

opsi Version 4.0.2 (Part 2) Release Notes



uib gmbh
Bonifaziusplatz 1b
55118 Mainz
Tel.: +49 6131 275610
www.uib.de
info@uib.de

Contents

1	Copyright	1
2	Overview of New features	2
3	Installation Notes	4
4	Tool: The Management Interface <i>opsi-configd</i>	5
4.1	Selection of Clients	5
5	Extensions to the Driver Integration	7
5.1	Simplified Driver Integration during the Automatic Windows Installation	7
5.1.1	General Driver Packages	7
5.1.2	Preferred Drivers	8
5.1.3	Drivers that will be Manually Assigned to Computers	8
5.1.4	Drivers that will be Automatically Assigned to the Computers using the Fields <vendor>/<model>	8
5.1.5	Structure of the Driver Directory and Driver Files:	8
5.1.6	Processing of the Different Levels of Driver Integration	9
5.1.7	Driver Addition and Checking	9
6	Inventory	13
6.1	Hardware Inventory	13
6.2	Software Inventory	16
7	Changes to the New <i>opsi-linux-bootimage</i>	17
8	Changes to <i>opsiconfd</i>	18
9	<i>opsi</i> PAM Authentication	19
10	<i>/etc/opsi/opsi.conf</i>	20
11	Finding Silent or Unattended Switches	21
11.1	Script for Installations in the Context of a Local Administrator	22

12 opsi-winst 4.11.3	23
12.1 Start and Command Line Options	23
12.2 Global Text Constants	23
12.2.1 <i>opsi-winst</i> Path and Directory	23
12.3 Primary Sections	23
12.3.1 Reacting to Errors	23
12.3.2 Show Window Mode	24
12.3.3 Include Commands	24
12.3.3.1 Include Commands: Syntax	24
12.3.3.2 Include Commands: Library	25
12.3.4 String Handling	27
12.3.5 Other String Functions	30
12.3.6 Boolean Expressions	32
12.3.7 Stringlists from the Registry	32
12.3.8 Stringlists from the Product Properties	34
12.3.9 Other String Lists	34
12.3.10 Keeping Track of Failed Installations	35
12.3.11 Commands to Control the Log	35
12.4 Secondary Sections	35
12.4.1 Patches-Sections	35
12.4.1.1 Call Parameter	35
12.4.2 ExecWith Sections	35
12.4.3 WinBatch Sections	36
12.4.3.1 Call Parameter (Modifier)	36
12.5 Miscellaneous	38
13 Miscellaneous	39
13.1 Changelogs:	39
13.1.1 Changelog opsi-configed	39
13.1.2 Changelog opsi-winst	40
13.1.3 Changelog hwaudit	43
13.1.4 Changelog swaudit	44
13.1.5 Changelog jedit	44
13.1.6 Changelog opsi-adminutils	45
13.1.7 Changelog opsi-template	46
13.1.8 Changelog opsi-template-with-admin	46
13.1.9 Changelog opsi-client-agent	47
13.1.10 Changelog windows Netboot Products	47
13.1.11 Changelog python-opsi	48

13.1.12 Changelog opsipxeconfd	49
13.1.13 Changelog opsiconfd	49
13.1.14 Changelog opsi-utils	50
13.1.15 Changelog opsi-linux-bootimage	50
13.1.16 Changelog opsi-depotserver	50
13.1.17 Changelog opsi4ucs	51
13.1.18 Changelog opsi-atftp	51

Chapter 1

Copyright

The Copyright of this manual is held by uib gmbh in Mainz, Germany.

This manual is published under the creative commons license
Attribution - ShareAlike (by-sa).



A German description can be found here:

<http://creativecommons.org/licenses/by-sa/3.0/de/>

The legally binding German license can be found here:

<http://creativecommons.org/licenses/by-sa/3.0/de/legalcode>

The English description can be found here:

<http://creativecommons.org/licenses/by-sa/3.0/>

The English license can be found here:

<http://creativecommons.org/licenses/by-sa/3.0/legalcode>

Most parts of the opsi software are open source.

The parts of opsi that are not open source are still under cofunded development. Information about these parts can be found here:

http://uib.de/en/opsi_cofunding/index.html

All the open source code is published under the GPLv3:



The legally binding GPLv3 license can be found here: <http://www.gnu.org/licenses/gpl.html>

The name *opsi* is a registered trade mark of uib gmbh.

The opsi-logo is owned by uib gmbh, and may be used only with explicit permission.

Chapter 2

Overview of New features

The second part of the opsi Service Releases 4.0.2 has a wealth of new features and improvements. Here is an overview:

- opsi 4.0.2 works with the following Linux distributions:
 - Ubuntu 12.04
 - Redhat 6
 - CentOS 6
 - UCS 3.0 (with digitec-opsi-listener)
 - OpenSuse 12.1
 - OpenSuse 12.2
- A new preconfigured opsi-VM based on Ubuntu 12.04 with a a preconfigured opsi-Nagios-Connector.

The installation instructions for all supported Linux distributions and the VM can be found in the *opsi-getting-started* guide.

- opsi-configed
 - A new client-search functionality with highly advanced features
See below for more information
 - Search calls can be stored and executed multiple times
 - Directory-Tree in TreeView
Under *Directory*, a hierarchical tree can be created for client management. In contrast to *groups*, ever client must be in exactly one point under *Directory*.
 - The group description will be displayed as a tooltip.
 - Differences between the client default values and host-specific product property values are displayed in bold.
 - If the version on a client differs from that on the server, then the version information is highlighted in color.
 - *Product name* is now an additional column in the tabs *Product configuration* and *Netboot products*, and it is configurable.
 - Pressing *Re-initialize localboot products* in the context menu of the Client tab deletes all the information about installed products on this client.
 - The values of the *Report* column can be reset in the tabs *Product configuration* and *Netboot products*.
For example, the *Report* column can be manually reset after the status of *failed* has been reported.
 - Connection requests and sessions information requests are shifted in Threads, so that "hanging" connections are avoided in opsi-configed.
 - There is a check-box in the Web service login window that sets communications to gzip-compressed format.

- Hardware Inventory
 - The *Dell Express Code* is now determined for *Dell* computers.
 - The Hardware Inventory can be extended by calling special methods. See the description below.
- Driver Integration

The driver packages are automatically assigned to computers based on the data model from the vendor and the hardware inventory. See the descriptions below.
- There is a configuration file in `/etc/opsi/opsi.conf`, which controls Samba 4 access.
- Windows Netboot Products
 - Security fix for the WinPE version
 - Adapted for advanced driver integration
- opsi setup detector
- opsi-winst:

There are a variety of extensions, which are described below.

Chapter 3

Installation Notes

The products that are published as part of this release are dependent upon each other. You should not try to install a single part without the other parts.

No special procedures are required at installation time. The installation takes place during the server and opsi product updates.

The server should be updated before the opsi-Products. You can use the opsi-product-updater to update your opsi-Products:

```
opsi-product-updater -i -vv
```

If you have a multiple depots, then upgrade the server before upgrading the depots.

Chapter 4

Tool: The Management Interface *opsi-configed*

4.1 Selection of Clients

The client list can be one or more clients. This enables processing of multiple clients together.

The list of clients that are displayed in the clients tab can be reduced to showing only the selected ones by using the funnel icon or the menu item *Selection / Show only selected clients*.

The function *Save selected clients as group* has been abandoned. You can use the mouse and add the selected clients to an existing group by dragging them to an existing group which is displayed in the tree view.

In the client selection dialog (as called via menu *Selection / Define search*) clients can be selected using a variety of criteria based on their configurations.

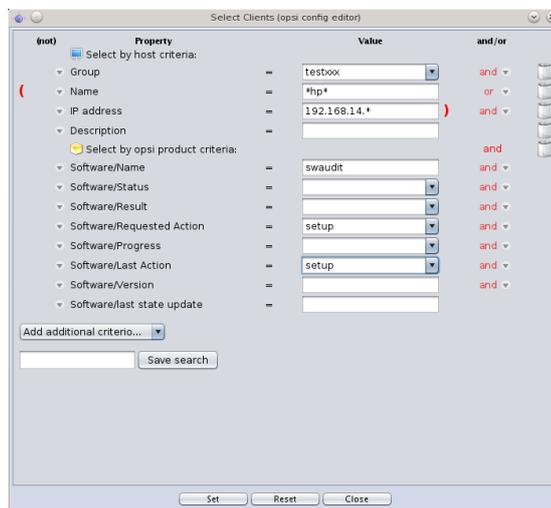


Figure 4.1: *opsi-configed*: Auswahldialog

E.g., it is possible to search for opsi installed products as well as software as found by the opsi software audit. You may as well search for PCs satisfying certain hardware conditions. Criteria may be combined by logical *and* or *or* operations and may be negated. Search strings can be given as fixed strings combined with asterisks *** as wildcard symbols.

Search definitions can be saved and then again used via the menu item *Selection/Use saved search definition*.

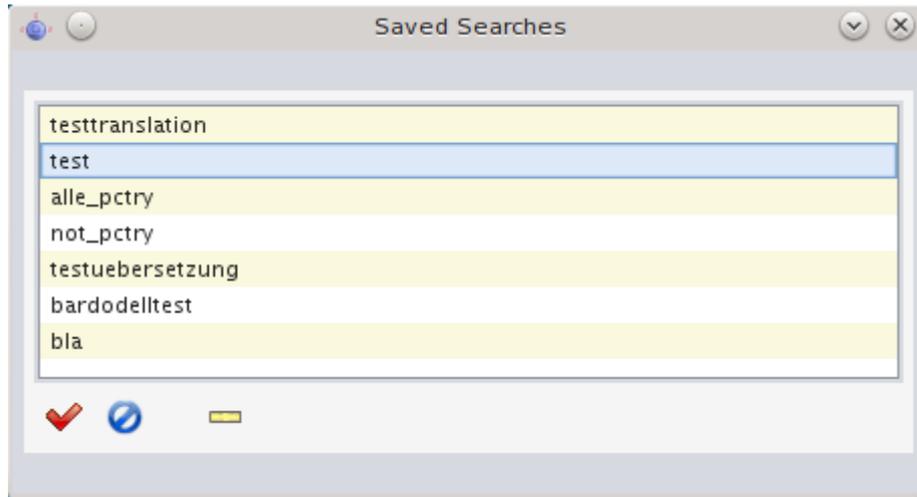


Figure 4.2: *opsi-configed*: Saved Search

It is also possible to run a saved search from the command line when the opsi-configed editor is started. By including the flag "-qs" and the name of the saved search, the configuration editor will start with the saved search results. If the name is omitted, then a list of available searches will be displayed.

Chapter 5

Extensions to the Driver Integration

The chapter on driver integration in *opsi-getting-started-manual* has been revised. More specifically, the chapter describes the new features for *Drivers that will be automatically assigned to computers given the inventory fields <vendor> / <model>*.

5.1 Simplified Driver Integration during the Automatic Windows Installation

If a group of computers have drivers that are not part of the Windows default installation, it is best to put these computers into a pool and integrate their drivers during installation time.

Opsi supports the automatic integration of drivers into the installation, and therefore simplifies driver deployment. In this case, the drivers simply need to be placed into the correct directory. When the installation script is called it searches through these directories and creates a catalog. The boot image automatically uses this catalog to embed the correct drivers. Opsis supports the automatic installation of standard drivers, USB drivers, HD audio drivers, and disk controller drivers (text-mode drivers).

In order for a driver to be immediately installed with the Windows installation, you must place the drivers on the server in a specific format. The drivers must be placed in the drivers directory, with the format **.inf*, where the file name describes the driver for the Windows setup program. The drivers in the *setup.exe* or **.zip* are not used here. If you have a computer that already has the drivers installed, then you can extract the appropriate drivers using the program *double driver* (<http://www.boozet.org/dd.htm>).

There are many levels of driver integration:

- General driver packages
- Preferred drivers that belong to your hardware, but are not assigned to specific computers
- Drivers that will be manually assigned to computers
- Drivers that will be automatically assigned to the computers using the fields <vendor>/<model>

Below is a detailed discussion about how to include each of these drivers

5.1.1 General Driver Packages

When the hardware configuration is very heterogeneous, then it may make sense to work with general driver packages. General drivers can be placed under *./drivers/drivers*.

You can find example general driver packages here <http://driverpacks.net/>.

Download the appropriate driver package to a temporary directory, and then unpack the driver package using the opsi script *extract_driver_pack.py* as such:

```
./extract_driver_pack.py <path to the temporary directory with the compressed driverpacks>
```

This will unpack and store the drivers in the directory `./drivers/drivers/`.

It may be the case that the drivers found by opsi in this location do not necessarily work with your hardware.

For the drivers which are found in `./drivers/drivers/`, the driver will be matched to the corresponding hardware using the PCI IDs (i.e. USB- or HD_Audio-ID) in the description file, and then integrated into the Windows setup as needed.

5.1.2 Preferred Drivers

In the case that you have to support special hardware, and you can find the additional drivers from the manufacturers, then use the following procedure to include them in the installation.

Place the additional drivers in their own directory under:

```
./drivers/drivers/preferred.
```

(the naming and depth of the directory structure is not important). Drivers that are found in the directory `./drivers/drivers/preferred` will be integrated into the Windows setup, assuming that opsi finds a suitable match to the drive hardware based off of the PCI IDs (i.e. USB or HD_Audio-ID) in the description file.

Problems can occur when the same PCI ID of the drivers is found in `preferred`. In this case, a direct mapping of the drivers to the devices is needed.

5.1.3 Drivers that will be Manually Assigned to Computers

When installing additional drivers based on the PCI-IDs or USB-IDs, they should be installed under the directory `./drivers/drivers/additional` (where name and depth of the directory structure is not important). You can map one or more drivers to a client using the Product-Property `additional_drivers` and a list of driver directories under `./drivers/drivers/additional`. The directories specified by `additional_drivers` are searched recursively until all drivers are found. This method can be used to make a specific directory based on the client type (i.e. dell-optiplex-815).

When a driver is found under the drivers directory that is specified by `additional_drivers` and also matches the PCI identifier, then other drivers in `drivers/preferred` or `drivers/` will not be used. Therefore the drivers under `additional_drivers` add functionality that would not have been found with the normal drivers. Also, the drivers that are manually bound to a client using `additional_drivers` receive priority over other drivers (`additional_drivers` can be thought of as *super-preferred*).

5.1.4 Drivers that will be Automatically Assigned to the Computers using the Fields <vendor>/<model>

The previously described mechanisms that directly map drivers to devices is automated since the 4.0.2 Release 2 of opsi. Opsi will search the directory `./drivers/drivers/additional/byAudit` for a director name that matches the field `Vendor` that was given in the Hardware Inventory. This `Vendor` directory will be search for a `Model` directory that corresponds to what is seen in Hardware Inventory. If this directory is found, then it will be manually assigned to the product property `additional_drivers`.

The directory name `byAudit` is case sensitive. The directory names for `Vendor` and `Model` are not case sensitive (`Dell` and `dELL` are treated the same way).

5.1.5 Structure of the Driver Directory and Driver Files:

```
/opt/  
  !-pcbin/  
    !-install/  
      !-winxppro/  
        !-drivers  
          |-classes/          (Links to driver device classes)  
          |-hdaudioids/      (Links to HD-Audio drivers)
```

pciids/	(Links to PCI-ID drivers)
pci.ids	(PCI database)
usbids/	(Links to USB-ID drivers)
usb.ids	(USB database)
vendors/	(Links to manufacturer drivers)
!-drivers	(place for general driver packages)
-additional/	(manually assigned drivers)
-byAudit/	Model-specific drivers that
-<vendor>	will be assigned by
-<model>	the Hardware Inventory
-buildin/	(data for the i386 version)
-preferred/	(certified drivers)
-exclude/	(excluded drivers)
!-mydriverpacks/	(example driver packages)

5.1.6 Processing of the Different Levels of Driver Integration

The top priority is given to drivers that are found using the property *additional_drivers* or using the inventory data in `./drivers/drivers/additional/byAudit`. Tests will be made during driver integration to determine which drivers are already installed on which hardware devices. When a driver is not found for a device, the following method will be used to search for drivers.

For devices that have drivers that were not listed in *additional_drivers*, opsi will search for, and integrate, an appropriate driver based off of the PCI ID (ie. USB-, HD_Audio-ID).

Integration of drivers means the following:

- The driver will be copied to the local hard drive at `c:\drv\<num>`.
- The Windows Setup will search for the driver sub-directory under `c:\drv\`, where the sub-directory name will be read from the unattended file in the root *unattend* sub-directory.

5.1.7 Driver Addition and Checking

After any changes in the directory `./drivers/drivers` have been made, call the following command from the Netboot Products root directory in order to set the permissions:

```
opsi-setup --set-rights ./drivers
```

Then call the script `./create_driver_links.py`. The script searches through the directories under `./drivers/drivers` and generates a list of links using PCI-IDs, USB-IDs, HD-Audio-IDs that enables hardware to know about drivers. The script will use the drivers in the preferred directories.

The script `setup.py` examines the hardware of the installed computers and identifies the necessary drivers. These will be copied to the disk and the file `unattended.txt` will be patched. The script `create_driver_links.py` examines the NT5 products one at a time in the *i386* tree and extracts the `.inf` files of the necessary drivers into *windows_builtin*. If you make a change to the *i386* directory tree (i.e. after installing a service pack), then delete that directory and run `create_driver_links.py` again. The recognized drivers for NT6 products are found in WinPE at *windows_builtin*.

With the following command, one can see the hardware inventory for a client:

```
./show_drivers.py <clientname>
```

When this command is called, it will show a selection for which drivers would be chosen for installation to the Bootimage via PCI-IDs, USB-IDs, HD-Audio-IDs, and *additional_drivers* (or *byAudit*), and which hardware still has no driver.

Use the output of `show_drivers.py` to check and see which drivers need to be installed.

It could be that manufacturers include different drivers for different operating systems (i.e. Vista vs. Win7) or different configurations (ie. SATA vs. SATA RAID). The `create_driver_links.py` cannot make this distinction. If

you think the wrong driver has been installed, then move the driver to the *drivers/exclude* directory and then call `create_driver_links.py` again. Drivers in the directory *drivers/exclude* are not used during the integration.

Example output of a `show_drivers.py` call:

```
./show_drivers.py pcdummy

PCI-Devices
[(Standardsystemgeräte), PCI Standard-PCI-zu-PCI-Brücke]
  No driver - device directory /opt/pcbin/install/winxppro/drivers/pciids/1022/9602 not found
[ATI Technologies Inc., Rage Fury Pro (Microsoft Corporation)]
  Using build-in windows driver
[(Standard-IDE-ATA/ATAPI-Controller), Standard-Zweikanal-PCI-IDE-Controller]
  /opt/pcbin/install/winxppro/drivers/drivers/D/M/N/123
[Realtek Semiconductor Corp., Realtek RTL8168C(P)/8111C(P) PCI-E Gigabit Ethernet NIC]
  /opt/pcbin/install/winxppro/drivers/drivers/preferred/realtek_gigabit_net_8111_8168b
[IEEE 1394 OHCI-konformer Hostcontroller-Hersteller, OHCI-konformer IEEE 1394-Hostcontroller]
  No driver - device directory '/opt/pcbin/install/winxppro/drivers/pciids/197B/2380' not found
[Advanced Micro Devices, Inc., AMD AHCI Compatible RAID Controller]
  /opt/pcbin/install/winxppro/drivers/drivers/preferred/ati_raid_sb7xx
[(Standard-USB-Hostcontroller), Standard OpenHCD USB-Hostcontroller]
  No driver - device directory '/opt/pcbin/install/winxppro/drivers/pciids/1002/4397' not found
[ATI Technologies Inc, ATI SMBus]
  /opt/pcbin/install/winxppro/drivers/drivers/preferred/ati_smbus

USB-Devices
[(Standard-USB-Hostcontroller), USB-Verbundgerät]
  /opt/pcbin/install/winxppro/drivers/drivers/preferred/brother_844x_pGerb
[Microsoft, USB-Druckerunterstützung]
  /opt/pcbin/install/winxppro/drivers/drivers/preferred/brother_844x_pGerb

Additional drivers
[ati_hdaudio_azalia]
  /opt/pcbin/install/winxppro/drivers/drivers/additional/ati_hdaudio_azalia
```

Example with *additional_drivers*:

```
./show_drivers.py e5800
Manually selected drivers (additional)
[hp_e5800]
[/opt/pcbin/install/win7-x64-professional-msdn/drivers/drivers/additional/hp_e5800/sp52852/Vista64/HDXHPAI3.inf]
[/opt/pcbin/install/win7-x64-professional-msdn/drivers/drivers/additional/hp_e5800/sp52852/Vista64/HDX861A.inf]
[/opt/pcbin/install/win7-x64-professional-msdn/drivers/drivers/additional/hp_e5800/sp52852/Vista64/HDXHPAI1.inf]
[/opt/pcbin/install/win7-x64-professional-msdn/drivers/drivers/additional/hp_e5800/sp52852/Vista64/HDXCPC.inf]
[/opt/pcbin/install/win7-x64-professional-msdn/drivers/drivers/additional/hp_e5800/sp52852/Vista64/HDXHPAI2.inf]
[/opt/pcbin/install/win7-x64-professional-msdn/drivers/drivers/additional/hp_e5800/sp50134/autorun.inf]
[/opt/pcbin/install/win7-x64-professional-msdn/drivers/drivers/additional/hp_e5800/sp50134/ibxHDMI/IntcDAud.inf]
[/opt/pcbin/install/win7-x64-professional-msdn/drivers/drivers/additional/hp_e5800/sp50134/HDMI/IntcHdmi.inf]
[/opt/pcbin/install/win7-x64-professional-msdn/drivers/drivers/additional/hp_e5800/sp50134/Graphics/kit24890.inf]
[/opt/pcbin/install/win7-x64-professional-msdn/drivers/drivers/additional/hp_e5800/sp50134/IIPS/Impcd.inf]
[/opt/pcbin/install/win7-x64-professional-msdn/drivers/drivers/additional/hp_e5800/sp54284/Realtek_64bit/hp64win7\
.inf]

PCI-Devices
[8086:27C8] Intel : Intel(R) N10/ICH7 Family USB Universal Host Controller - 27C8
  /opt/pcbin/install/win7-x64-professional-msdn/drivers/drivers/preferred/R293337/WIN7
[8086:27DA] Intel : Intel(R) N10/ICH7 Family SMBus Controller - 27DA
  /opt/pcbin/install/win7-x64-professional-msdn/drivers/drivers/preferred/R293337/WIN7
[8086:27C9] Intel : Intel(R) N10/ICH7 Family USB Universal Host Controller - 27C9
  /opt/pcbin/install/win7-x64-professional-msdn/drivers/drivers/preferred/R293337/WIN7
[8086:27DF] Intel : Intel(R) ICH7 Family Ultra ATA Storage Controllers - 27DF
  /opt/pcbin/install/win7-x64-professional-msdn/drivers/drivers/preferred/R293337/WIN7
[8086:27CA] Intel : Intel(R) N10/ICH7 Family USB Universal Host Controller - 27CA
  /opt/pcbin/install/win7-x64-professional-msdn/drivers/drivers/preferred/R293337/WIN7
[8086:2E30] Intel : Intel(R) 4 Series Chipset Processor to I/O Controller - 2E30
  /opt/pcbin/install/win7-x64-professional-msdn/drivers/drivers/not_preferred/x64/C/Intel/1
[8086:27CB] Intel : Intel(R) N10/ICH7 Family USB Universal Host Controller - 27CB
  /opt/pcbin/install/win7-x64-professional-msdn/drivers/drivers/preferred/R293337/WIN7
```

```
[8086:2E32] Intel Corporation : Intel(R) G41 Express Chipset
  Manually selected [hp_e5800] /opt/pcbin/install/win7-x64-professional-msdn/drivers/drivers/additional/hp_e5800/\
  sp50134/Graphics
[8086:27CC] Intel : Intel(R) N10/ICH7 Family USB2 Enhanced Host Controller - 27CC
  /opt/pcbin/install/win7-x64-professional-msdn/drivers/drivers/preferred/R293337/WIN7
[8086:244E] Intel : Intel(R) 82801 PCI-Brücke - 244E
  Using build-in windows driver
  This driver will not be integrated, because same device already integrated in: '/opt/pcbin/install/win7-x64-\
  professional-msdn/drivers/drivers/not_preferred/x64/C/Intel/1/dmi_pci.inf'
[8086:27D0] Intel : Intel(R) N10/ICH7 Family PCI Express Root Port - 27D0
  /opt/pcbin/install/win7-x64-professional-msdn/drivers/drivers/preferred/R293337/WIN7
[8086:27B8] Intel : Intel(R) ICH7 Family LPC Interface Controller - 27B8
  /opt/pcbin/install/win7-x64-professional-msdn/drivers/drivers/preferred/R293337/WIN7
[8086:27D2] Intel : Intel(R) N10/ICH7 Family PCI Express Root Port - 27D2
  /opt/pcbin/install/win7-x64-professional-msdn/drivers/drivers/preferred/R293337/WIN7
[8086:27C0] Intel : Intel(R) N10/ICH7 Family Serial ATA Storage Controller - 27C0
  /opt/pcbin/install/win7-x64-professional-msdn/drivers/drivers/preferred/R293337/WIN7
[8086:27D8] Microsoft : High Definition Audio-Controller
  No driver - device directory '/opt/pcbin/install/win7-x64-professional-msdn/drivers/pciids/8086/27D8' not found
[10EC:8136] Realtek : Realtek RTL8102E/RTL8103E-Familie-PCI-E-Fast-Ethernet-NIC (NDIS 6.20)
  Manually selected [hp_e5800] /opt/pcbin/install/win7-x64-professional-msdn/drivers/drivers/additional/hp_e5800/\
  sp54284/Realtek 64bit
```

USB-Devices

```
[0461:0010] (Standardsystemgeräte) : USB-Eingabegerät
  No driver - vendor directory '/opt/pcbin/install/win7-x64-professional-msdn/drivers/usbids/0461' not found
[0461:4D20] (Standardsystemgeräte) : USB-Eingabegerät
  No driver - vendor directory '/opt/pcbin/install/win7-x64-professional-msdn/drivers/usbids/0461' not found
[058F:6366] Kompatibles USB-Speichergerät : USB-Massenspeichergerät
  No driver - vendor directory '/opt/pcbin/install/win7-x64-professional-msdn/drivers/usbids/058F' not found
[0461:0010] (Standard-USB-Hostcontroller) : USB-Verbundgerät
  No driver - vendor directory '/opt/pcbin/install/win7-x64-professional-msdn/drivers/usbids/0461' not found
```

HD-Audio-Devices

```
[10EC:0662] Realtek High Definition Audio
  Manually selected [hp_e5800] /opt/pcbin/install/win7-x64-professional-msdn/drivers/drivers/additional/hp_e5800/\
  sp52852/Vista64
```

Example with *byAudit*:

```
./show_drivers.py pctry5detlef
Manually selected drivers (additional)
[/opt/pcbin/install/win7-x64-professional-msdn/drivers/drivers/additional/byAudit/nvidia/awrdacpi]
[/opt/pcbin/install/win7-x64-professional-msdn/drivers/drivers/additional/byAudit/nvidia/awrdacpi/pctry5detlef/\
Display/Radeon X300-X550-X1050 Series Secondary (Microsoft Corporation - WDDM)/atiilhag.inf]
[/opt/pcbin/install/win7-x64-professional-msdn/drivers/drivers/additional/byAudit/nvidia/awrdacpi/pctry5detlef/\
Display/Radeon X300-X550-X1050 Series (Microsoft Corporation - WDDM)/atiilhag.inf]
[/opt/pcbin/install/win7-x64-professional-msdn/drivers/drivers/additional/byAudit/nvidia/awrdacpi/pctry5detlef/\
MEDIA/Realtek AC'97 Audio/oem21.inf]

PCI-Devices
[1002:5B70] ATI Technologies Inc. : Radeon X300/X550/X1050 Series Secondary (Microsoft Corporation - WDDM)
  Manually selected [/opt/pcbin/install/win7-x64-professional-msdn/drivers/drivers/additional/byAudit/nvidia/\
awrdacpi] /opt/pcbin/install/win7-x64-professional-msdn/drivers/drivers/additional/byAudit/nvidia/awrdacpi/\
pctry5detlef/Display/Radeon X300-X550-X1050 Series Secondary (Microsoft Corporation - WDDM)
  Multiple selected [/opt/pcbin/install/win7-x64-professional-msdn/drivers/drivers/additional/byAudit/nvidia/\
awrdacpi] /opt/pcbin/install/win7-x64-professional-msdn/drivers/drivers/additional/byAudit/nvidia/awrdacpi/\
pctry5detlef/Display/Radeon X300-X550-X1050 Series (Microsoft Corporation - WDDM)
[10BE:0053] (Standard-IDE-ATA/ATAPI-Controller) : Standard-Zweikanal-PCI-IDE-Controller
  No driver - device directory '/opt/pcbin/install/win7-x64-professional-msdn/drivers/pciids/10BE/0053' not found
[10DE:005D] (Standardsystemgeräte) : PCI Standard-PCI-zu-PCI-Brücke
  No driver - device directory '/opt/pcbin/install/win7-x64-professional-msdn/drivers/pciids/10DE/005D' not found
[1022:1100] AMD : AMD HyperTransport(tm)-Konfiguration
  Using build-in windows driver
[10DE:0054] (Standard-IDE-ATA/ATAPI-Controller) : Standard-Zweikanal-PCI-IDE-Controller
  /opt/pcbin/install/win7-x64-professional-msdn/drivers/drivers/preferred/evb_potsdam_fsc__esprimo_p625/\
FTS_NVIDIASATAAHCIDRIVERVISTA64V103042MCP78__1026963/NVIDIA_SATA_AHCI_DRIVER_Vista64_V10.3.0.42_MCP78 (textmode \
capable)
```

```
[1022:1101] AMD : AMD-Adresszuordnungskonfiguration
Using build-in windows driver
[10DE:0055] (Standard-IDE-ATA/ATAPI-Controller) : Standard-Zweikanal-PCI-IDE-Controller
/opt/pcbin/install/win7-x64-professional-msdn/drivers/drivers/preferred/evb_potsdam_fsc__esprimo_p625/\
FTS_NVIDIASATAAHCIDRIVERVISTA64V103042MCP78_1026963/NVIDIA_SATA_AHCI_DRIVER_Vista64_V10.3.0.42_MCP78 (textmode \
capable)
[1022:1102] AMD : AMD DRAM und HyperTransport(tm)-Nachverfolgungsmoduskonfiguration
Using build-in windows driver
[10DE:0057] NVIDIA : NVIDIA nForce-Netzwerkcontroller
Using build-in windows driver
[1022:1103] AMD : Sonstige AMD-Konfiguration
Using build-in windows driver
[10DE:0059] Realtek : Realtek AC'97 Audio
Manually selected [/opt/pcbin/install/win7-x64-professional-msdn/drivers/drivers/additional/byAudit/nvidia/\
awrdacpi] /opt/pcbin/install/win7-x64-professional-msdn/drivers/drivers/additional/byAudit/nvidia/awrdacpi/\
pctry5detlef/MEDIA/Realtek AC'97 Audio
[10DE:005E] NVIDIA : NVIDIA nForce4 HyperTransport-Brücke
/opt/pcbin/install/win7-x64-professional-msdn/drivers/drivers/preferred/ga-ma78-pcbon4/chipset_win7-64/SMBUS
[104C:8025] Texas Instruments : OHCI-konformer Texas Instruments 1394-Hostcontroller
No driver - device directory '/opt/pcbin/install/win7-x64-professional-msdn/drivers/pciids/104C/8025' not found
[10DE:005A] (Standard-USB-Hostcontroller) : Standard OpenHCD USB-Hostcontroller
No driver - device directory '/opt/pcbin/install/win7-x64-professional-msdn/drivers/pciids/10DE/005A' not found
[10DE:0050] (Standardsystemgeräte) : PCI Standard-ISA-Brücke
No driver - device directory '/opt/pcbin/install/win7-x64-professional-msdn/drivers/pciids/10DE/0050' not found
[10DE:005B] (Standard-USB-Hostcontroller) : Standard PCI-zu-USB erweiterter Hostcontroller
No driver - device directory '/opt/pcbin/install/win7-x64-professional-msdn/drivers/pciids/10DE/005B' not found
[1002:5B60] ATI Technologies Inc. : Radeon X300/X550/X1050 Series (Microsoft Corporation - WDDM)
Manually selected [/opt/pcbin/install/win7-x64-professional-msdn/drivers/drivers/additional/byAudit/nvidia/\
awrdacpi] /opt/pcbin/install/win7-x64-professional-msdn/drivers/drivers/additional/byAudit/nvidia/awrdacpi/\
pctry5detlef/Display/Radeon X300-X550-X1050 Series Secondary (Microsoft Corporation - WDDM)
Multiple selected [/opt/pcbin/install/win7-x64-professional-msdn/drivers/drivers/additional/byAudit/nvidia/\
awrdacpi] /opt/pcbin/install/win7-x64-professional-msdn/drivers/drivers/additional/byAudit/nvidia/awrdacpi/\
pctry5detlef/Display/Radeon X300-X550-X1050 Series (Microsoft Corporation - WDDM)
[10DE:0052] NVIDIA : NVIDIA nForce PCI-Systemverwaltung
Using build-in windows driver
[10DE:005C] (Standardsystemgeräte) : PCI Standard-PCI-zu-PCI-Brücke
No driver - device directory '/opt/pcbin/install/win7-x64-professional-msdn/drivers/pciids/10DE/005C' not found

USB-Devices
[1241:1111] (Standardsystemgeräte) : USB-Eingabegerät
No driver - vendor directory '/opt/pcbin/install/win7-x64-professional-msdn/drivers/usbids/1241' not found

HD-Audio-Devices
No devices installed
```

TIPS

- Directory names NDIS1 are Vista-Drivers ; NDIS2 are Win7-Driver
- Some chip driver contain description files, which perform a lot without actually providing drivers. An example would be the *cougar.inf* from Intel. If such a *Pseudo-Driver* were to be placed in *additional_drivers* (or *byAudit*), then the other drivers may be excluded. It is better to move these directories to the *preferred* subdirectory.
- SATA drivers and SATA-RAID drivers refer to the same PCI ID. A SATA-RAID driver will not function with a single-disk system.
- Check the output of `./show_drivers` carefully !

Chapter 6

Inventory

The inventory can be ordered with the Localboot products `hwaudit` and `swaudit` or with the Netboot product `hwinvent`.

6.1 Hardware Inventory

The hardware inventory is controlled by an opsi configuration file. This means that the information about which data will be compiled are not hardwired into the corresponding products `hwaudit` and `hwinvent`. In fact, the products will be controlled by a configuration file. The configuration file will be called and interpreted with every dispatch of the Web service. Simultaneously, the configuration file controls the structure of the database, so that a change of this configuration file changes the database schema.

The configuration file is `/etc/opsi/hwaudit/opsihwaudit.conf`.

All the inventoried objects are defined and described in this file, like how these objects and their data are instantiated (under Linux and Windows). This file will also define the associated data structure. To be more specific, this configuration file contains the object-oriented inheritance definitions. The reason for this is the fact that a lot of objects contain identical data fields (i.e. like `Name` and `Vendor`). The general information will be defined in *virtual* Hardware base classes. The actual inventory objects are then *structural* Hardware classes, where many properties could possibly be inherited from overridden *virtual* base classes.

The following example may be instructive:

At first, the configuration file defines a *virtual Class* called `"BASIC_INFO"`. This defines the properties (*Values*):

- "name"
- "description"

Next comes the *virtual Class* called `"HARDWARE_DEVICE"`, which inherits all the additional parameters from `"BASIC_INFO"`, and includes the following:

- "vendor"
- "model"
- "serialNumber"

Next follows the first object that is found in the inventory, which is the first *structural Class* called `"COMPUTER_SYSTEM"`, which inherits of all the additional parameters from `"HARDWARE_DEVICE"`, it is defined (and overwrites properties) as:

- "name"

- "systemType"
- "totalPhysicalMemory"

The class definition will include a description of various parameters and their *Values*:

- Class definition:
 - "Type"
is "STRUCTURAL" or "VIRTUAL"
 - "Super"
this class which it will be inheriting.
 - "Opsi"
gives the name of the class, which will be used later in opsi as a display name.

Further more, the class definition can define how the data will be compiled. This information can also be found in the definition of the *Values*.

- For the inventory under Linux:
 - "Linux": "[<command>]<parameter>"
Executes the command <command> on the command line, with the argument <parameter>.
 - "Python": "<python code with place holder>"
Executes the given Python code whose output will be placed in the place holder which is between the "#" signs (see example below).
- For the Inventory under Windows:
 - "WMI": "<wmi select statement>"
executes WMI when called
 - "Cmd": "<Python text object with place holder>"
In this case, this is the relative path to the Python executable program, whose output will be placed in the place holder.
 - "Registry": "[<registry key>] <value name>"
The value of <value name> will be read from the registry, and given the key name <registry key>.
The registry must be read in an architecture-specific manner. This means, that the 64 bit sector will be read on a 64 bit system.
- Value Definition:
 - "Type": "<MySQL database type>"
<MySQL Database type> gives the MySQL database type that will be applied to this value (i.e. a Python string will be a "<MySQL Database type>="varchar(200)").
 - "Scope": "<scope>"
The field <scope> will be used in the following way:
"g" means: This attribute is the same in every link of these types.
"i" means: This attribute can have different types of values with these links.
 - "Opsi": "<id>"
"<id>" is an internal name of the fields. This can be found in the file located at /etc/opsi/hwaudit/locales .
 - "WMI": "<id or command>"
<id or command> is either the name of a WMI command that prints the value or a single WMI command. If the WMI command is given in the Class definition (i.e. "select * from Win32_ComputerSystem") , then the results are assigned to the "WMI" variables in the "Values" class definition. If there is no WMI command, then the "WMI" variables in the "Values" section are WMI commands (see example below).

- "Linux": "<id>"
This is part of the class definition, <id> is the name of the displayed value when the Linux command is given.
- "Condition": "<condition>"
<condition> is a condition which must be fulfilled, with which the *Value* will be determined. So for example if the <condition> is defined as "vendor=[dD]ell*", then the values of "vendor" must contain either *Dell* or *dell*.

Here is an example of the class "COMPUTER_SYSTEM":

```
{
  "Class": {
    "Type": "STRUCTURAL",
    "Super": [ "HARDWARE_DEVICE" ],
    "Opsi": "COMPUTER_SYSTEM",
    "WMI": "select * from Win32_ComputerSystem",
    "Linux": "[lshw]system"
  },
  "Values": [
    {
      "Type": "varchar(100)",
      "Scope": "i",
      "Opsi": "name",
      "WMI": "Name",
      "Linux": "id"
    },
    {
      "Type": "varchar(50)",
      "Scope": "i",
      "Opsi": "systemType",
      "WMI": "SystemType",
      "Linux": "configuration/chassis"
    },
    {
      "Type": "bigint",
      "Scope": "i",
      "Opsi": "totalPhysicalMemory",
      "WMI": "TotalPhysicalMemory",
      "Linux": "core/memory/size",
      "Unit": "Byte"
    },
    {
      "Type": "varchar(50)",
      "Scope": "i",
      "Opsi": "dellexpresscode",
      "Condition": "vendor=[dD]ell*",
      "Cmd": "#dellexpresscode\dellexpresscode.exe#.split('=')[1]",
      "Python": "str(int(#{'COMPUTER_SYSTEM': 'serialNumber', 'CHASSIS': 'serialNumber'}#),36))"
    }
  ]
},
```

Regarding the "WMI" commands, the class definition contains "select * from Win32_ComputerSystem". This command is run by WMI, which has output columns of "Name", "SystemType", and "TotalPhysicalMemory". These values are then assigned to the opsi values of "name", "systemType", and "totalPhysicalMemory".

Especially interesting is here the last value "dellexpresscode":

This is really useful when it questions a Dell-computer, about its condition.

The command line program `dellexpresscode.exe` was designed for Windows, and tells `hwaudit.exe` that the `dellexpresscode` is provided in the directory `dellexpresscode\`. Items in between `#` are place holders for output. So the statement at "`dellexpresscode\dellexpresscode.exe`" runs `dellexpresscode.exe`, and produces output in the form: `dellexpresscode=123456789`. The value that will be used is the one after the split on the place holder =, which is done in Python using the `split()` method as such `.split('=')[1]`. Under Linux, there will be found a value for *serialNumber* for the elements (*COMPUTER_SYSTEM* or *CHASSIS*), that is then used to assign the Dell Express codes. The call `int(,36)` converts the output integer to base-36.

The OPSI names of the values will be translated using the files found in `/etc/opsi/hwaudit/locales/*`. The file `/etc/opsi/hwaudit/locales/en_US` may contain translations such as:

```
COMPUTER_SYSTEM = Computer  
COMPUTER_SYSTEM.systemType = Type
```

The class name `COMPUTER_SYSTEM` will be translated into "Computer". The Opsi attribute "systemType" of the class `COMPUTER_SYSTEM` will be translated into "Type" for English. If one were to look in the file `/etc/opsi/hwaudit/locales/de_DE`, one would see that the attribute of "`COMPUTER_SYSTEM.systemType`" will be translated into "Typ" for German. Finally another suggestion: When a new field is created, it should be placed in these files, even if one does not translate the term explicitly. This avoids any "Warning" messages.

6.2 Software Inventory

The software inventory is done with the Localboot product `swaudit`. In this case, information will be inherited from the uninstall of the Registry, and additional information will be obtained from the Hotfixes and License keys.

The source code for these packets can be found here:

<https://svn.opsi.org/listing.php?repname=swaudit>

Chapter 7

Changes to the New opsi-linux-bootimage

The *opsi-linux-bootimage* (version 20120728-1) works internally with *dhclient* instead of *pump* for the initial dhcp-request.

The *pump* program was being used in the previous version because *dhclient* does not process the *next-server* flag. This flag was used by opsi to define the opsiserver-address. It is not possible to configure the *pump* utility, and it gives a "hardreset" on the network card to actualize the ip-address from the client. This causes many problems in different network-infrastructures. For this reason we have decided to use *dhclient* instead of *pump*.

This modification causes the problem that the *opsi-clientbootcd* asks for the *opsi-server* even if you choose "with DHCP" on the startmenu.

Chapter 8

Changes to `opsiconfd`

The opsi webservice will now build as "noarch". For some rpm-based distributions it was built for a specific architecture. But that is no longer necessary, because there are no binaries in this package.

This could produce some problems for certain distributions when you try to update these packages. If you get an error like this: "`opsiconfd-4.0.2.3-1.1.i386.rpm` couldn't be found" , then you must reinstall the `opsiconfd`-package manually from repository.

The init-scripts were modified as well. The runlevel 4 will not be used on rpm-based distributions.

Chapter 9

opsi PAM Authentication

opsi uses some PAM-modules to authenticate the user. With this new release, opsi uses different modules for certain distributions. The following list will give you a small overview about which modules are used:

Default: common-auth

SLES: sshd

CentOS and RedHat: system-auth

RedHat 6: password-auth

You can see in the list above, which different PAM-modules are used. However, it could be the case that a different PAM-module is required, depending on the local configuration. The source code can be modified to account for these changes. To provide more flexibility without changing the sourcecode, it is possible with this release to create a file named: `opsi-auth` in the directory `/etc/pam.d/`. If this file exists, opsi will use this configuration automatically instead of the modules listed above.

The following example will show you the new feature: If you run a debian/ubuntu-System and you get a PAM-authentication-error, even though you can connect with the same credentials over ssh on the server, then you can create the file: `/etc/pam.d/opsi-auth` with following content:

```
@include sshd
```

After restarting `opsiconfd`, opsi will use automatically the `sshd-PAM`-module for authentication of users.

Chapter 10

/etc/opsi/opsi.conf

Since Version 4.0.2-2

General opsi configurations.

Example:

```
[groups]
fileadminigroup = pcpatch
```

Background: The classical installation with the user: *pcpatch* with the primary group: *pcpatch* does not function with Samba 4. Samba 4 has fundamental restrictions on Active-Directory, such that groups with the same name as the user (as is common in Unix/Linux) are no longer allowed. For this reason, a new configuration file has been introduced: */etc/opsi/opsi.conf*, which will control how the groups will access Samba-Access. So for example, UCS 3 will use this file to rename the group *pcpatch* to **opsifileadmins**. That means that the user who belonged to the group *pcpatch* under UCS 3, must now belong to the group *opsifileadmins*. This special feature is only for UCS 3 and is not the same for other distributions (so long as they are using Samba 4, or have Samba 4 installed).

Chapter 11

Finding Silent or Unattended Switches

In „Unattended" or „silent" mode, command line arguments are called during non-interactive installation, and these command line arguments can be found in the original setup program.

The problem is then to find the appropriate command line arguments in the original setup program.

Identify the manufacturer of the setup program: Most setup programs are built using frameworks like *Inno*, *NSIS*, *Installshield* or *Wise*. Each one of these setup frameworks has their own switch. The following method can be used to determine the framework and other necessary information: The input strings can be determined using the command line program *strings* given the setup program *setup.exe*, and the output framework names can be found using *grep* or *findstr*.

The same method is used in the `opsi-setup-detector`. See the example below:

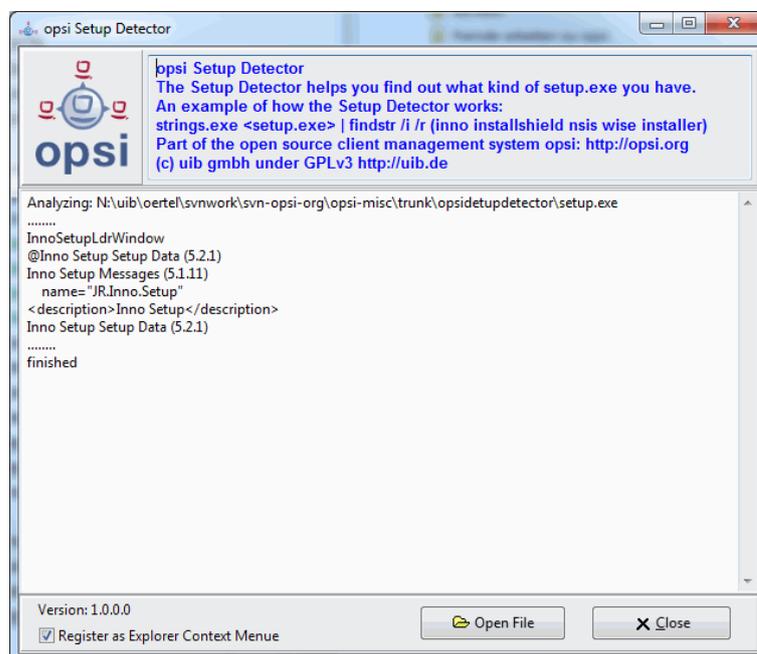


Figure 11.1: opsi setup detector

This GUI program can be called from the Windows context menu Explore.

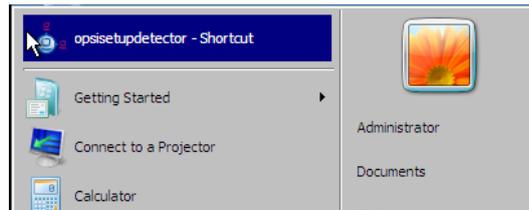


Figure 11.2: opsi setup detector in Windows Explore context menu

The *opsi setup detector* is part of the *opsi-adminutils* package. The program can be downloaded as a stand-alone application from:

<http://download.uib.de/opsi4.0/helper/opsissetupdetector.exe>

11.1 Script for Installations in the Context of a Local Administrator

The product `opsi-template-with-admin` was completely revised:

- The external `psgetsid.exe` is now not needed any more.
- At the master script there are only three variables to change which are marked with `Please edit the following values`.
- The local script has now a full access to the opsi-server. This means, that functions like `GetProductProperty` or `OpsiServiceCall` or the access to the license management will work like normal.
- Also the function `isFatalError` works now how you would it expect.
- Locking Keyboard and Mouse and the `LogLevel` are now controled by a product property.
- The product requires `opsi-winst >= 4.11.3.3`

Chapter 12

opsi-winst 4.11.3

12.1 Start and Command Line Options

Since version 4.11.3, the *opsi-winst* program contains a manifest with the statement: `<requestedExecutionLevel level="highestAvailable" />`. This means that if *opsi-winst* is called on an NT6 OS by an Administrator, then it will run as an *elevated* process. If *opsi-winst* is called with normal user privileges, then it will run with the privileges of this user.

12.2 Global Text Constants

12.2.1 *opsi-winst* Path and Directory

`%opsiScriptHelperPath%`

Corresponds to: `%ProgramFiles32Dir%\opsi.org\opsiScriptHelper`

Path in which the help program, libraries, and items needed for script execution could be installed.

Since 4.11.3.2

12.3 Primary Sections

12.3.1 Reacting to Errors

- `FatalOnSyntaxError` = `<boolean value>`

- *true* = (default) If a syntax error occurs, then the script execution will be stopped and the script result will be set to *failed*. Also, the message *Syntax Error* will be passed to the opsi-server.
- *false* = If a syntax error occurs, then the script execution will not be stopped and the script result will be set to *success*.

In either case above, the syntax error will be logged as *Critical*.

In either case above, the error counter will be increased by 1.

Since 4.11.3.2

In older versions there was no logging of syntax errors, no increase of error counter, and the result was always set to *success*.

12.3.2 Show Window Mode

To change the modes of how the *opsi-winst* window is displayed, use these commands:

- **NormalizeWinst**
sets the *opsi-winst* window to the *normal* mode
- **IconizeWinst**
sets the *opsi-winst* window to the *minimized* mode
- **RestoreWinst**
sets the *opsi-winst* window to the *maximized* mode

12.3.3 Include Commands



Caution

Using include commands can lead to confusing code. Use with caution if you are a beginner.

12.3.3.1 Include Commands: Syntax

The `include_*` commands may be used to add external script files to the currently-running script at runtime. The `include_*` command can include external script files either as an insert (after the current line) or as an append (after the last line). The include commands may be used anywhere in a primary section. The external script files may contain their own include commands.

The include commands are available since version 4.11.3

- **include_insert** <file name>
inserts <file name> after the current line into the running script. So the first line of the included file is the next line that will be executed by *opsi-winst*.
- **include_append** <file name>
appends the content of <file name> to the running script. This kind of insert is normally used to include sections from a library.

In both cases <file name> is:

- A complete path to an existing file.
- A existing file in `%ScriptPath%`
- A file in `%opsiScriptHelperPath%\lib`
Is equivalent to: `%ProgramFiles32Dir%\opsi.org\opsiScriptHelper\lib`
- A existing file in `%WinstDir%\lib`

The tests for the location of the <file name> are done in the order above. *opsi-winst* uses the first file it finds that has a matching name.

Example:

When we run that contains the following commands:

```
[Actions]
include_append "section_Files_del_tmp_dummy.opsiinc"
include_insert "include-test1.opsiinc"
```

The file `include-test1.opsiinc` is run first. The contents of the included file `include-test1.opsiinc` are:

```
DefVar $inctestvar$
set $inctestvar$ = "inctest"
Files_del_tmp_dummy
include_append "section_Files_copy_inctest.opsiinc"
Files_copy_inctest

if fileExists("c:\opsi.org\tmp\dummy.txt")
    comment "passed"
else
    comment "failed"
    set $TestResult$ = "not o.k."
    LogWarning "include test failed"
endif

if fileExists("%scriptpath%\test-files\dummy.txt")
    comment "passed"
else
    comment "failed"
    set $TestResult$ = "not o.k."
    LogWarning "include test failed"
endif
Files_del_tmp_dummy
```

The contents of the included file `section_Files_copy_inctest.opsiinc` are:

```
[Files_copy_inctest]
copy "%scriptpath%\test-files\dummy.txt" "c:\opsi.org\tmp"
```

Since the call to `Files_del_tmp_dummy` happens inside of `include-test1.opsiinc` without `section_Files_del_tmp_dummy.opsiinc` being referenced inside of `include-test1.opsiinc`, we must call `include_append "section_Files_del_tmp_dummy.opsiinc"` at the very beginning of our script. Otherwise, `opsi-winst` will report that `Files_del_tmp_dummy` is not defined.

The contents of the included file `section_Files_del_tmp_dummy.opsiinc` are:

```
[Files_del_tmp_dummy]
del -f "c:\opsi.org\tmp\dummyt.txt"
```

12.3.3.2 Include Commands: Library

The following include files are shipped with version 4.11.3, and are located in `%WinstDir%\lib`:

`insert_check_exit_code.opsiinc`:

```
; opsi include file

DefVar $ExitCode$

include_append "section_sub_check_exitcode.opsiinc"
```

`insert_get_licensekey.opsiinc`:

```
; opsi include file

DefVar $LicenseRequired$
DefVar $LicenseKey$
```

```
DefVar $LicensePool$
```

```
include_append "section_sub_get_licensekey.opsiinc"
```

```
section_sub_check_exit_code.opsiinc:
```

```
;opsi include file
```

```
[Sub_check_exitcode]
```

```
comment "Test for installation success via exit code"
```

```
set $ExitCode$ = getLastExitCode
```

```
; informations to exit codes see
```

```
; http://msdn.microsoft.com/en-us/library/aa372835(VS.85).aspx
```

```
; http://msdn.microsoft.com/en-us/library/aa368542.aspx
```

```
if ($ExitCode$ = "0")
```

```
    comment "Looks good: setup program gives exitcode zero"
```

```
else
```

```
    comment "Setup program gives a exitcode unequal zero: " + $ExitCode$
```

```
    if ($ExitCode$ = "1605")
```

```
        comment "ERROR_UNKNOWN_PRODUCT 1605 This action is only valid for products  
that are currently installed."
```

```
        comment "Uninstall of a not installed product failed - no problem"
```

```
    else
```

```
        if ($ExitCode$ = "1641")
```

```
            comment "looks good: setup program gives exitcode 1641"
```

```
            comment "ERROR_SUCCESS_REBOOT_INITIATED 1641 The installer has  
initiated a restart. This message is indicative of a success."
```

```
            ExitWindows /Reboot
```

```
        else
```

```
            if ($ExitCode$ = "3010")
```

```
                comment "looks good: setup program gives exitcode 3010"
```

```
                comment "ERROR_SUCCESS_REBOOT_REQUIRED 3010 A restart is  
required to complete the install. This message is indicative of a success."
```

```
                ExitWindows /Reboot
```

```
            else
```

```
                logError "Fatal: Setup program gives an unknown exitcode unequal  
zero: " + $ExitCode$
```

```
                isFatalError "Exit Code: " + $ExitCode$
```

```
            endif
```

```
        endif
```

```
    endif
```

```
endif
```

```
section_sub_get_licensekey.opsiinc:
```

```
; opsi include file
```

```
[Sub_get_licensekey]
```

```
if opsiLicenseManagementEnabled
```

```
    comment "License management is enabled and will be used"
```

```
    comment "Trying to get a license key"
```

```
    Set $LicenseKey$ = demandLicenseKey ($LicensePool$)
```

```
    ; If there is an assignment of exactly one license pool to the product the following call  
is possible:
```

```
    ; Set $LicenseKey$ = demandLicenseKey ("", $ProductId$)
```

```
    ;
```

```
    ; If there is an assignment of a license pool to a windows software id, it is possible to
    use:
    ; DefVar $WindowsSoftwareId$
    ; $WindowsSoftwareId$ = "...
    ; Set $LicenseKey$ = demandLicenseKey ("", "", $WindowsSoftwareId$)

    DefVar $ServiceErrorClass$
    set $ServiceErrorClass$ = getLastServiceErrorClass
    comment "Error class: " + $ServiceErrorClass$

    if $ServiceErrorClass$ = "None"
        comment "Everything fine, we got the license key ' " + $LicenseKey$ + "'
    else
        if $ServiceErrorClass$ = "LicenseConfigurationError"
            LogError "Fatal: license configuration must be corrected"
            LogError getLastServiceErrorMessage
            isFatalError $ServiceErrorClass$

        else
            if $ServiceErrorClass$ = "LicenseMissingError"
                LogError "Fatal: required license is not supplied"
                isFatalError $ServiceErrorClass$
            endif
        endif
    endif
endif
else
    LogError "Fatal: license required, but license management not enabled"
    isFatalError "No Licensemanagement"
endif
```

12.3.4 String Handling

- `contains(<str>, <substr>)`

A boolean function which returns *true* if <str> contains <substr>. This function is case sensitive.

Available since 4.11.3

Example:

```
set $ConstTest$ = "1xy451Xy451XY45"
set $CompValue$ ="xy"
if contains($ConstTest$, $CompValue$)
    comment "passed"
else
    set $TestResult$ = "not o.k."
    LogWarning "failed"
endif
set $CompValue$ ="xY"
if not(contains($ConstTest$, $CompValue$))
    comment "passed"
else
    set $TestResult$ = "not o.k."
    LogWarning "failed"
endif
```

- `stringReplace(<string>, <oldPattern>, <newPattern>)`

returns a string, which has all occurrences of <oldPattern> replaced with <newPattern> given then input string

<string>.

Available since 4.11.3

Example:

```
set $ConstTest$ = "123451234512345"
set $CompValue$ = stringReplace("1xy451Xy451XY45", "xy", "23")
if ($ConstTest$ = $CompValue$)
    comment "passed"
else
    set $TestResult$ = "not o.k."
    LogWarning "failed"
endif
```

- **strLength(<string>)**

Returns the number of chars in in <string>

Available since 4.11.3

Example:

```
set $tmp$ = "123456789"
set $ConstTest$ = "9"
set $CompValue$ = strLength($tmp$)
if $ConstTest$ = $CompValue$
    comment "passed"
else
    set $TestResult$ = "not o.k."
    LogWarning "failed"
endif
set $tmp$ = ""
set $ConstTest$ = "0"
set $CompValue$ = strLength($tmp$)
if $ConstTest$ = $CompValue$
    comment "passed"
else
    set $TestResult$ = "not o.k."
    LogWarning "failed"
endif
```

- **strPos(<string>, <sub string>)**

returns the first position of <sub string> in <string>. If <sub string> is not found, then "0" is the return value.

The function is case sensitive.

Available since 4.11.3

Example:

```
set $tmp$ = "1xY451Xy451xy45"
set $ConstTest$ = "7"
set $CompValue$ = strPos($tmp$, "Xy")
if $ConstTest$ = $CompValue$
    comment "passed"
else
    set $TestResult$ = "not o.k."
    LogWarning "failed"
endif
set $tmp$ = lower("1xY451Xy451xy45")
set $ConstTest$ = "2"
```

```
set $CompValue$ = strPos($tmp$,lower("xy"))
if $ConstTest$ = $CompValue$
    comment "passed"
else
    set $TestResult$ = "not o.k."
    LogWarning "failed"
endif
```

- **strPart(<string>, <start pos>, <number of chars>)**
returns the part of <string> starting with <start pos> and include the next <number of chars> chars. If there are fewer than <number of chars> after <start pos>, then the returned string will be the rest of the chars after <start pos>.
The counting of chars starts with 1.
Available since 4.11.3
Example:

```
set $tmp$ = "123456789"
set $ConstTest$ = "34"
set $CompValue$ = strPart($tmp$, "3", "2")
if $ConstTest$ = $CompValue$
    comment "passed"
else
    set $TestResult$ = "not o.k."
    LogWarning "failed"
endif
set $tmp$ = "123456789"
set $ConstTest$ = "56789"
set $CompValue$ = strPart($tmp$, strPos($tmp$, "56"), strLength($tmp$))
if $ConstTest$ = $CompValue$
    comment "passed"
else
    set $TestResult$ = "not o.k."
    LogWarning "failed"
endif
```

- **unquote(<string>, <quote-string>)**
returns the unquoted version of <string>, if <string> is quoted with <quote-string>
Only one char (the first char) of <quote-string> is accepted as a quote char. The leading white spaces are ignored.
Available since 4.11.2.1

```
set $ConstTest$ = "b"
set $CompValue$ = unquote('`b`', '`')
comment "compare values"
if ($ConstTest$ = $CompValue$)
    comment "passed"
else
    set $TestResult$ = "not o.k."
    LogWarning "failed"
endif
comment "double quote"
set $ConstTest$ = "b"
set $CompValue$ = unquote('`b`', '`')
comment "compare values"
if ($ConstTest$ = $CompValue$)
```

```
        comment "passed"
else
    set $TestResult$ = "not o.k."
    LogWarning "failed"
endif
comment "quote string will be trimmed and then only the first char is used"
comment "note: brackets are different chars"
set $ConstTest$ = "b]"
set $CompValue$ = unquote("[b]", "[{ ")
comment "compare values"
if ($ConstTest$ = $CompValue$)
    comment "passed"
else
    set $TestResult$ = "not o.k."
    LogWarning "failed"
endif
comment "not usable to remove brackets"
set $ConstTest$ = "b]"
set $CompValue$ = unquote("[b]", "[")
set $CompValue$ = unquote($CompValue$, "]"")
set $CompValue$ = unquote("[b]", "]"")
set $CompValue$ = unquote($CompValue$, "["")
set $CompValue$ = unquote(unquote("[b]", "[") , "]"")
comment "compare values"
if ($ConstTest$ = $CompValue$)
    comment "passed"
else
    set $TestResult$ = "not o.k."
    LogWarning "failed"
endif
comment "if string not quoted it will be come back without changes"
set $ConstTest$ = "b"
set $CompValue$ = unquote("b", "'")
comment "compare values"
if ($ConstTest$ = $CompValue$)
    comment "passed"
else
    set $TestResult$ = "not o.k."
    LogWarning "failed"
endif
```

12.3.5 Other String Functions

- `getDiffTimeSec`
returns a string with the integer number of seconds since the last call of `marktime`.
Available since 4.11.3.1
- `SidToName(<well known sid>)`
returns a string with the localized name of the group with the `<well known sid>`. For example, if `<well known sid>` is equal to `S-1-5-32-544` then `SidToName` returns `Administrators`.
Available since 4.11.3.1
- `GetMyIpByTarget(<target ip addr>)`
returns a list of interface IP-addresses, which are trying to reach the operating system at `<target ip addr>`. This function returns a value that is safer than the constant `%IPAddress%`.

Since Version 4.11.3.1

Example:

```
set $CompValue$ = getMyIpByTarget("%opsiServer%")
```

- `GetIpByName(<ip addr / ip name>)`
returns the IP-addresses of the computers with the <ip addr / ip name>
Since Version 4.11.3.2

```
set $ConstTest$ = "%IPAddress%"
set $string1$ = "%IPAddress%"
set $CompValue$ = getIpByName($string1$)
if ($ConstTest$ = $CompValue$)
    comment "passed"
else
    set $TestResult$ = "not o.k."
    LogWarning "failed"
endif
set $CompValue$ = getIpByName("%HostID%")
if ($ConstTest$ = $CompValue$)
    comment "passed"
else
    set $TestResult$ = "not o.k."
    LogWarning "failed"
endif
set $CompValue$ = getIpByName("%PCName%")
if ($ConstTest$ = $CompValue$)
    comment "passed"
else
    set $TestResult$ = "not o.k."
    LogWarning "failed"
endif
```

- `getLastExitCode`
returns a string that contains the value of the exitcode of the last process called by a WinBatch / DosBatch / ExecWith section.
When using a DosBatch or ExecWith section, you will normally get the exitcode from the interpreter that was called. To get the exitcode of your script, you have to define it explicitly.

Example:

```
DosInAnIcon_exit1
set $ConstTest$ = "1"
set $CompValue$ = getLastExitCode
if ($ConstTest$ = $CompValue$)
    comment "DosBatch / DosInAnIcon exitcode passed"
else
    set $TestResult$ = "not o.k."
    LogWarning "DosBatch / DosInAnIcon exitcode failed"
endif

[DosInAnIcon_exit1]
rem create an errolevel= 1
VERIFY OTHER 2> NUL
echo %ERRORLEVEL%
exit %ERRORLEVEL%
```

12.3.6 Boolean Expressions

- **runningAsAdmin**
Returns *true* if the currently running script was executed with Administrator privileges.
Available since 4.11.1.1
- **isLoginScript**
Returns *true* if the currently running script was called as *userLoginScript* using the opsi extension *User Profile Management*.
Available since 4.11.2.1
- **contains(<str>, <substr>)**
Boolean function which returns *true* if <str> contains <substr>. This function is case sensitive.
Available since 4.11.3
- **isNumber(<str>)**
Boolean function which returns *true* if <str> represents an integer.
Available since 4.11.3

12.3.7 Stringlists from the Registry

- **getRegistryKeyList32(<regkey>)**
Returns a stringlist with the names of all keys within <regkey>.
32 Bit mode (with redirection). Available since 4.11.3
- **getRegistryKeyList64(<regkey>)**
Returns a stringlist with the names of all keys within <regkey>.
64 Bit mode (without redirection). Available since 4.11.3
- **getRegistryKeyListSysnative(<regkey>)**
Returns a stringlist with the names of all keys within <regkey>.
Mode (redirection) depends on the architecture of the OS. Available since 4.11.3
- **getRegistryVarList32(<regkey>)**
Returns a stringlist with the names of all value entries associated with key <regkey>.
32 Bit mode (with redirection). Available since 4.11.3
- **getRegistryVarList64(<regkey>)**
Returns a stringlist with the names of all value entries associated with key <regkey>.
64 Bit mode (without redirection). Available since 4.11.3
- **getRegistryVarListSysnative(<regkey>)**
Returns a stringlist with the names of all value entries associated with key <regkey>.
Mode (redirection) depends on the architecture of the OS. Available since 4.11.3
- **getRegistryVarMap32(<regkey>)**
Provides a map of all name=value pairs in the given registry key <regkey>.
32 Bit Mode (with redirection). Since 4.11.3
- **getRegistryVarMap64(<regkey>)**
Provides a map of all name=value pairs in the given registry key <regkey>.
64 Bit Mode (with redirection). Since 4.11.3
- **getRegistryVarMapSysnative(<regkey>)**
Provides a map of all name=value pairs in the given registry key <regkey>.
Mode depend on the architecture of the operating system. Since 4.11.3

Example:

At first, we create entries in the registry with the following example code:

```
Registry_createkeys /32Bit
```

```
[Registry_createkeys]
openkey [HKEY_LOCAL_MACHINE\SOFTWARE\opsi.org\opsi-winst-test]
set "var1" = "value1"
set "var2" = REG_SZ:"value2"
set "var3" = REG_EXPAND_SZ:"value3"
set "var4" = REG_DWORD:444
set "var5" = REG_BINARY:05 05 05 0F 10
set "var6" = REG_MULTI_SZ:"value6|value7|de"
openkey [HKEY_LOCAL_MACHINE\SOFTWARE\opsi.org\opsi-winst-test\key1]
openkey [HKEY_LOCAL_MACHINE\SOFTWARE\opsi.org\opsi-winst-test\key2]
openkey [HKEY_LOCAL_MACHINE\SOFTWARE\opsi.org\opsi-winst-test\key3]
openkey [HKEY_LOCAL_MACHINE\SOFTWARE\opsi.org\opsi-winst-test\key4]
```

Given the registry entries in the example above, and the following code:

```
set $list$ = getRegistryVarList32("hklm\software\opsi.org\opsi-winst-test")
```

we will see the following values in the log:

```
Set $list$ = GetRegistryVarList32("hklm\software\opsi.org\opsi-winst-test")
Registry started with redirection (32 Bit)
  retrieving strings from GetRegistryVarList32 [switch to loglevel 7 for debugging]
    (string 0) var1
    (string 1) var2
    (string 2) var3
    (string 3) var4
    (string 4) var5
    (string 5) var6
```

Then we call:

```
set $list$ = getRegistryVarMap32("hklm\software\opsi.org\opsi-winst-test")
```

The following Log:

```
Set $list$ = GetRegistryVarMap32("hklm\software\opsi.org\opsi-winst-test")
retrieving strings from GetRegistryVarMap32 [switch to loglevel 7 for debugging]
  (string 0) var1=value1
  (string 1) var2=value2
  (string 2) var3=value3
  (string 3) var4=444
  (string 4) var5=05 05 05 0F 10
  (string 5) var6=value6
```

Given the registry entries in the example above, and the following code:

```
set $list$ = getRegistryKeyList32("hklm\software\opsi.org\opsi-winst-test")
```

we will get the following key in the log:

```
Set $list$ = GetRegistryKeyList32("hklm\software\opsi.org\opsi-winst-test")
Registry started with redirection (32 Bit)
  retrieving strings from GetRegistryKeyList32 [switch to loglevel 7 for debugging]
    (string 1) key1
    (string 2) key2
    (string 3) key3
    (string 4) key4
```

12.3.8 Stringlists from the Product Properties

- `getProductPropertyList(<propname>,<default value>)`
returns a stringlist of values that are referred to by the multivalue product property <propname>. If there is no connection to the opsi server, then the resulting stringlist contains only <default value>.
If you call the function `GetProductProperty` with a multivalue property, then you will get the selected values in a comma-separated string format. This will lead to problems if the returned values contain any comma chars that are not meant to be parsed.
Available since 4.11.3
Example:

```
;Property "dummymulti" has the values: ("ab", "cd", "ef", "g,h")
set $list$ = GetProductPropertyList ("dummymulti","True")
if not (" " = takeFirstStringContaining($list$,"g,h"))
    comment "GetProductPropertyList passed"
else
    set $TestResult$ = "not o.k."
    LogWarning "GetProductPropertyList failed"
endif

set $ConstTest$ = "ab,cd,ef,g,h"
set $CompValue$ = GetProductProperty ("dummymulti","True")
if ($ConstTest$ = $CompValue$)
    comment "GetProductProperty passed"
else
    set $TestResult$ = "not o.k."
    LogWarning "GetProductProperty failed"
endif
```

12.3.9 Other String Lists

- `getProfilesDirList`
Provides a list of paths to the local profiles.
Profiles that contain the following words will **not** be considered:
 - *localservice*
 - *networkservice*
 - *systemprofile*

The profile of *Default Users* is included in the list.
All **User** or **Public** are not included in the list.

Example:

```
set $list1$ = getProfilesDirList
```

results in the following log:

```
Set $list1$ = getProfilesDirList
Registry started with redirection (32 Bit)
  retrieving strings from getProfilesDirList [switch to loglevel 7 for debugging]
    (string 0)C:\Users\Administrator
    (string 1)C:\Users\Default
```

12.3.10 Keeping Track of Failed Installations

If a product installation fails, then this should be signaled to the server.

Due to the fact that there is no automatic method that detects a failed installation, testing for a failed installation has to be done using script commands.

To indicate in a *opsi-winst* script that the installation is failed we have to call the statement:

isFatalError

If this statement is called, then *opsi-winst* stops the normal execution of the script and sets the product result to *failed* (otherwise it is *success*).

Since 4.11.3.2 there is a new variant of this command:

- **isFatalError** <string>
in this case, a short error message string is passed as *actionProgress* to the opsi-server and displayed in the opsi-configed.

12.3.11 Commands to Control the Log

- **includeLog** <file name> <tail size> //since 4.11.2.1
Includes the file <file name> as a log file, where the last <tail size> lines of the full log are written into this log file. If you start another program that produces a log file, you could see that other program's log file in the opsi-winst script log using this command.
Since version 4.11.3.2, a negative <tail size> can be given, which will then include the first <tail size> lines from the top of the log file (referred to as *Head* mode).
Example:

```
includeLog "%Scriptpath%\test-files\10lines.txt" "5"
```

12.4 Secondary Sections

12.4.1 Patches-Sections

12.4.1.1 Call Parameter

The name of the file to be patched is passed as a parameter.

There are optional modifiers:

- **/AllNTUserProfiles**
If a patch section is called with this modifier, then all directories under %UserProfileDir% will be patched, which means that this patch is performed for all user profiles.
When a *Patches* is called within a [ProfileActions] section, then the modifier */AllNTUserProfiles* is implicit.
In logscript mode, %UserProfileDir% will be interpreted as %CurrentProfileDir%.
(Since Version 4.11.3.2)

12.4.2 ExecWith Sections

If you call an ExecWith section with the interpreter *powershell*, then the content of the section will be stored in a temporary file with the extension *.ps1*.

12.4.3 WinBatch Sections

12.4.3.1 Call Parameter (Modifier)

- `/WaitSeconds` [number of seconds]
If a call includes the parameter `/WaitSeconds` [number of seconds], then *opsi-winst* is waiting for [number of seconds] before proceeding. In the default configuration, we also wait for any programs that are currently running to finish. If we combine the parameter `/WaitSeconds` with the option `/LetThemGo`, then *opsi-winst* continues processing after the waiting time is finished.
- `/WaitForProcessEnding` <program name>
Waits for the process called <program name> to end.
Should be combined with `/TimeOutSeconds`.

Explanation:

When starting an external process from a winbatch call, the *opsi-winst* waits for the current process to finish before executing the next command in the script.

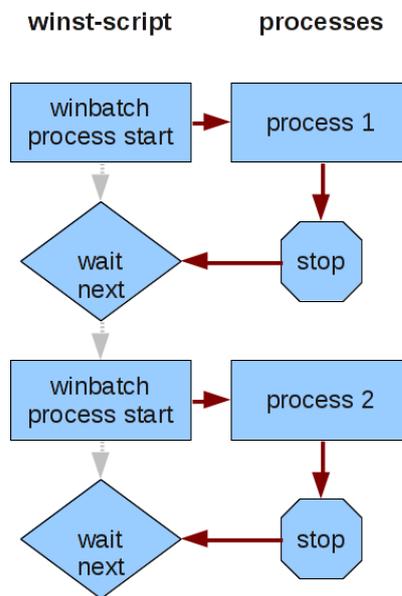


Figure 12.1: Sequential processing of a script that waits for the end of a program

There are some external programs which start another process and then end without waiting for their child process to end. From the point of view of *opsi-winst*, the process is ended and the next command could be started.

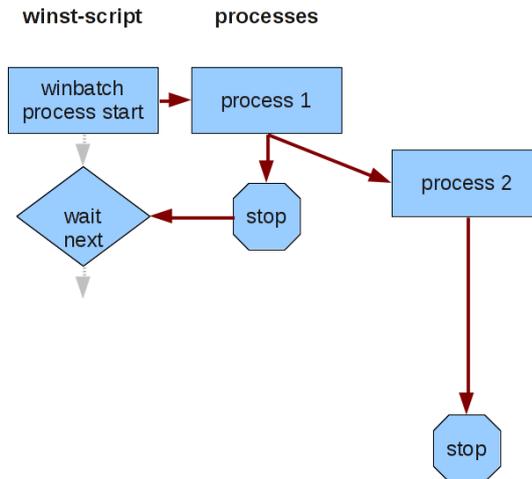


Figure 12.2: End of process while child process is still running

If you run an uninstall program and a setup program in sequence and the uninstall program works with such a child process, you can have conflicting processes running because the uninstallation and installation processes are running at the same time.

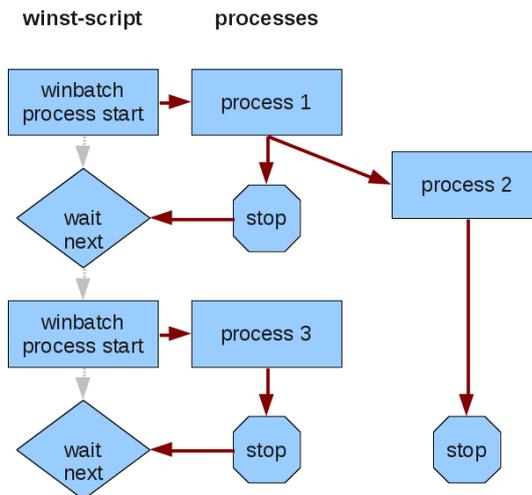


Figure 12.3: Overlapping of a child process and a parent process

Using the modifier `/WaitForProcessEnding` helps to avoid such a situation.

- `/TimeoutSeconds <seconds>`
 A timeout setting. After waiting `<seconds>`, *opsi-winst* will end the process.
 Since version 4.11.3, `/TimeoutSeconds` may be used without a waiting condition (e.g. `/WaitForProcessEnding`) but not in combination with `/WaitSeconds`.
 Example:

```
Winbatch_uninstall /WaitForProcessEnding "uninstall.exe" /TimeoutSeconds 20
[Winbatch_uninstall]
"%ScriptPath%\uninstall_starter.exe"
```

- **/RunElevated**
Starts a process that has a security token with elevated privileges. This modifier has the following restrictions:
 - For NT5 it does not change anything.
 - A process started with this modifier has no network access. So you should copy a program to a temporary local directory, but do not start it from a network share.
 - You may see problems while using the graphical interface. Therefore true silent installations are the better choice in this case.
 - Functions only in the context of *opsi-winst*.
- **getLastExitCode**
Returns a string that contains the value of the exitcode of the the process that was last called by a WinBatch / DosBatch / ExecWith section.
When using a DosBatch or ExecWith section, you will normally get the exitcode of the interpreter that was called. To get the exitcode of your script you have to define it explicitly.

12.5 Miscellaneous

- wisynth: TuibInstScript.doInfo: optimal fill for LabelInfo
- name of installing product shown in initial info block. fixes bug #367
- wisynth: executeWith: uses extension *.ps1* if interpreter = powershell, fixes bug #382
- logfiles are not locked for reading (revert to previous methodology) while opsi-winst is running fixes: <https://forum.opsi.org/viewtopic.php?f=7&t=3655>
- winbatch: tests to determine if a file extension is exe or com, and is not case sensitive. fixes: <https://forum.opsi.org/viewtopic.php?f=7&t=3592#p18127>
- Winbatch: The parameter for /timeoutseconds may now be a variable. fixes bug #352

Chapter 13

Miscellaneous

13.1 Changelogs:

13.1.1 Changelog opsi-configed

opsi-configed (4.0.2.5.8-1) stable; urgency=low

- fixed: host groups were not reloaded on reload request

-- Rupert Roeder <r.roeder@uib.de> Mon, 15 Oct 2012 12:00:00 +0200

opsi-configed (4.0.2.5.6-1) stable; urgency=low

- fixed: problem when server property is missing
- fixed: a new client must not have the name of the opsi server (or depots)

-- Rupert Roeder <r.roeder@uib.de> Wed, 26 Sep 2012 12:00:00 +0200

opsi-configed (4.0.2.5.5-1) stable; urgency=low

- fixed: some problem when changing between not synchronized depots

-- Rupert Roeder <r.roeder@uib.de> Fri, 07 Sep 2012 18:00:00 +0200

opsi-configed (4.0.2.4-1) stable urgency=low

- updated FR language file
- eliminated NPEs for not authenticated users

—Rupert Roeder <r.roeder@uib.de> Thu, 16 Aug 2012 11:38:32 +0200

opsi-configed (4.0.2.2-1) testing; urgency=low

- new button indication of existing product property entries
- new client action resetLocalProducts

—Rupert Roeder <r.roeder@uib.de> Thu, 16 Jul 2012 11:38:32 +0200

opsi-configed (4.0.2.1-1) testing; urgency=low

- calls to `hostControl_reachable` and `hostControl_getActiveSessions` are made threaded to avoid blocking of the config if there are problems
- improvements in gui of client search dialogs
- added support of description field for a saved search
- added support of description field for tree groups
- new button indication of existing product property entries
- new client action `resetLocalProducts`

—Rupert Roeder <r.roeder@uib.de> Thu, 20 Jun 2012 14:38:32 +0200

opsi-configed (4.0.2.0-1) testing; urgency=low

- implementation of `DIRECTORY` extension of treeview
- coloring of product versioninfo which are not equal to the server version
- report column can now be edited (by transforming the values to values of `ActionResult`, `ActionProgress`, `LastState`)
- product properties and host configs are marked by bold font when not being identical with the default values
- an additional product column showing the (full) product names can be used
- if a opsi client is created there now is a check for existing host names which will not be reused without confirmation
- new sorting of depots, tooltip added for depots

—Rupert Roeder <r.roeder@uib.de> Thu, 14 Jun 2012 14:38:32 +0200

opsi-configed (4.0.1.27-1) testing; urgency=low

- improved window handling of client selection dialog

—Rupert Roeder <r.roeder@uib.de> Thu, 03 May 2012 14:38:32 +0200

opsi-configed (4.0.1.26-1) testing; urgency=low

- bugfix: product properties are reloaded for each selected client if a product was selected

—Rupert Roeder <r.roeder@uib.de> Thu, 26 Apr 2012 14:38:32 +0200

opsi-configed (4.0.1.25-1) testing; urgency=low

- improvement: Added a new client selection dialog

—Felix Rohrbach <fxrh@gmx.de> Wed, 21 Mar 2012 14:38:32 +0200

13.1.2 Changelog opsi-winst

opsi-winst (4.11.3.3) stable; urgency=low

- new command `setActionProgress <string>`
- commandline parameter `/productid <string>` This parameter forces the winst to try to start a service connection using the local `opsiclientd.conf`. This will be used in the `opsi-template-with-admin` to run the local script in opsi context
- Versioninfo: try to find informations also on other than the stored codepage

- wisynth: doRegistryAllNTUserDats: use getProfilesDirList
- widatamodul: retrieveJSONObject: JSON Bench Log entry for every service call
- wimain: saveVersionToProfile/readVersionFromProfile : if there is no access to Appdata, we use the profile directory for the .opsi.org\userLoginScripts.ini
- wilog: PartOpenForReading: fix for timing problem *Floatingpoint Underflow*—Detlef Oertel <d.oertel@uib.de> Thu, 13 Sep 2012:15:00:00 +0200

opsi-winst (4.11.3.2) stable; urgency=low

- new command FatalOnSyntaxError (default= true)
- every script syntax error is logged
- every script syntax error increases the errorcounter
- if FatalOnSyntaxError then every script syntax error is a fatal error and *Syntax Error* is set as actionProgress
- now isFatalError accepts a string which is stored to the server as actionprogress and showed in the opsi-configed
- includeLogFile now works in head mode if you give a negative number as line count
 - changed in TLogInfo.includelogtail
- new stringlist function: getProfilesDirList: All existing Profiles dirs except: *localservice, networkservice, systemprofile*
- Patches sections may now called with %userprofiledir% as part of the file name and /AllNTUserProfiles so it will run on all profiles. If running in Machine mode, the /AllNTUserProfiles is implizit in the [ProfileActions]. If running in LoginScript mode, the %userprofiledir% is the %CurrentProfileDir%.
- new stringfunction: GetIpByName(<ip addr / ip name>) : string
- Files sections changed: wisynth: doFileActions: Modifier /AllNtUserProfiles and /AllNtUserSendto now based on getProfilesDirList
- new stringlist commands to retrieve maps from the registry:
 - getRegistryVarMap32(<regkey>)
 - getRegistryVarMap64(<regkey>)
 - getRegistryVarMapSysnative(<regkey>)
- winbatch: (Startprocess*) catching exceptions if execute bit is missing on samba4 shares

— Detlef Oertel <d.oertel@uib.de> Mon, 30 Jul 2012:15:00:00 +0200

opsi-winst (4.11.3.1) stable; urgency=low

- winbatch: /timeoutseconds now allowed without /waitforprocessending (not with /waitsec)
- Winbatch: The parameter to /timeoutseconds may now be a variable fixes #352
- winbatch /RunElevated
- patchhosts sections: 64 bit compatibility, fixes #327
- manifest added: <requestedExecutionLevel level="highestAvailable" />:
 - <http://www.lazarus.freepascal.org/index.php?topic=14976.0>
 - <http://lazarus.freepascal.org/index.php/topic,11659.msg69861.html>
 - <http://msdn.microsoft.com/en-us/library/bb756929.aspx>
- new stringfunction: getDiffTimeSec (Time in seconds since last marktime)

- name of installing product shown in initial info block. fixes #367
- Exitcode of DosBatch, Execwith is now logged and can be retrieved by getLastExitcode
 - be aware: this is the exitcode of the called program so use something like: exit %ERRORLEVEL% , fixes #161
- on userLoginScript: retry if there no user could be detected
- wisynth: TuibInstScript.doInfo: optimal fill for LabelInfo
- wisynth: executeWith: use extension *.ps1* if interpreter = powershell, fixes #382
- new function: NormalizeWinst (set normal window state), fixes #346
- old function IconizeWinst (set minimized window state), fixes #346
- old function RestoreWinst (set maximized window state), fixes #346
- new stringfunction: stringReplace(string, oldPattern,newPattern) : string ;
this function replaces all occurrences of one substring with another in a string
it works not case sensitive , fixes #328
- new stringfunction: GetMyIpByTarget(<target ip addr>) : string
- new boolean function: contains(<str>, <substr>) : boolean ; case sensitive, fixes #365
- old boolean function: isNumber(<str>) : boolean
- new stringlist function: getProductPropertyList(<propname>,<default value>)
- wispecfolder: fix using winapi call GetDefaultUserProfileDirectory for %DefaultUserProfileDir%
- wispecfolder: fix using winapi call GetAllUsersProfileDirectory for %CommonProfileDir%
- new stringfunction SidToName(<well known sid>) : gives localized name of the sid
- wisynt: try-except in doRegistryHackRegeditFormat
- logfiles now again not locked against reading while opsi-winst is running
fixes: <https://forum.opsi.org/viewtopic.php?f=7&t=3655>
- new commands: include_insert "<file name>" : Replaces the command with a empty line and then inserts the content of "<file name>" after this line
include_append "<file name>" : Replaces the command with a empty line and then appends the the content of "<file name>" to the end of the script
 - <file name> is either:
 - * a complete valid Path
 - * The name of a file in %WinstDir%\lib
 - * The name of a file in %ScriptPath%
 - should be placed after requiredWinstVersion
- new stringlist commands to retrieve lists from the registry:
 - getRegistryKeyList32(<regkey>)
 - getRegistryKeyList64(<regkey>)
 - getRegistryKeyListSysnative(<regkey>)
 - getRegistryVarList32(<regkey>)
 - getRegistryVarList64(<regkey>)
 - getRegistryVarListSysnative(<regkey>)
- new stringfunction strLength(<string>)

- new stringfunction strPos(<string>, <substr>)
- new stringfunction strPart(<string>, <start pos>, <number of chars>)
- winbatch: test if extension of existsting file is exe or com now not case sensitive. fixes: <https://forum.opsi.org/-viewtopic.php?f=7&t=3592#p18127>

— Detlef Oertel <d.oertel@uib.de> Tue, 11 Jan 2012:15:00:00 +0200

opsi-winst (4.11.2.12) stable; urgency=low

- fix for getproductmap widatamodul: TOpsi4Data.initProduct: reset FProductOnClient_aktobject with nil

— Detlef Oertel <d.oertel@uib.de> Tue, 31 Jul 2012:15:00:00 +0200

opsi-winst (4.11.2.11) stable; urgency=low

- wispecfolder: fix for the fix using winapi call GetProfilesDirectory for %profiledir%

— Detlef Oertel <d.oertel@uib.de> Tue, 3 Jun 2012:15:00:00 +0200

opsi-winst (4.11.2.10) stable; urgency=low

- wispecfolder: fix using winapi call GetProfilesDirectory for %profiledir%

— Detlef Oertel <d.oertel@uib.de> Tue, 2 Jun 2012:15:00:00 +0200

13.1.3 Changelog hwaudit

hwaudit (4.0.5-1) stable; urgency=low

- added dell-expresscode to computer_system
- added productProperty loglevel log-file
- changeDirectory added (package requires now winst: 4.11.2.6)
- new temp-Directory C:\opsi.org\tmp instead of old C:\tmp

— Erol Ueluekmen <e.ueluekmen@uib.de> 31 Jul 2012

hwaudit (4.0.4-3) stable; urgency=low

- get timeout from property
- Ticketid 2012021510000025

— Bardo Wolf <b.wolf@uib.de> 23 Feb 2012

hwaudit (4.0.4-3) stable; urgency=low

- get timeout from property
- Ticketid 2012021510000025

— Bardo Wolf <b.wolf@uib.de> 23 Feb 2012

hwaudit (4.0.4-2) stable; urgency=low

- get timeout from property
- Ticketid 2012021510000025

— Bardo Wolf <b.wolf@uib.de> 23 Feb 2012

13.1.4 Changelog swaudit

swaudit (4.0.2-1) stable; urgency=low

- code cleanup
- remove all opsi 3 stuff
- use opsi-winst 4.11.3 methods
- extended search off by default (no data at NT6)
- replace delete by del
- replace c:\tmp by c:\opsi.org\tmp

—detlef oertel <d.oertel@uib.de> Fri, 10 Aug 2012 15:00:00 +0100

swaudit (4.0-15) testing; urgency=low

- new property: readfromfile to switch off audit and use the filebuffer files instead ONLY DEBUG - do not use if you are not sure what you doing! (Do not make a audit but read values from c:\tmp and send them)

—detlef oertel <d.oertel@uib.de> Fri, 11 May 2012 15:00:00 +0100

swaudit (4.0-14) testing; urgency=low

- workaround for bug#354 (set version to " " if zero) deleted because it is fixed

—detlef oertel <d.oertel@uib.de> Tue, 26 Apr 2012 16:00:00 +0100

swaudit (4.0-13) testing; urgency=low

- set version to " " if zero - workaround for bug#354

—detlef oertel <d.oertel@uib.de> Tue, 15 Mar 2012 16:00:00 +0100

swaudit (4.0-12) testing; urgency=low

- new property: usekeyfinder to switch off keyfinder call

—detlef oertel <d.oertel@uib.de> Tue, 29 Nov 2011 16:00:00 +0100

13.1.5 Changelog jedit

jedit_4.5.2-2 stable; urgency=low

- opsi-winst.xml for version 4.11.3.2

—detlef oertel <d.oertel@uib.de> Tue, 31 Jul 2012 16:01:53 +0200

jedit_4.5.2-1 stable; urgency=low

- update to version 4.5.2
- opsi-winst.xml for version 4.11.3.1
- opsi-winst.xml for **ins**,.opsiscript,*opsiinc

—detlef oertel <d.oertel@uib.de> Thu, 12 Jul 2012 16:01:53 +0200

jedit_4.5.1-1 stable; urgency=low

- update to version 4.5.1
- property gui_language description: jedit localization values: ["auto", "de", "en", "fr"] default: ["en"]

—bardo wolf <b.wolf@uib.de> 12 Jun 2012

13.1.6 Changelog opsi-adminutils

opsi-adminutils (4.0.2.1-1) stable; urgency=low

- replace ussf by opsi-setup-detector
- update auf InstEd-1.5.15.26.msi
- winscp 4.3.9

— detlef oertel <d.oertel@uib.de> 06 Aug 2012

opsi-adminutils (4.0.1.24-1) stable; urgency=low

- try to fix [2012031610000012] opsi-adminutils webstart funktioniert nicht.
- postinst check if newer configed.jar swingx.jar on server
- winscp 4.3.7
- property javaweb installer or local configed
- swingx.jar
- configed.jar 4.0.1.24
- language selection
- replace %ProgramfilesDir% by %Programfiles32Dir%

— Bardo Wolf <b.wolf@uib.de> 16 Mar 2012

opsi-adminutils (4.0.1.12-4) stable; urgency=low

- corrected linkfolder for opsi-configed

— Rupert Röder <r.roeder@uib.de> 07 Mar 2012

opsi-adminutils (4.0.1.12-3) stable; urgency=low

- putty 0.62
- winscp 4.3.5

— Susanne Ehlers <s.ehlers@uib.de> 13 Dec 2011

opsi-adminutils (4.0.1.12-2) stable; urgency=low

- JXv3.1 replaced by jxplorer-3.2.2-windows-installer.exe

— Bardo Wolf <b.wolf@uib.de> 17 Oct 2011

13.1.7 Changelog opsi-template

opsi-template (4.0.2-2) stable; urgency=low

- requiredWinstVersion >= "4.11.3.2"
- Set \$LogDir\$ = "%SystemDrive%\opsi.org\tmp"
- new: *_short32.opsiscript scripts with includes
- winbatch_install: changed inno-setup example: default without answer file
- winbatch_uninstall: changed NSIS example: /WaitForProcessEnding "Au_.exe"

—detlef oertel <d.oertel@uib.de> Fri, 03 Aug 2012 16:01:53 +0200

opsi-template (4.0.2-1) stable; urgency=low

- requiredWinstVersion >= "4.11.2.6"
- added ChangeDirectory "%SCRIPTPATH%"
- changed delete to del (in delsub)
- changed at msi to ALLUSERS=1

—detlef oertel <d.oertel@uib.de> Mon, 02 Jun 2012 16:01:53 +0200

13.1.8 Changelog opsi-template-with-admin

opsi-template-with-admin (4.0.2-2) stable; urgency=low

- requires opsi-winst >= 4.11.3.3
- uses /productid switch at opsi-winst call
- code cleanup (use of synative instead of 64bit)
- property debug: switches loglevel and lock keyboard
- code cleanup

—detlef oertel <d.oertel@uib.de> Thu, 19 Sep 2012 15:01:53 +0200

opsi-template-with-admin (4.0.2-1) stable; urgency=low

- requires opsi-winst >= 4.11.3
- using SidToName instead of psgetsid.exe
- priority set to -20
- first reboot is commented (;ExitWindows /ImmediateReboot). Uncomment it if you have problems with the autologon

—detlef oertel <d.oertel@uib.de> Wed, 04 Jul 2012 15:01:53 +0200

13.1.9 Changelog opsi-client-agent

opsi-client-agent (4.0.2.1-2) stable; urgency=low

- no interactive message on copy error if \$INST_SubModus\$="CREATE_CLIENT"
- opsi-winst 4.11.3.2
- use of GetMyIpByTarget()
- use of GetIpByName()
- devided Files_redist_cleanup to Files_redist_cleanup_c and Files_redist_cleanup_d

— Detlef Oertel <d.oertel@uib.de> Mon, 4 Jun 2012 15:00:00 +0100

13.1.10 Changelog windows Netboot Products

windows (4.0.2-1) stable; urgency=low

- Security fix in net mount while winpe works (NT6)
- set alignment for partitions (NT5)
- show_drivers.py modified to show automated including additional driver over inventory
- create_driver_links for handling with fileadmingroup from opsi.conf-file

— uib GmbH <info@uib.de> Fr, 29 Jun 2012 15:19:15 +0000

windows (4.0.1-11) stable; urgency=low

- fix in multilanguage control file (time zone)
- fix image names windows server

— uib GmbH <info@uib.de> Mon, 2 Apr 2012 15:19:15 +0000

windows (4.0.1-10) stable; urgency=low

- multilanguage control file
- additional locales and keyboard layouts

— uib GmbH <info@uib.de> Tue, 24 Jan 2012 14:19:15 +0000

windows (4.0.1-9) stable; urgency=low

- blockAlignment added.

— uib GmbH <info@uib.de> Wed, 16 Nov 2011 11:06:24 +0000

windows (4.0.1-8) stable; urgency=low

- descriptions now english
- Added new properties for patching unattend.xml Component Microsoft-Windows-International-Core-WinPE
winpe_uilanguage winpe_uilanguage_fallback winpe_inputlocale
- added new properties to unattend.xml.template

- show_drivers.py

—bardo wolf <b.wolf@uib.de> Mon, 31 Oct 2011 09:09:50 +0000

windows (4.0.1-7) stable; urgency=low

- unattend.xml patched for postinst-routines like 2008-r2
- show_drivers.py modified to show right list of drivers; now with messages for duplicate drivers.

—uib GmbH <info@uib.de> Thu, 22 Sep 2011 14:19:15 +0000

windows (4.0.1-6) stable; urgency=low

- Remove Display settings from unattend.xml.template

—uib GmbH <info@uib.de> Thu, 25 Aug 2011 12:25:50 +0000

windows (4.0.1-5) stable; urgency=low

- Call auditHardwareOnHost_setObsolete

—uib GmbH <info@uib.de> Tue, 19 Jul 2011 13:17:02 +0000

windows (4.0.1-4) stable; urgency=low

- Modify driver integration

—uib GmbH <info@uib.de> Mon, 18 Jul 2011 15:05:10 +0000

13.1.11 Changelog python-opsi

python-opsi (4.0.2.4-1) stable; urgency=low

- fixes method setProductActionRequestWithDependencies after host_createOpsClient
- added default dhcp string and text options that the values will be set in double-quotes (fixes#403)
- added method userIsReadOnlyUser()
- WindowsDriverIntegration: do not break when no devices found in txtsetup.oem (corrupted txtsetup.oem)

—Erol Ueluekmen <e.ueluekmen@uib.de> Thu, 27 Sep 2012 10:35:17 +0200

python-opsi (4.0.2.3-1) testing; urgency=low

- Workaround for bootimage: wait if blockfile to partition does not exist.
- Automated additional-driver integration support.
- hostControl-Fix for host_reachable method.
- added opsiFileAdminhandling, added new opsi.conf File.
- dellexpresscode for hwinvent implemented
- licensekey length increased to 1024

—Erol Ueluekmen <e.ueluekmen@uib.de> Tue, 17 Jul 2012 13:33:13 +0200

python-opsi (4.0.2.2-1) testing; urgency=low

- Workaround for python 2.7 in jsonrpc-backend: compressed data will send as bytearray
- fix for isc-dhcp-server for oneiric and precise

—Erol Ueluekmen <e.ueluekmen@uib.de> Mon, 11 Jun 2012 13:42:58 +0200

13.1.12 Changelog opsipxeconfd

opsipxeconfd (4.0.2.6-1) stable; urgency=low

- Fix: if always switched to none, don't create new pipe for client.

— e.ueluekmen@uib.de <erol@erollinux.uib.local> Mon, 08 Oct 2012 14:42:45 +0200

opsipxeconfd (4.0.2.5-1) experimental; urgency=low

- Fix: dont lose append entry from template, if no productProperty set

— Erol Ueluekmen <e.ueluekmen@uib.de> Tue, 25 Sep 2012 14:48:08 +0200

opsipxeconfd (4.0.2.4-1) experimental; urgency=low

- write pipe again, if product has set always actionRequest for client.

— Erol Ueluekmen <e.ueluekmen@uib.de> Tue, 18 Sep 2012 18:27:36 +0200

opsipxeconfd (4.0.2.3-1) experimental; urgency=low

- when spaces are in append, does not shuffle productPropertyStates

— Jim Pizagno <j.pizagno@uib.de> Wed, 05 Sep 2012 17:03:22 +0200

opsipxeconfd (4.0.2.2-1) testing; urgency=low

- fixing init.d lsb-header for SUSE based servers. Removing runlevel 4
- fileadmingroup added in /etc/opsi/backends/file.conf

— Erol Ueluekmen <e.ueluekmen@uib.de> Fri, 22 Jun 2012 16:26:42 +0200

13.1.13 Changelog opsiconfd

opsiconfd (4.0.2.4-1) stable; urgency=low

- added readOnly-State in session and worker

— Erol Ueluekmen <e.ueluekmen@uib.de> Thu, 04 Oct 2012 11:18:36 +0200

opsiconfd (4.0.2.3-1) testing; urgency=low

- fixing floatdivision in statistics-class

— Erol Ueluekmen <e.ueluekmen@uib.de> Wed, 18 Jul 2012 17:05:54 +0200

opsiconfd (4.0.2.2-1) testing; urgency=low

- omb: python 2.4 fixing import hashlib
- fixing init.d lsb-header for SUSE based servers. Removing runlevel 4

— Erol Ueluekmen <e.ueluekmen@uib.de> Fri, 22 Jun 2012 15:06:59 +0200

13.1.14 Changelog opsi-utils

opsi-utils (4.0.2.3-1) stable; urgency=low

- opsi-product-updater: Fix for using more than one repositories with different onlyDownload options.

—Erol Ueluekmen <e.ueluekmen@uib.de> Thu, 11 Oct 2012 14:13:35 +0200

opsi-utils (4.0.2.2-1) testing; urgency=low

- Added opsiFileAdminGroup handling.
- opsi-product-updater: Timestamps in Mail

—Erol Ueluekmen <e.ueluekmen@uib.de> Thu, 19 Jul 2012 11:44:37 +0200

13.1.15 Changelog opsi-linux-bootimage

opsi-linux-bootimage (20120728-1) stable; urgency=low

- kernel 3.5
- python-opsi 4.0.2.3-1

—Erol Ueluekmen <e.ueluekmen@uib.de> Mon, 16 Jul 2012 16:42:42 +0200

opsi-linux-bootimage (20120119-1) stable; urgency=low

- kernel 3.2.1
- fix french-localisation
- added gptfdisk toolset
- python-opsi 4.0.1.37-1

—Erol Ueluekmen <e.ueluekmen@uib.de> Thu, 19 Jan 2012 15:01:04 +0100

13.1.16 Changelog opsi-depotserver

opsi-depotserver (4.0.2.3-1) stable; urgency=low

- Config-default-creation by first running --init-current-config fixed

—Erol Ueluekmen <e.ueluekmen@uib.de> Tue, 18 Sep 2012 17:42:23 +0200

opsi-depotserver (4.0.2.2-1) quantal; urgency=low

- Added opsiFileAdmins handling.

—Erol Ueluekmen <e.ueluekmen@uib.de> Thu, 19 Jul 2012 11:46:20 +0200

13.1.17 Changelog opsi4ucs

opsi4ucs (4.0.2.3-1) stable; urgency=low

- Join script modified:
 - only patch dhcp if it's installed
 - restart and patch samba4 only, if samba4 is installed
- don't build opsi4ucs-ldap-schema.

—Erol Ueluekmen <e.ueluekmen@uib.de> Tue, 02 Oct 2012 15:14:44 +0200

opsi4ucs (4.0.2.2-1) experimental; urgency=low

- For handling with new pcpatch-group added opsi.conf.
- UCSOverview-page changed from applet to java-webstart.
- Added new opsi-auth pam module for UCS 3

—Erol Ueluekmen <e.ueluekmen@uib.de> Thu, 24 Aug 2012 10:17:21 +0200

opsi4ucs (4.0.2.1-1) testing; urgency=low

- Featurepack-Release 4.0.2 for ucs.
- First Release for UCS 3 Support:
 - uninvention-ldap-backend have reached his EOL since ucs3 (default backend: file)
- Fixes 99opsi4ucs.inst-Joinscript

—Erol Ueluekmen <e.ueluekmen@uib.de> Fri, 29 Jun 2012 10:13:10 +0200

13.1.18 Changelog opsi-atftp

opsi-atftp (0.7.dfsg-2.1) stable; urgency=low

- fix: don't overwrite use_inetd in /etc/default/atftp if exists
- Erol Ueluekmen <e.ueluekmen@uib.de> Wed, 17 Oct 2012 11:11:37 +0200

opsi-atftp (0.7.dfsg-2) stable; urgency=low

- fix uncommented entries for tftp-service in inetd.conf
- Erol Ueluekmen <e.ueluekmen@uib.de> Fri, 12 Oct 2012 13:25:16 +0200

opsi-atftp (0.7.dfsg-1.9) testing; urgency=low

- Replaces univention-tftp package
- Erol Ueluekmen <e.ueluekmen@uib.de> Fri, 10 Aug 2012 11:21:13 +0200

opsi-atftp (0.7.dfsg-1.8) stable; urgency=low

- LSB Header added in init-script
- Erol Ueluekmen <e.ueluekmen@uib.de> Tue, 24 Jan 2012 18:01:39 +0100