
Frei verwendbar

currently supported languages. The language packs can also be automatically installed using the hotfix deployer. The hotfix deployer is therefore stored in this directory and can be run from it (see chapter 4).

2.1.4. Other files

Other elements on DVD4 are:

File	Description	Directory
HotfixDeployer.exe	HotfixDeployer (installation in "armed" mode)	10_HFS 30_NLS
HotfixDeployer_Checkonly.cmd	Script for bringing up the hotfix deployer in simulation mode	10_HFS 30_NLS
Readme_ManualUpdates.txt	Readme on the manual updates	20_Manually
utcversion.txt	List of baselines of the basic package	<basis>
UtcVersionCheck.exe	Tool for checking the current version status	<basis>
Check4Hotfixes.exe	Tool for checking for missing hotfixes	<basis>
RegisterLanguage.exe	Tool for registering foreign languages	30_NLS

2.2. Deployment in the local Scala system

It is recommended to set up a deployment directory on the relevant Scala system in which all the hotfixes are to be gathered and to make it available in the system via sharing. All hotfixes can then be accessed via sharing from the individual computers (servers and clients) as well as installed from them as described in this document.

The shared directory should only be used for installation purposes and not for system-wide distribution of other files or customer data.

The shared directory is to be created on a Scala server accessible via cRSP (usually *configsrv*) on a suitable partition (mind the size) in the root directory and enabled for system sharing (normally only visible in the network) with full access permissions.

Recommendation

Share: **<lw>:\scalainstshare**

Sharing: **scalainstshare\$**

If this sharing is not visible to normal users, the permissions (e.g. for the user) must be modified accordingly.

The following subdirectories are to be set up in units in the shared directory:

```
<lw>:\scalainstshare
├── media
│   ├── DVD2:    Copy of installation DVD2
│   └── DVD4:    Copy of hotfix DVD4 and collection point for new hotfixes
├── mastercfg:  Storage of master configuration files, e.g. Topology.xml
└── hoststate:  Collection of hotfix info from the individual hosts
```

The deployment directory **<lw>:\scalainstshare\media\DVD4** contains the contents of the DVD4 provided as well as all hotfixes provided at a later time (see chapter 2.3).

A link between the shared directory with the **local drive letter S:** (recommended, if available)

Frei verwendbar

can be set up on every computer of the Scala system from which access to the shared directory is desired, even on the *configsrv* computer itself.

! Warning !:

The structure of the deployment directory has been reorganized. See chapter 2.1.

2.3. Deployment for Service-MA via Argus or Livelink

In addition to deployment on DVD4 (see chapter 2.1), hotfixes of CS&L are also to be distributed via appropriately made directories via Argus and Livelink in order to be able to make the hotfixes available as soon as possible. Depending on the system, the files of the hotfixes are to be stored not individually but bundled in the form of zip files. The zip files can contain one or more hotfixes.

These zip files are structured internally in such a way that extracting their contents into a local **deployment directory** (for Scala systems, see chapter 2.2) reproduces the structure of DVD4 exactly.

To prepare an update, the following steps are therefore necessary:

- Copy the currently available DVD4 into a deployment directory, e.g. **e:\sitraffic\scalainst\dvd4** (for Scala systems, see chapter 2.2)
! Warning!
The Scala version on DVD4 must match the installed version on the system.
- Extract all newer hotfixes that are not yet included on DVD4 into the deployment directory.
- Execute hotfix installations from the deployment directory (see chapter 3.1).

The deployment directory must contain the complete and correct directory structure including the root directory and any help directories before executing the hotfix deployer.

The path to the deployment directory may not contain any blanks and may be 50 characters at most.

! Warning !:

All hotfixes must be extracted in all cases because otherwise unpredictable system behavior may result.

Hotfixes that are not to be extracted into the deployment directory are missing during installation and are therefore not installed. **These can no longer be installed at a later time without the support of the support center.**

3. Execution of hotfix installation

This chapter describes hotfix installation using the hotfix deployer.

3.1. *Recommended procedure*

- 1) Move to DVD4 or the deployment directory.
- 2) Run HotfixDeployer in CheckMode in directory `.\10_HFS` (or `.\30_NLS`). This step is only used for a first glance and is not absolutely necessary.
- 3) Quit the applications to be updated.
- 4) Carry out any preparatory manual changes.
- 5) Run HotfixDeployer in "armed" mode in directory `.\10_HFS` (or `.\30_NLS`).
- 6) Update the IGs and ESes (RTCs) with ConfigGenerator (if an update was included for these)
- 7) Carry out any manual changes needed afterwards.
- 8) In case of problems, look at the log file and consult the TS Support Center if needed
- 9) Check the update with the program UtcVersionCheck.exe.
- 10) If needed, install the hotfixes from the directory `.\20_Manually`.
- 11) After updating all computers, carry out log inventory scan and send results to TS Support Center.

3.2. Operating procedure of the hotfix deployer

The hotfix deployer (HotfixDeployer.exe or HotfixDeployer_Checkonly.cmd) is to be run from the deployment storage medium / directory which also contains the current hotfixes and/or language packs.

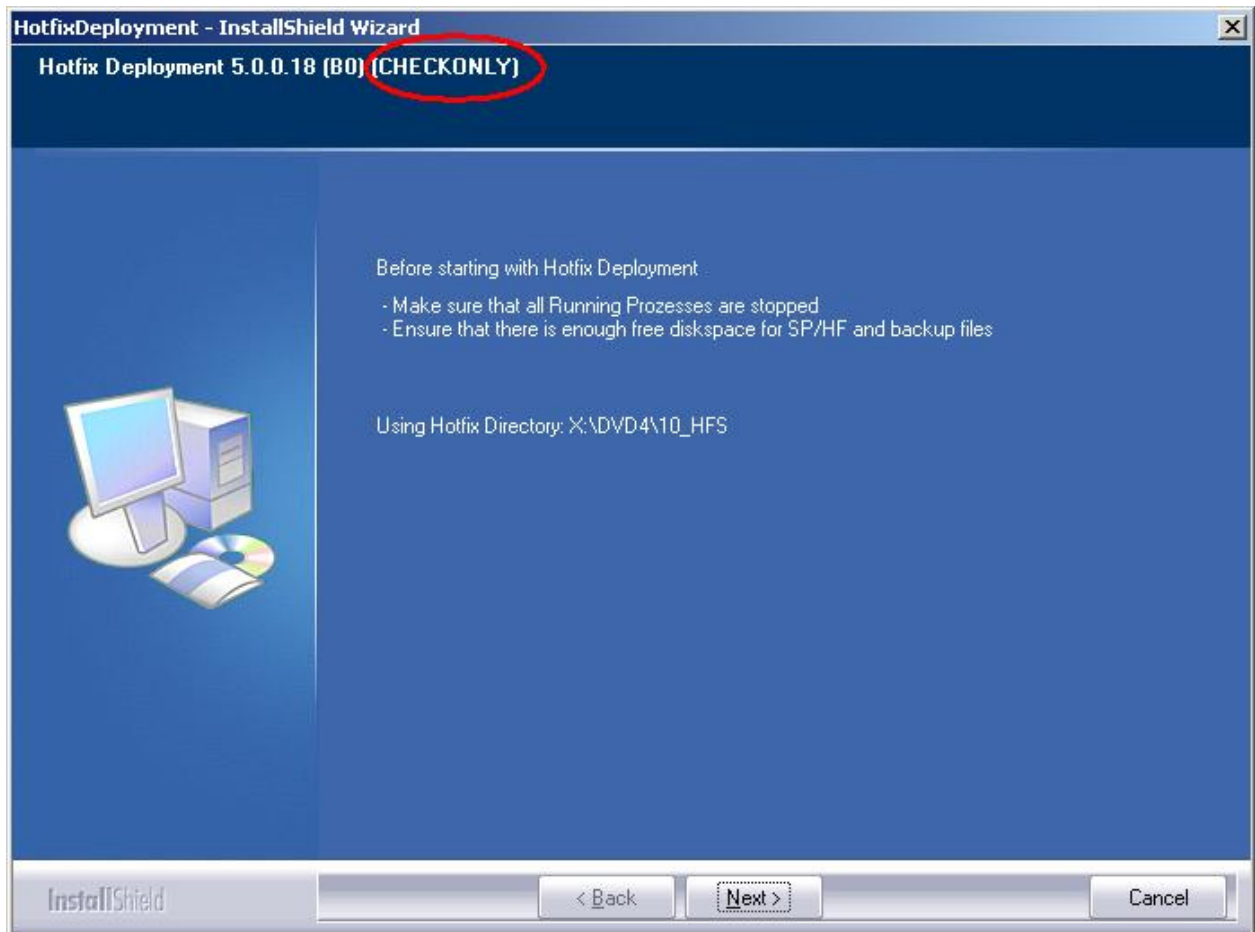


Figure 3-1: Start screen of the hotfix deployer

Start screen of the deployer. Mode is not displayed in "armed mode" (marked red). The directory from which the hotfixes are to be used also appears.

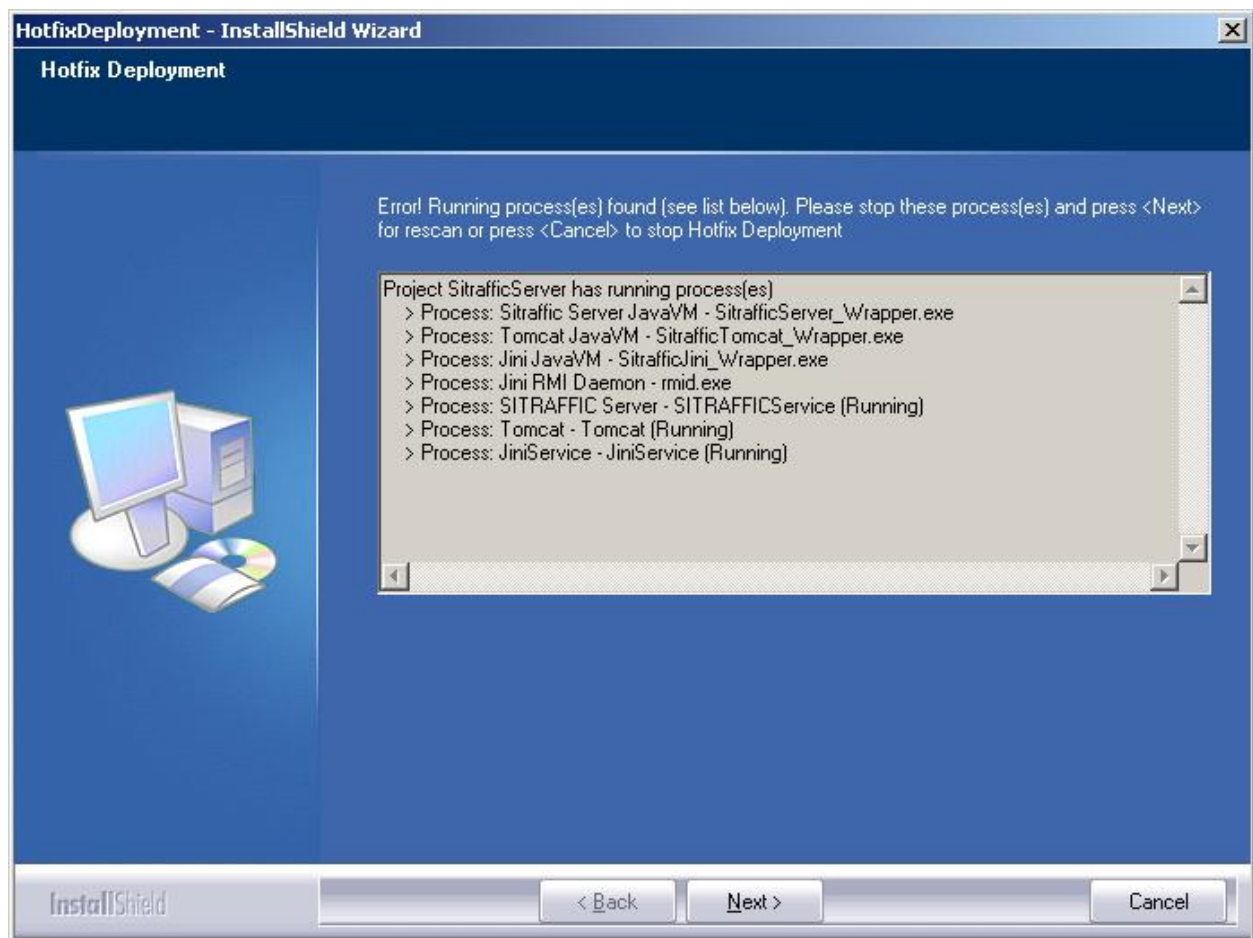


Figure 3-2: Display of the processes to quit

Display of the processes still running that prevent the installation of the hotfixes. These processes must be ended manually before the installation of the hotfixes can be continued.

The process check can be repeated via *Next*. Only when all processes have been ended does *Next* bring you to the next installation step.

Alternatively, this can be aborted via *Cancel*.

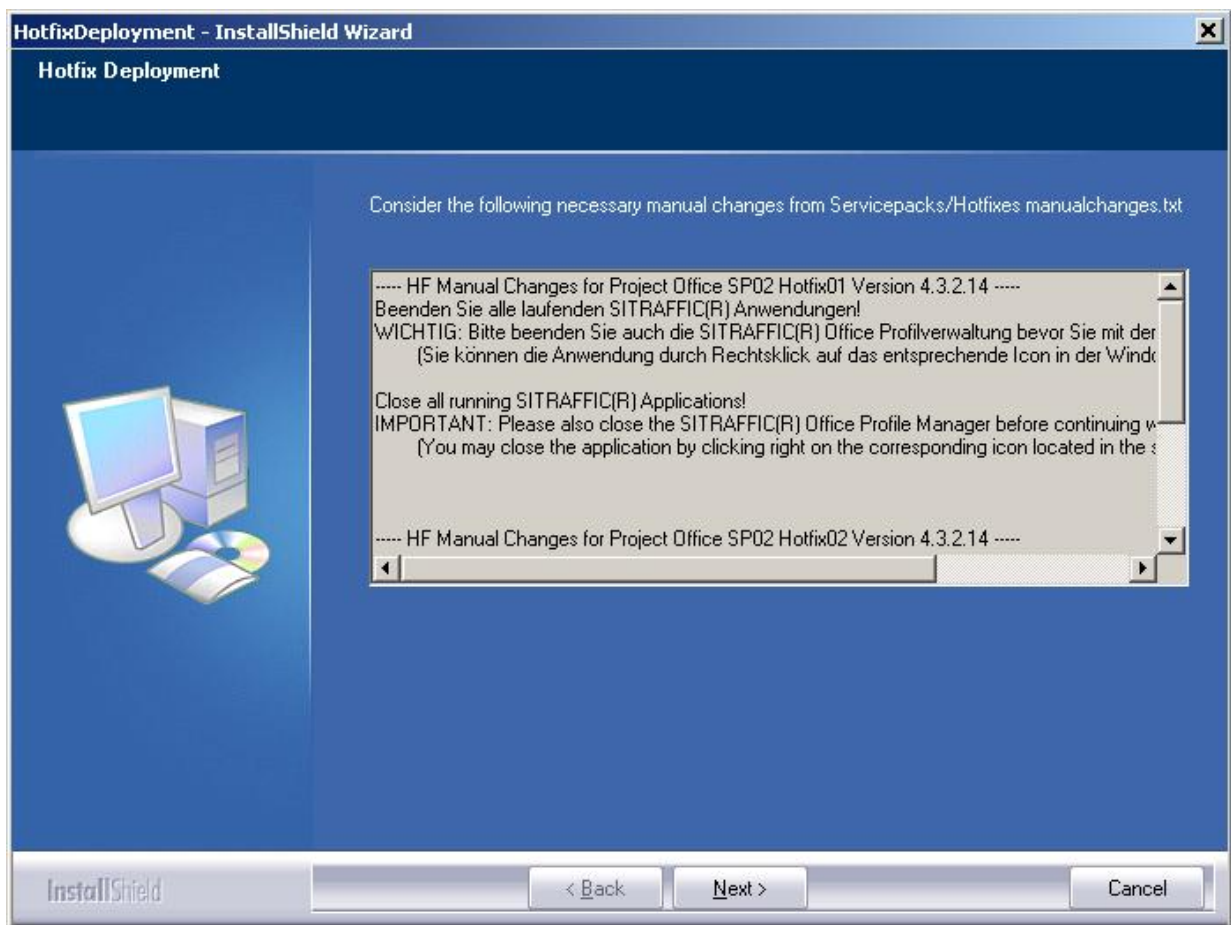


Figure 3-3: Display of the required preliminary and closing work

Display of required preliminary and closing work, i.e. work BEFORE and AFTER the actual installation of the hotfix. If no specific indication is made at the time (i.e. preliminary or closing work is concerned), closing work is to be assumed.

In the standard case in which no manual work is required, the following is displayed:
"No necessary manual changes found!"

The texts on preliminary and closing work displayed here are automatically saved in the log directory and displayed again after hotfix installation.

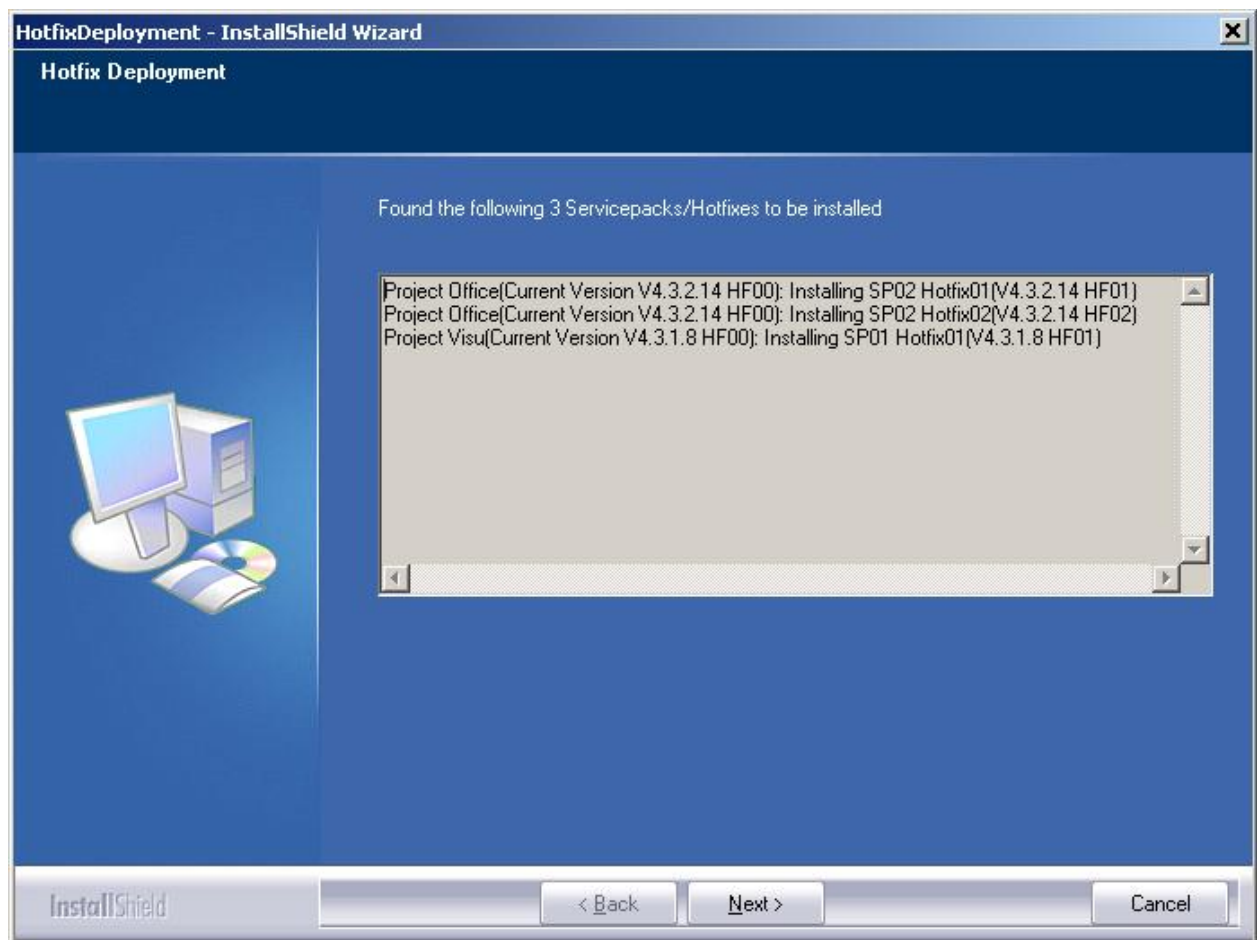


Figure 3-4: Display of the hotfixes that are to be installed

Display of the hotfixes that are to be installed. Please perform a rough check of plausibility using the deployment document and using the available hotfixes in the deployment storage medium / directory. Please also note that some hotfixes are only to be installed in certain server roles, i.e. it is normal if the deployed hotfixes are not installed on every single computer. In case of doubt, please consult the release notes.

The hotfix installation can still be aborted via *Cancel*.

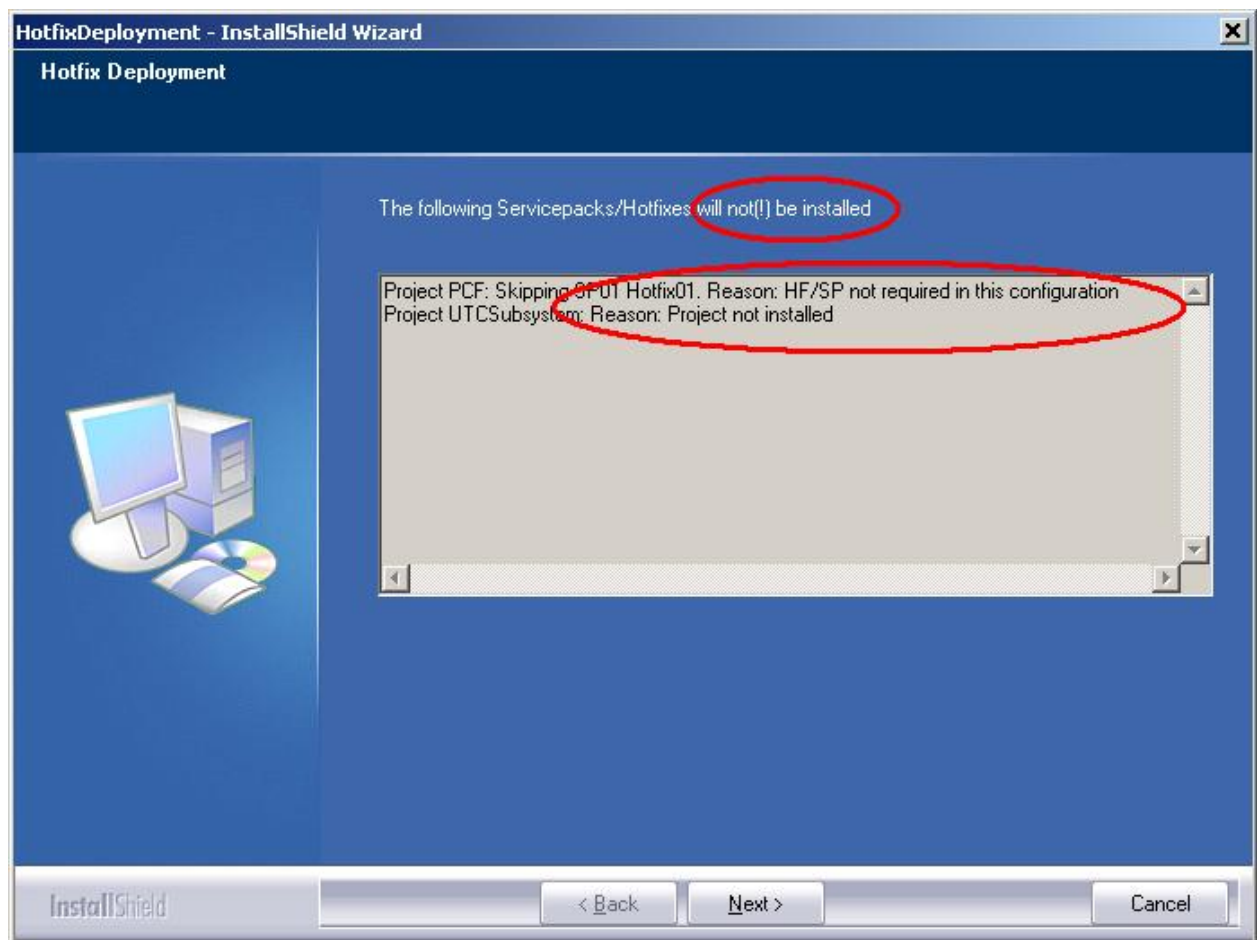


Figure 3-5: Display of the hotfixes that are NOT to be installed

Display of the hotfixes in the deployment storage medium / directory that are NOT to be installed (including a short justification). This display is also used for plausibility checking.

It can still be aborted via *Cancel*.

After clicking on *Next*, the relevant hotfixes are installed. During the installation of the hotfix, a small window with the hotfix data and a copy counter is displayed. For hotfixes for which only few or small files are to be installed, installation is very quick, so the text may not be easy to read. This is not an error.

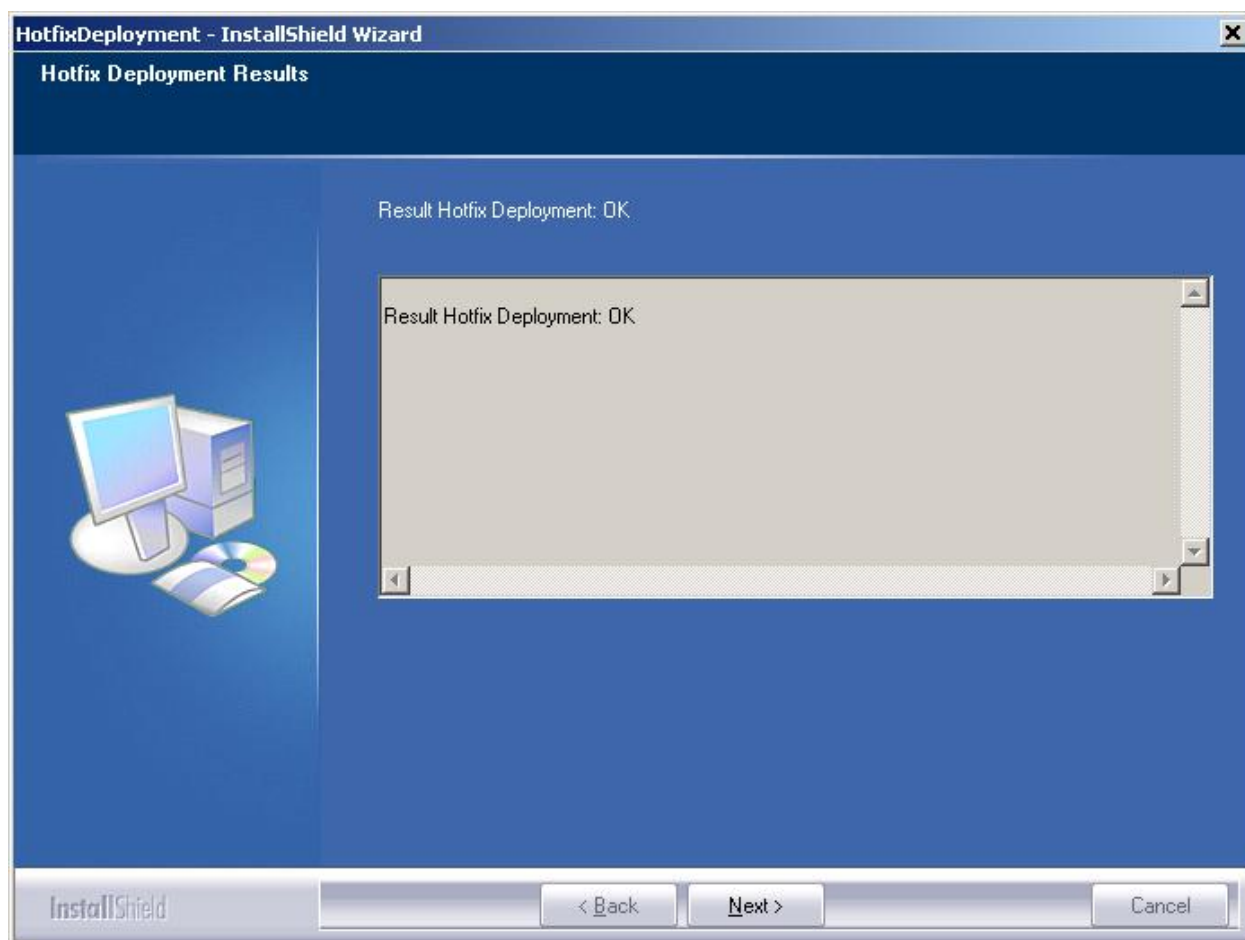


Figure 3-6: Display of the installation status

Display of the status of the installation. In case of errors, the log file (dialog after next) is to be sent.

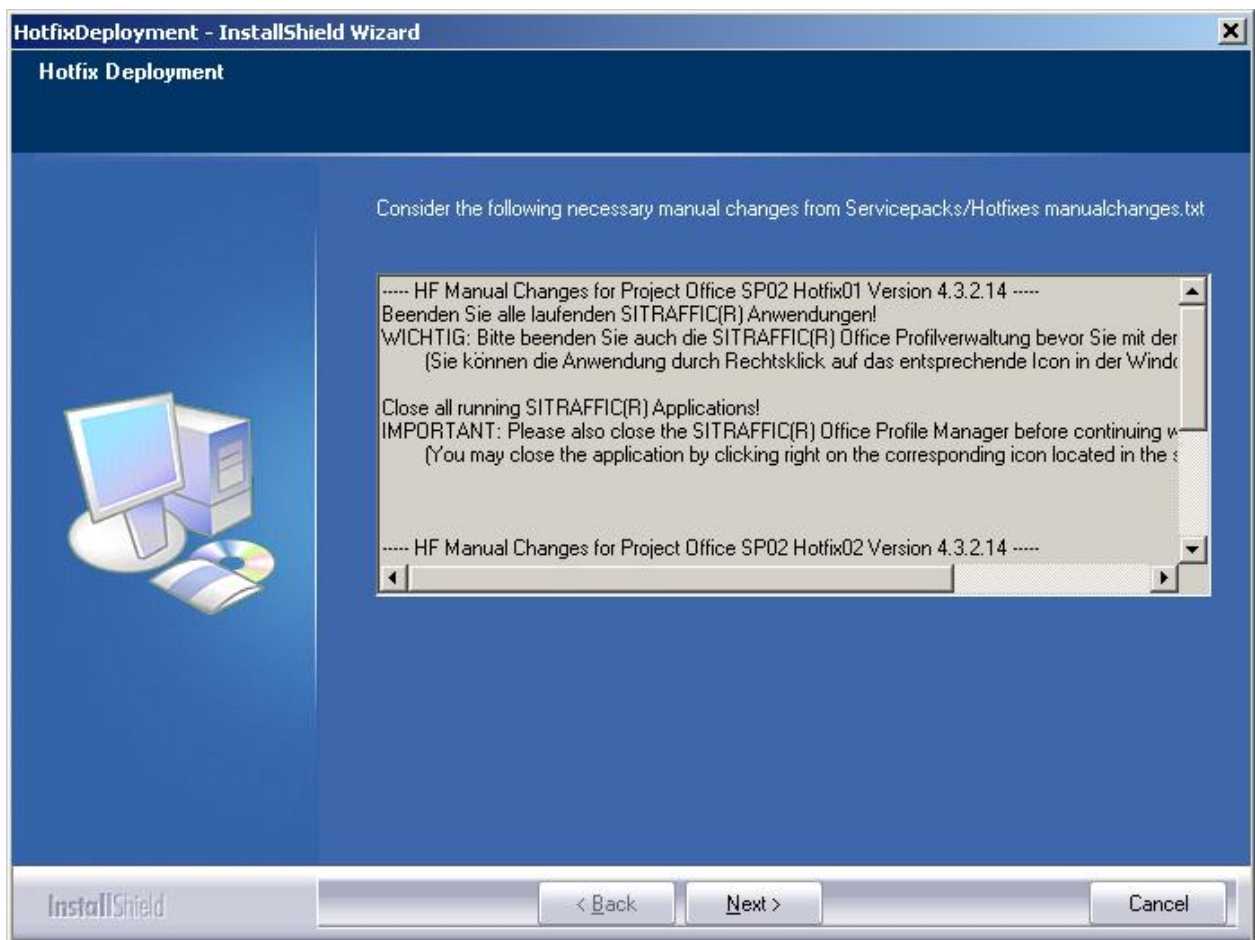


Figure 3-7: Display of the required preliminary and closing work

Display once again of the changes to be carried out manually, now relevant for closing work.

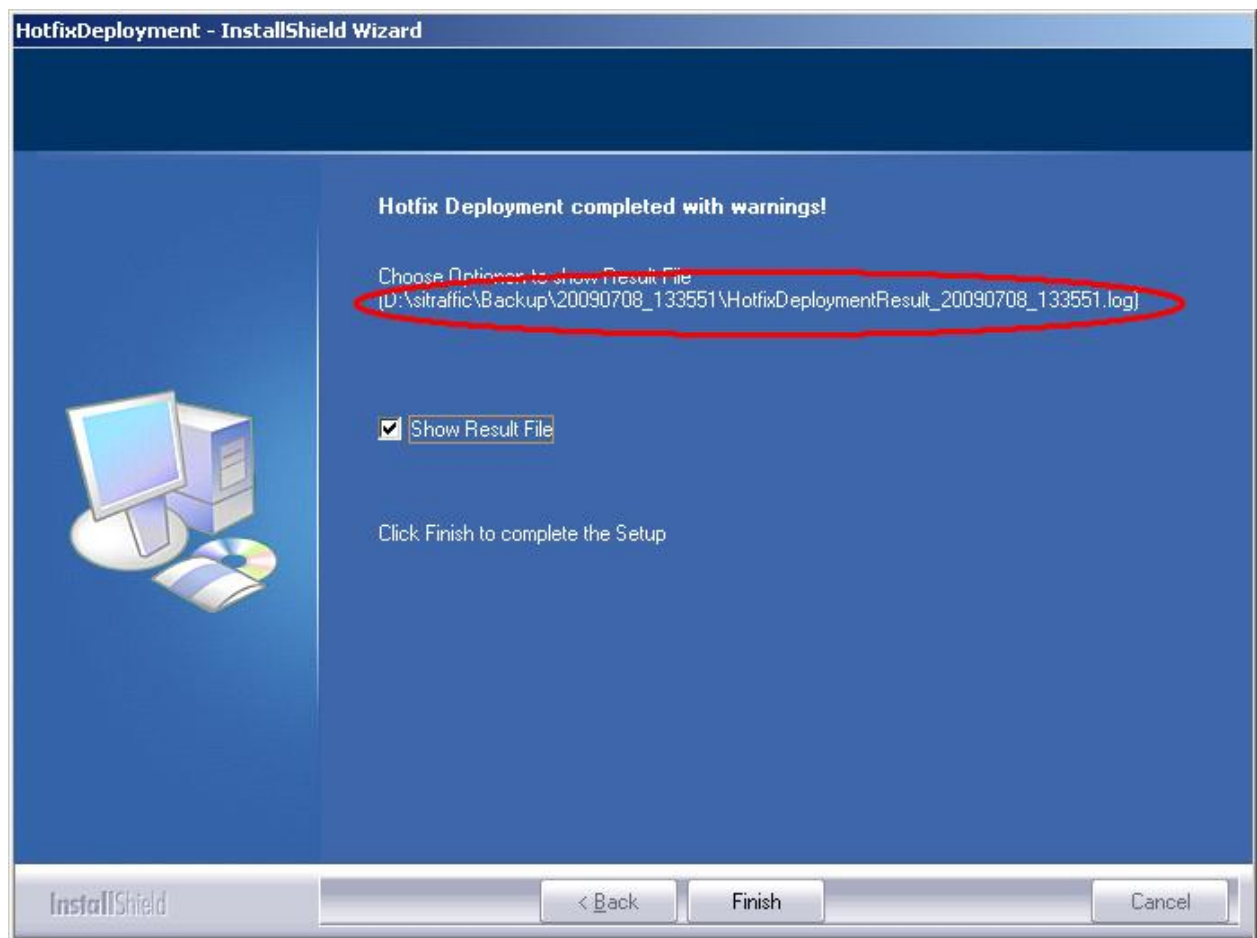


Figure 3-8: Completion dialog

Completion dialog.

The result file that can also be displayed here (see check marks) contains the required manual changes and the list of all installed / not-installed service packs and hotfixes. The path of this log file has a red frame on top (see example below).

In case of errors:

If errors arise during the update, these have been logged in a separate log file (same name but without the addition of "...Result") in the same directory. Please submit this log file as part of an error report to the support center.

Result File (example):

```
-----
----- List of all Manual Changes -----
----- HF Manual Changes for Project Office SP02 Hotfix01 Version 4.3.2.14 -----
Beenden Sie alle laufenden SITRAFFIC(R) Anwendungen!
WICHTIG: Bitte beenden Sie auch die SITRAFFIC(R) Office Profilverwaltung bevor Sie mit der
Installation des Hotfix fortfahren!
(Sie können die Anwendung durch Rechtsklick auf das entsprechende Icon in der Windows
Taskleiste beenden.)

Close all running SITRAFFIC(R) Applications!
IMPORTANT: Please also close the SITRAFFIC(R) Office Profile Manager before continuing with the
hotfix installation!
(You may close the application by clicking right on the corresponding icon located in the
```

Frei verwendbar

system tray area.)

----- HF Manual Changes for Project Office SP02 Hotfix02 Version 4.3.2.14 -----

Beenden Sie alle laufenden SITRAFFIC(R) Anwendungen!

WICHTIG: Bitte beenden Sie auch die SITRAFFIC(R) Office Profilverwaltung bevor Sie mit der Installation des Hotfix fortfahren!

(Sie können die Anwendung durch Rechtsklick auf das entsprechende Icon in der Windows Taskleiste beenden.)

Close all running SITRAFFIC(R) Applications!

IMPORTANT: Please also close the SITRAFFIC(R) Office Profile Manager before continuing with the hotfix installation!

(You may close the application by clicking right on the corresponding icon located in the system tray area.)

---- Found the following 3 Servicepacks/Hotfixes to be installed ----

Project Office(Current Version V4.3.2.14 HF00): Installing SP02 Hotfix01(V4.3.2.14 HF01)

Project Office(Current Version V4.3.2.14 HF00): Installing SP02 Hotfix02(V4.3.2.14 HF02)

Project Visu(Current Version V4.3.1.8 HF00): Installing SP01 Hotfix01(V4.3.1.8 HF01)

---- The following Servicepacks/Hotfixes will not(!) be installed ----

Project PCF: Skipping SP01 Hotfix01. Reason: HF/SP not required in this configuration

Project UTCSysystem: Reason: Project not installed

Frei verwendbar

4. Installation of foreign languages

This chapter describes the installation of language packs using the tools **RegisterLanguage.exe** and **HotfixDeployer.exe**.

The foreign language installation is to be carried out separately on each Scala computer (servers and clients) or on a Sitraffic Office notebook.

It is only not required on client computers that contain exclusively small-client portions (configuration via server), such as a simple Sitraffic Concert client.

4.1. Registration of the language packs

The foreign language expansions of the individual Scala products are delivered in units in the form of language packs that are also structured like hotfixes and can be assigned uniquely to the associated product via the naming convention (e.g. Office-LangENus_V4.3.2.0). This way, the same mechanisms can be used for installation as for hotfixes (see HotfixDeployer).

Before using a foreign language for the first time, the individual language packs must be declared (registered) in the system. This is most easily performed with the tool

RegisterLanguage.exe (in the directory .\30_NLS of DVD4 or the deployment directory), which must be called up on each of the computers concerned (see above).

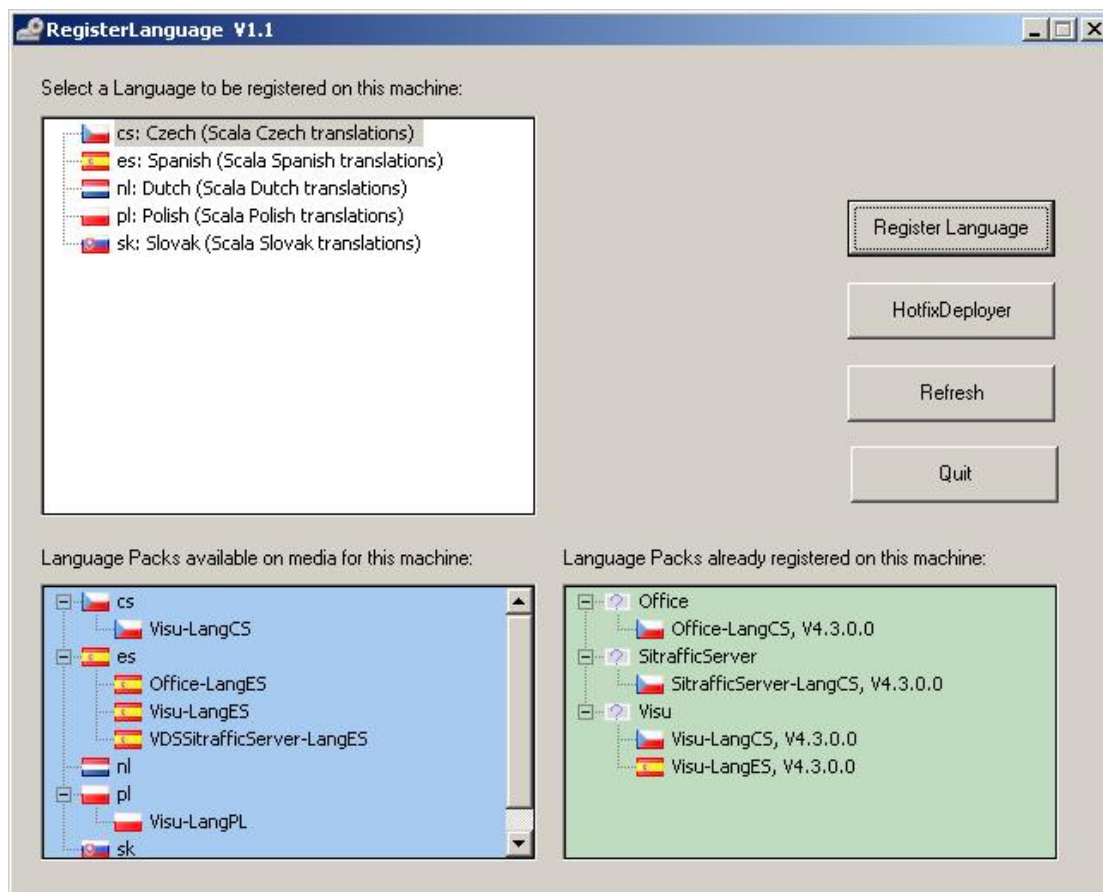


Figure 4-1: Registration of language packs

After the call, the tool displays all the languages for which a language pack has been found on the data storage medium (DVD4 or deployment directory) and with which they can be registered

Frei verwendbar

on the computer.

To register a language, simply select it in the white window and then press the button **Register Language**. This is the way all language packs are installed for which the associated Scala component is already registered. The result can be seen in the green window.

Next, the language packs just registered can be installed directly with the hotfix deployer (button "**HotfixDeployer**"), also see chapter 4.2

4.2. *Installation and update of the language packs*

The installation of a language pack or even an updates (hotfix) for it is performed exactly like that of regular hotfixes with the tool **HotfixDeployer.exe** (see chapter 3).

Requirement for the installation of a language pack:

- the necessary language packs are registered (see chapter 4.1),
- the language packs for the foreign language needed (and any hotfixes for it) are located in the directory `.\30_NLS` on the data storage medium (DVD4 or deployment directory) and
- the hotfix deployer is called up from the directory `.\30_NLS`.

Like with hotfixes, the installed version status of the language packs can be checked with the tool **UtcVersionCheck.exe**.

! Warning !:

Upon executing the hotfix deployer from the directory `.\30_NLS`, only the language packs of the registered languages are installed but not the regular hotfixes (program updates) for the registered Scala components.

4.3. *Conversion of the Scala components to the new language*

After the language packs for the foreign language needed have been installed on all computers concerned, the Scala components must be converted to the new language.

The process for this depends on the relevant Scala component and can therefore be found in the relevant description.

5. Automatic hotfix check

This chapter describes how to check for hotfixes still to be installed using the tools **Check4Hotfixes.exe** and **HotfixDeployer.exe**. This function is available with Scala 1.5 or higher.

5.1. Requirements

Checking for hotfixes still to be installed is based on the comparison of the Scala components locally installed on a computer with the entirety of the available hotfixes and service packs.

The following requirements must be met for this function:

- The directory structure **scalainstshare** must be set up in the Scala system on a central server accessible from all other computers (see chapter 2.2).
- The directory structure **scalainstshare** must always be updated to the current hotfix status (see chapter 2.2).
- The hotfix check must be run on the individual computers of the Scala system (servers and clients) at regular intervals (see chapter 5.3).

Alternatively, the check can be performed against an available (and current) DVD4. In this case, however, only the current computer can be checked due to write protection (read-only medium).

5.2. Checking for missing hotfixes

5.2.1. Running the check

The tool **Check4Hotfixes.exe** provides an overview of the hotfixes still not installed. The overview may refer to the current computer (see chapter 5.2.2) or the entire Scala system (see chapter 5.2.3).

The check is based on the collected hotfixes that are stored in the DVD4 directory of the central Scala shared installation directory **scalainstshare**. The tool **Check4Hotfixes.exe** must therefore also be called up from this DVD4 directory. This can be carried out manually (via selection of the relevant directory in the shared directory) or instead using the starter **StartHotfixChecker.exe**.

This tool is available on the individual computers via the start menu (Start -> Programs -> ...). The exact storage location depends on the relevant Scala components installed. On the configsrv computer, for example, this is Start -> Programs -> Sitraffic Server -> Support -> CheckForHotfixes.

For the initial call, the DVD4 directory must be selected once on the central Scala shared installation directory. To do so, the shared directory may have to first be assigned to a local drive letter, UNC paths (e.g. [\\server\dir1\dir2](#)) are not supported.

The call for the hotfix check is then performed simply by clicking the button "*Start Hotfix Check*".

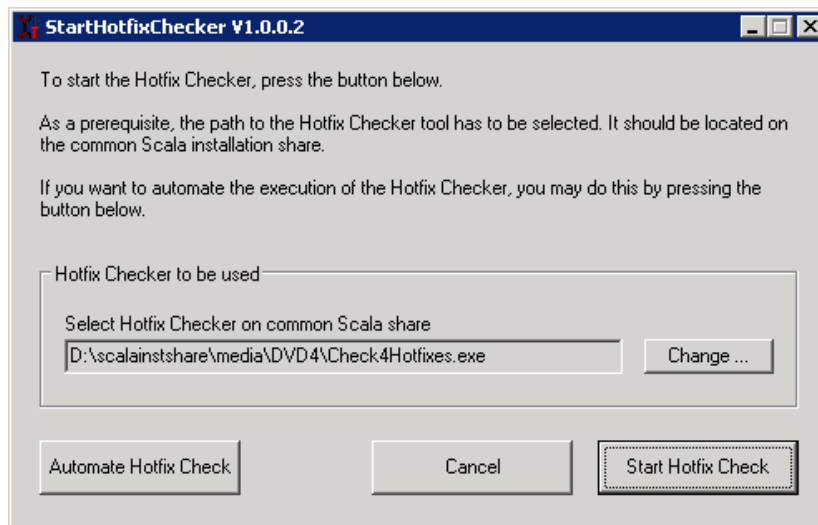


Figure 5-1: Starter for hotfix check

5.2.2. On the current computer

After start-up, the tool Check4Hotfixes.exe by default shows all Scala components on the current computer that are indeed available on the Scala shared installation directory but have not yet been installed on this computer. The check (scan) can be repeated at any time using the button "*Check now*".

If needed, the hotfixes not yet installed can also be installed directly. The button "*Install Hotfixes*" runs the hotfix deployer twice: once to install the hotfixes themselves and a second time it install any language packs.

! Warning !:

The installation of the hotfixes should be performed by a Siemens staff member because there may be dependencies to other computers (especially to the servers) that must also be updated. Otherwise, unpredictable malfunctions may arise.

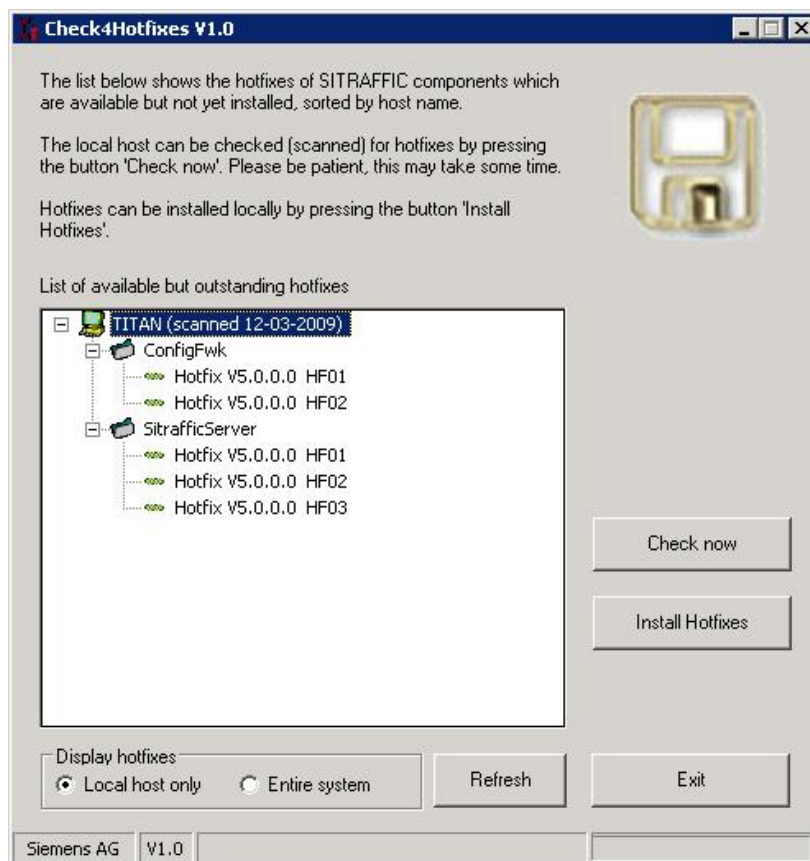


Figure 5-2: Hotfix check on current computer

5.2.3. In the entire Scala system

To display the hotfix status of the entire system, the display mode must be switched after running the tool Check4Hotfixes.exe (display hotfixes: Entire system) and button "*Refresh*" clicked. Now, all computers in the Scala system for which corresponding information is available are listed (for preconditions, see chapter 5.1).

Current information is provided most appropriately if the individual computers of the system are checked automatically (see chapter 5.3).

In display mode "*Entire system*", the hotfix check (scan) itself is available but the installation of the missing hotfixes is not.



Figure 5-3: Hotfix check in the Scala system

5.3. Set-up of automatic updating

The hotfix check (scan) can be automated on the individual computers of the Scala system. The starter **StartHotfixChecker.exe** installed locally (see chapter 5.2.1) has a button for this: "Automate Hotfix Check" which opens the following dialog.

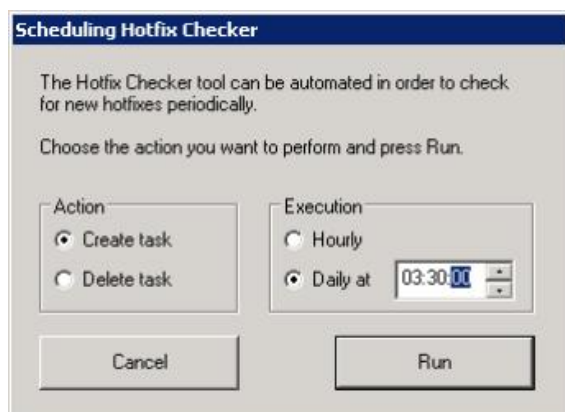


Figure 5-4: Automation of the hotfix check

Here, a system service can be set up that runs the tool Check4Hotfixes.exe at the set time (hourly or once a day) and makes the results of the check available on the Scala shared installation directory.

The service runs in the background and therefore is largely unseen. Only the start dialog of the hotfix deployer appears because it cannot be suppressed.

! Warning !:

To set up the service, the starter installed on the local computer (StartHotfixChecker.exe) must be called up because other files, that otherwise are no longer available, are to be set up here locally. In no case may the service be set up by being run from DVD4 (or the Scala shared installation directory)!