Mandriva

Pulse 2

User Manual

55, boulevard Saint-Martin - 75003 Paris - France
Tel. +33 (0) 1 40 41 00 41 - Fax +33 (0) 1 40 41 92 00
www.mandriva.com

User Manual for Mandriva Pulse 2

Version 1.2.4 Rev A

by Jean-Philippe Braun, Damien Chrisment & Nicolas Rueff

Table of Contents

Introduction

Pulse 2 is an Open Source tool that simplifies application deployment, inventory, and maintenance of an IT network.

It provides the following features:

- Save and restore hard drives: creation of a rescue disk image to restore a computer, or masters to install on a new computer;
- remote application deployment: deployment of new applications or updates on the whole network;
- maintenance: software and hardware inventory, remote diagnostic and control.

1. Main Features

Unique web administration console

- Easy to use
- Use it anywhere
- To manage heterogeneous environments

Application deployment: install and uninstall easily

- Distributed architecture takes advantage of the existing infrastructure
- Deployment scheduling
- Individual or (functional or technical) group-based deployment
- Dynamic inventory update
- Limited bandwidth during transfers
- Target computer Wake On LAN and shutdown functions

Remote distribution

- Distributed architecture that uses the existing infrastructure
- Planned deployment
- Single deployment or by functional or technical group
- Inventory dynamic update
- Bandwidth limitation during the transfers
- Wake On LAN and shutdown of the target computer

Inventory

- Software inventory
- Hardware inventory
- Inventory history
- Creation of groups based on inventory criteria

Supports heterogeneous platforms

- Remote deployment and inventory on MS Windows, GNU/Linux (Mandriva, RedHat, Debian, Ubuntu., etc.), Mac OSX, HP-UX, IBM AIX and Solaris systems.

Interoperability, integration and development

- Interoperable with GLPI
- Integration with external inventories
- Documented API for fast and maintainable integration
- XML-RPC services

The Pulse 2 Environment

1. Overview

1.1. Components of Pulse 2

Server side

For the purposes of application deployment, the Pulse 2 system is divided into several independent logical parts, each of which handles one aspect of application deployment:

- **MMC interface** which allows the user to manage the application deployments,
- **MMC interface** service: in addition to authentication, retrieval of various entities, etc., this handles the injection of application deployment queries into the database,
- **Scheduler**, which regularly consults the lists of deployments to be scheduled and dispatches them to the various launchers according to the load, availability, and performance objectives. It also collects the results of deployments in progress and injects the result into the database on completion.
- **Launcher**, in charge of setting up the actual connections to client workstations: carrying out deployment orders and retrieving the software deployment statuses.
- **Package server**, sends out the list of available packages and serves these packages.
- **Inventory server**, records inventory feedback from client workstations.

Application deployments are distributed on the various existing launchers as far as possible. In the event of unavailability (saturated launchers, client cannot be reached, etc.), the command is automatically rescheduled.

The flexibility of this architecture allows the various components to be distributed over several different machines according to the expected performance (number of simultaneous application deployments, etc.), the network architecture, etc.

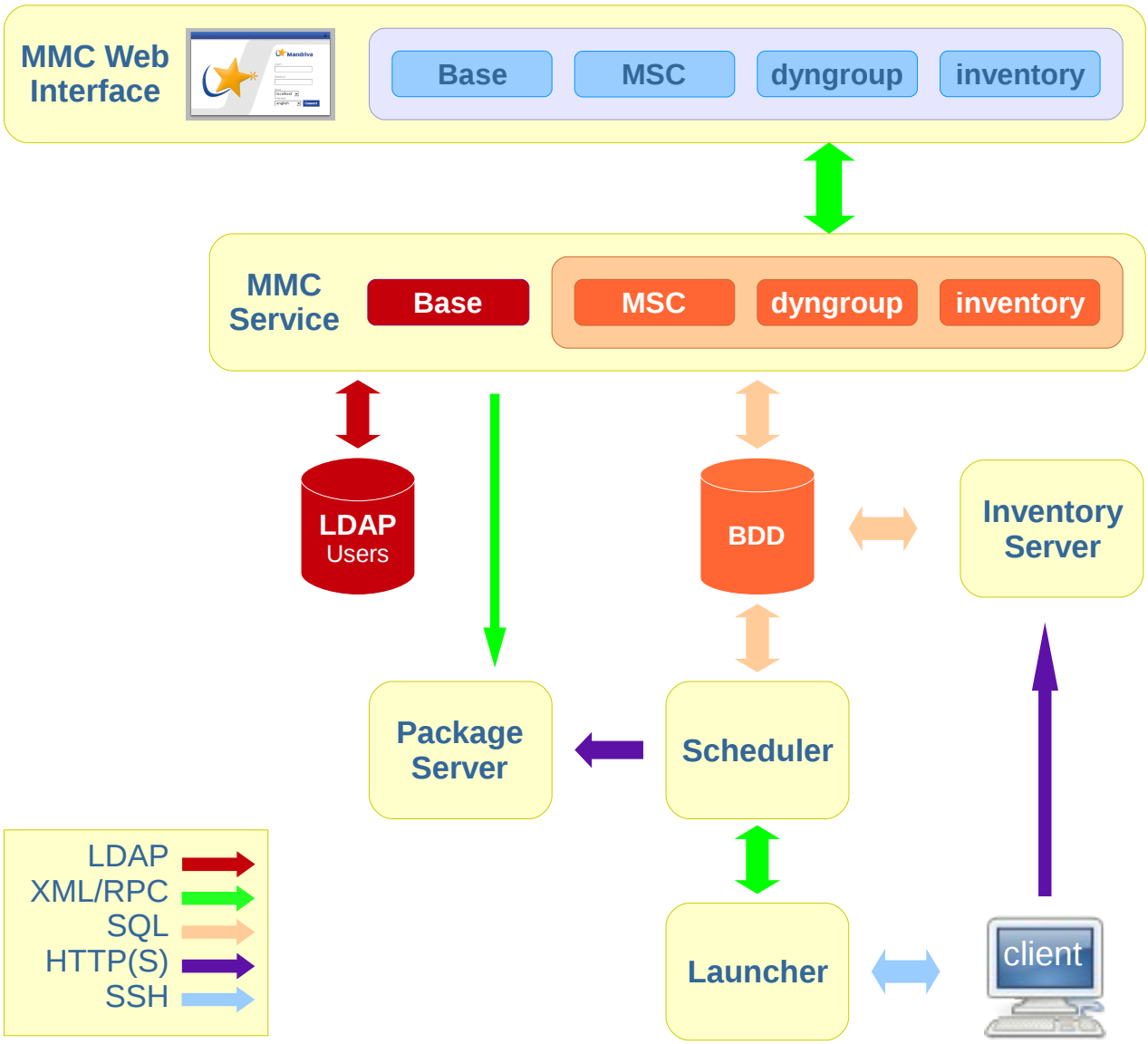
Internally, a launcher comprises instances of the Pulse 2/launcher agent, each of which can create up to 300 connections at the same time.

Client side

A **SSH agent** must be installed on the client workstations. Pulse 2 uses the SSH secure protocol for deployments.

An **inventory agent** uploads the software and hardware inventory to the inventory server.

Block Diagram



1.2. Overview of application deployment

Deployment of a package (between the Launcher and the Client) takes place in three separate phases:

1. Transfer,
2. Execution,
3. Deletion.

The deletion phase is usually followed by an inventory uploading phase, the transfer can be preceded by a Wake on LAN (WOL), and deletion can be followed by a machine shutdown.

Transfer Phase

Pulse 2 can send a package to the target machine via two modes.

Push Mode

When deployment is performed in Push Mode, the package is copied to the client on the initiative of the server. The server (in this case, the Pulse 2 launcher) takes the package and copies it to the client.

Push/Pull Mode

When deployment is performed in Push/Pull Mode, the package is copied to the client on the initiative of the client. The server only sends the execution order and the package downloading URL to the client. The client then downloads the package by itself.

Execution Phase

When the package has been copied to the client, it is installed by an installation script contained in the files of the package. The launcher initiates the execution of this script on the target machine.

Deletion Phase

If the execution phase is successfully completed, the package installation files must be deleted from the client. The launcher initiates the file deletion command on the target machine. If one or more files were generated by the installation of the package, they will not be deleted.

Error Management

The deployment process can sometimes halt unexpectedly during one of these three main phases. If an error is detected in one of these phases, the deployment process is stopped. By default, deployment is restarted after 60 minutes. This can be repeated up to three times if the subsequent deployment attempts fail.

When a deployment fails, its status changes to “rescheduled”. If all repeat deployment attempts fail, the deployment status changes to 'aborted'. The deployment can then be relaunched from the interface.

1.3. Application deployment workflow

The following diagram describes the internal sequence of the deployment phase within Pulse 2.

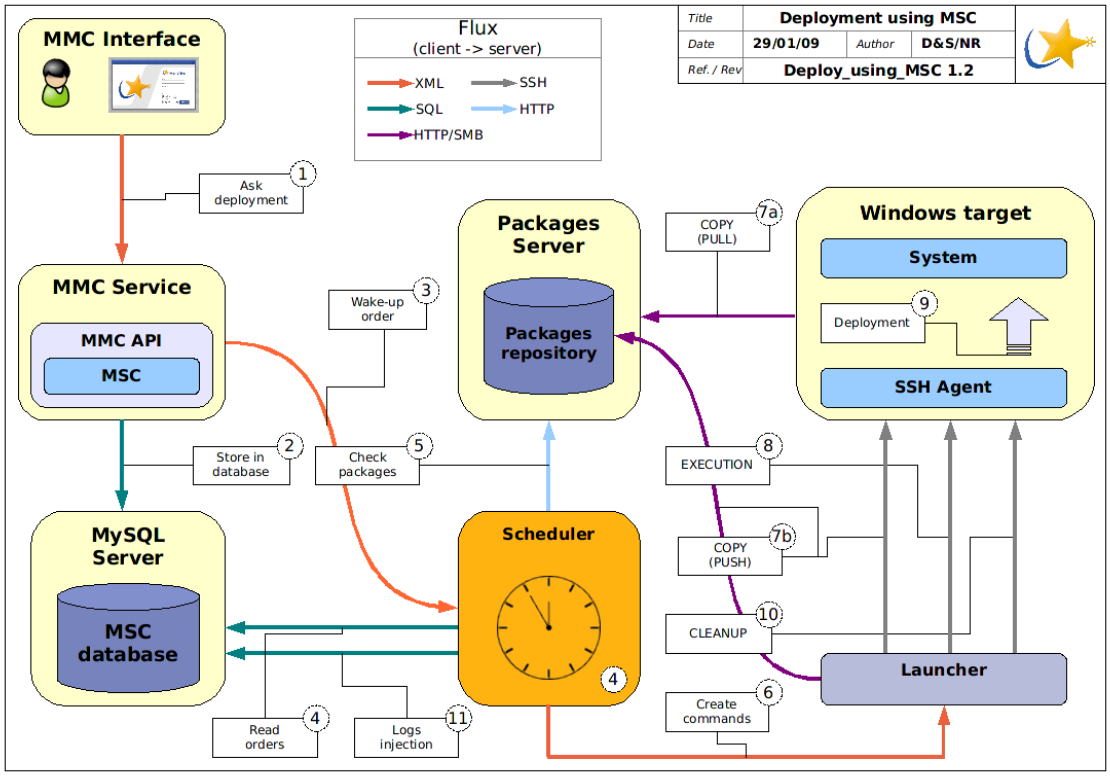


Illustration 1: Detailed application deployment workflow

1. The user enters a new deployment request and the **MMC service** checks that the package to be remotely deployed is present,
2. The order is written to the database,
3. The **scheduler** – launched periodically – consults the database,
4. and retrieves the characteristics of the deployment to be carried out,
5. It also checks that the package is present on the package server,
6. It then sends the list of orders to be carried out on the client to the **launcher**,
7. The **launcher** connects to the client and, initially,
   - Deposits the package (Push Mode), or
   - Asks the client to retrieve the package (Push/Pull Mode),
8. The **launcher** requests installation of the package,
9. The package is installed and the logs and return codes are retrieved by the **launcher**...
10. the temporary data are deleted ...
11. and the logs and error codes are uploaded to the **scheduler** to be written to the database.

Description of the User Interface

1. Access to the interface

The MMC (Mandriva Management Console) interface is accessed via a user ID and password. The web interface can be installed in a different location from the main service (MMC agent), which provides the possibility of connecting to different MMC agents (and therefore different Pulse 2 servers).

The interface can be accessed at the following URL: `http://mmc_server_ip/mmc/`

Illustration 2: MMC interface login window

This interface is compatible with Firefox and Internet Explorer 6+ browsers

2. Home Page

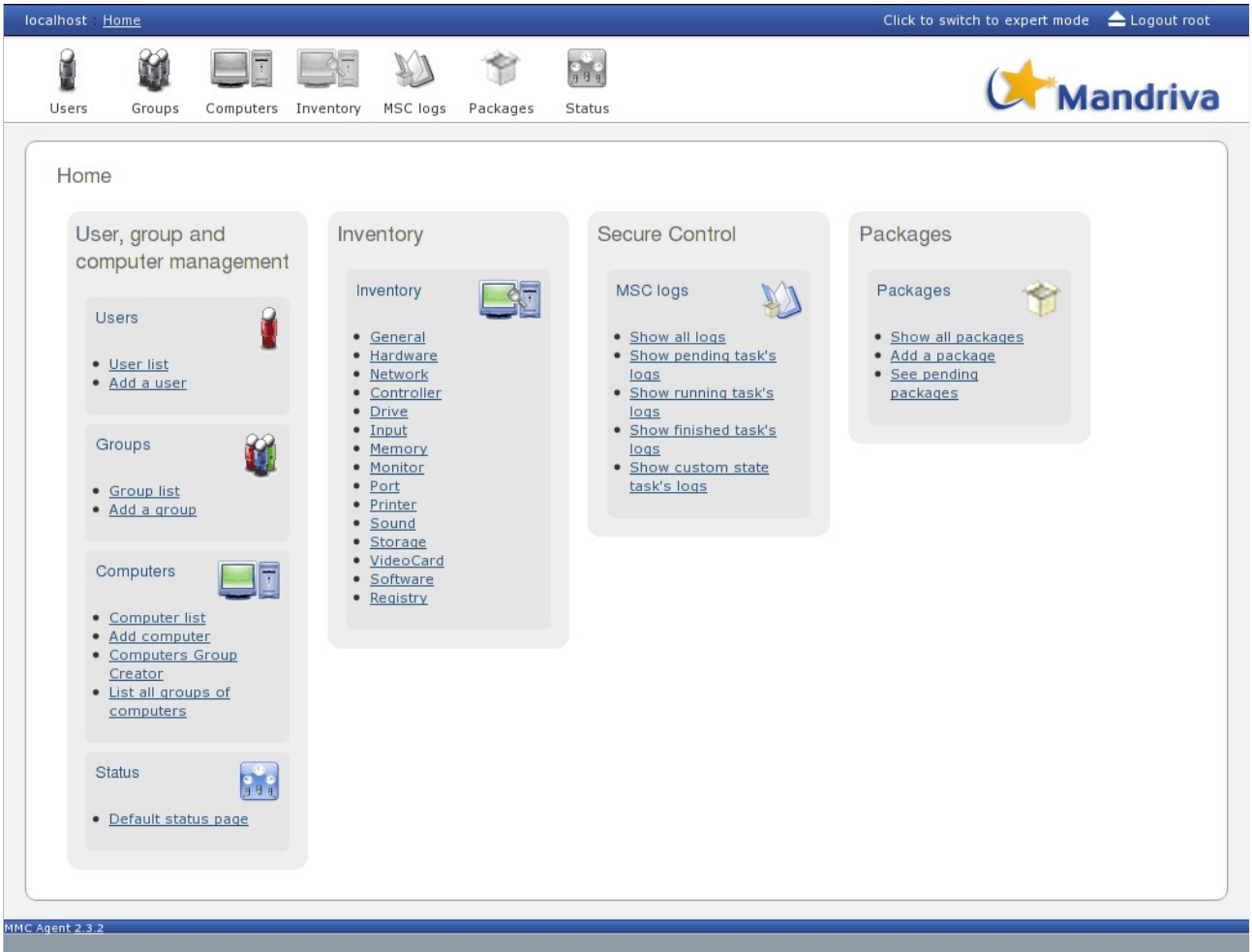


Illustration 3: MMC Home Page

The top bar contains the modules to which the user has access. Let us take the example of an administrator who has access to the complete functionality of the interface.

- Users and Groups module: management of users and access rights
- Computers module: list of machines, management of machine groups
- Inventory module: to view the software and hardware inventory
- MSC (Mandriva Secure Control) module: to view the deployment logs
- Packages module: management of Pulse 2 packages
- Status module: information about server status

The link used to disconnect from the interface is in the upper right-hand corner of the screen.

Expert Mode can be used to display additional information in the pages of the modules. The Pulse 2 modules used do not offer additional parameters in Expert Mode (the MMC framework used is common to Pulse 2 and Mandriva Directory Server).

3. Identification of machines

The list of machines is available in the Computers module via the '*All computers*' tab.

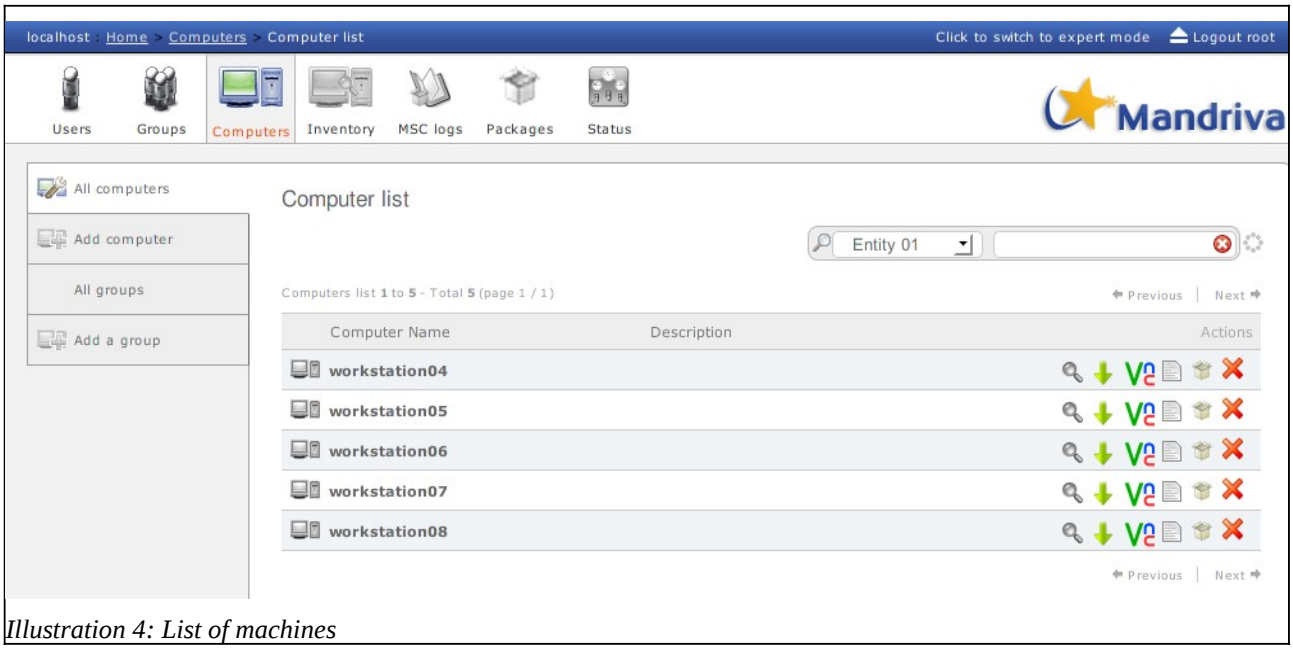


Illustration 4: List of machines

The machines in the scope of the user can be viewed as a whole entity or by selecting the appropriate entity from the drop down list located above the list of machines.

Several actions can be applied to each machine:

1. viewing the machine inventory
2. downloading the remote diagnosis
3. launching a VNC connection on the machine
4. viewing the history and logs of deployments performed on the client
5. deploying one or more packages on the machine
6. deleting the machine from the inventory

Access to these actions can be configured in the user profiles.

The list of machines is paginated (10 machines per page) and can be filtered by name using the field provided for this purpose above the machine list.

4. Management of machine groups

You can access the list of machine groups in the Computers module via the '*All groups*' tab.

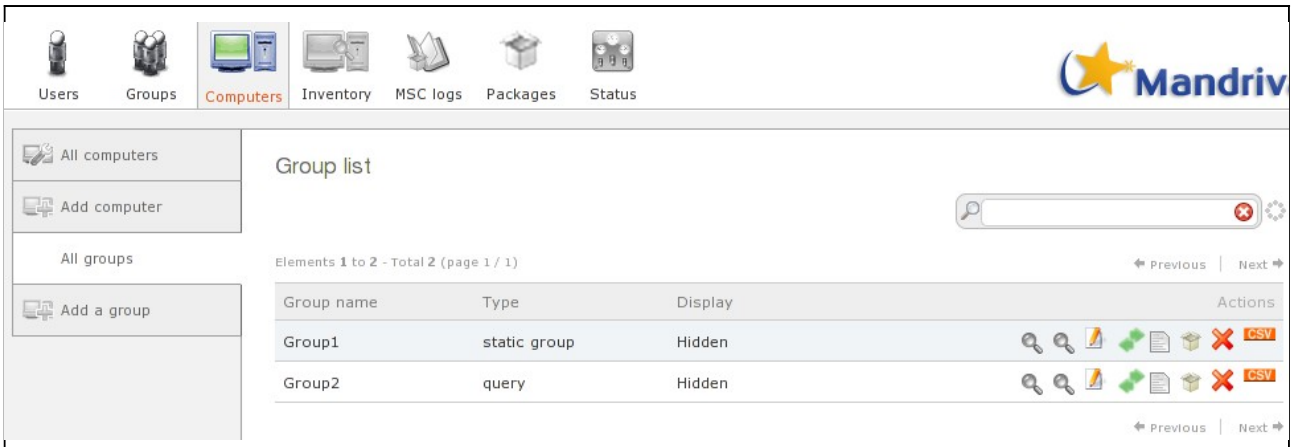


Illustration 5: List of groups

For each group of machines, you can:

1. look up the content of the group (list of machines in the group)
2. look up the inventory of the machines in the group
3. edit the group
4. share the group with other users or groups of users
5. view the history and deployment log of the group of machines
6. deploy one or more packages on the group
7. delete the group
8. CSV export a CSV file containing the main information (displayed on the computer list page) of the machines in the group

Access to these actions can be configured in the user profiles.

Creating Machine Group

The group management module is located in the Computers module.

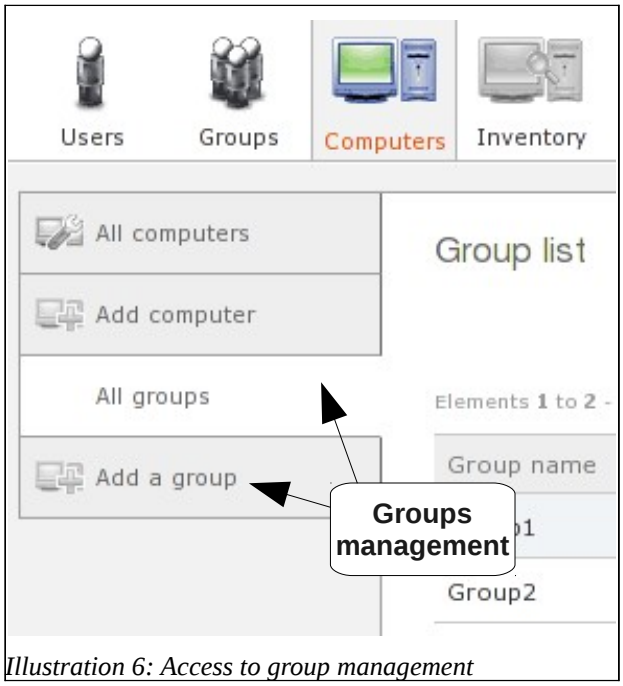


Illustration 6: Access to group management

From the Pulse 2 interface, you can manage two types of group:

1. Static groups,
2. Dynamic groups.

Groups belong to the user who created them. They can, however, be shared with another user.

1. Static groups

A static group is a group of machines selected arbitrarily from a list.

The group is identified by a freely-chosen name.

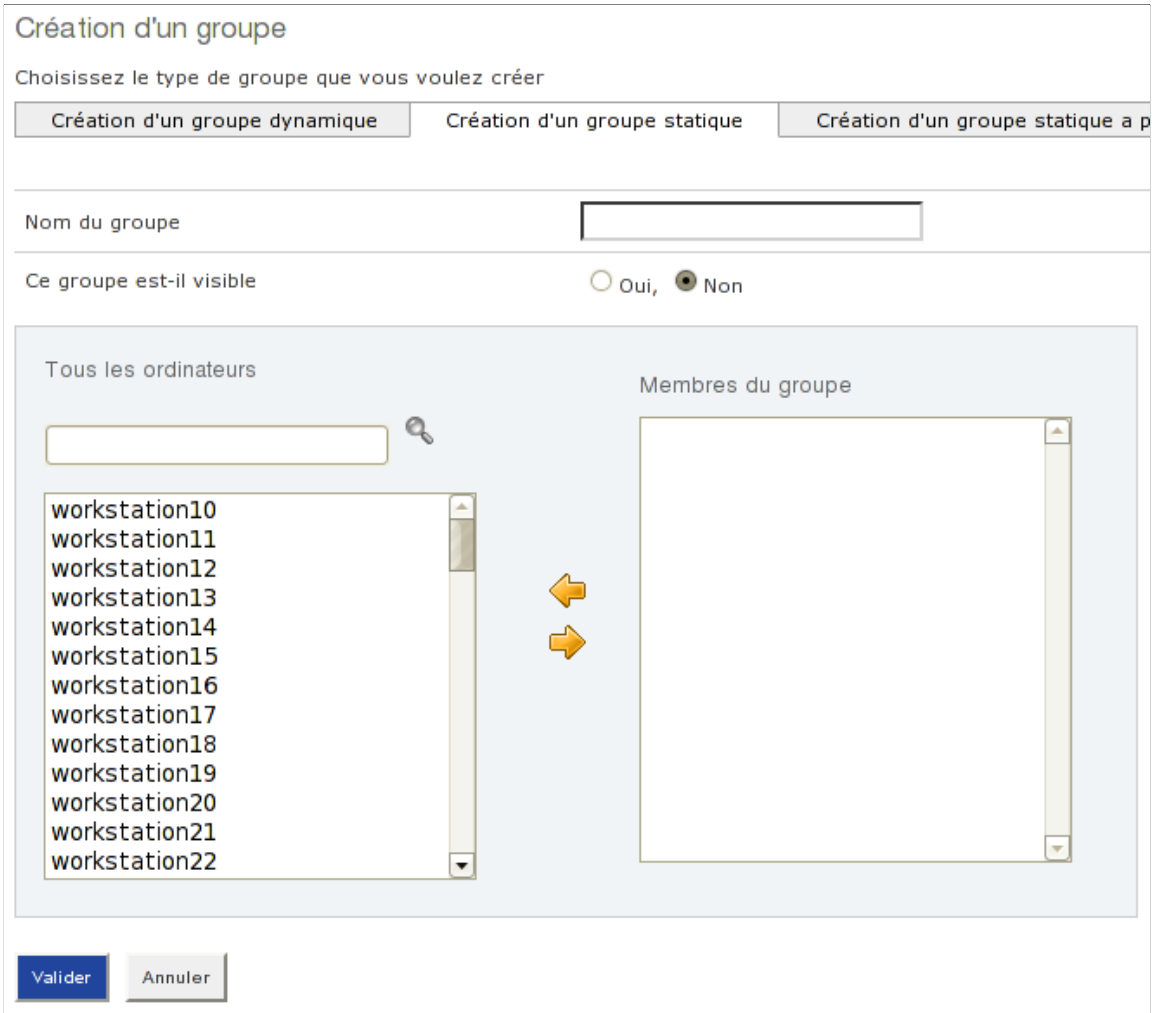


Illustration 7: Creating a static group

1.1. Visible group

To make a (static or dynamic) group visible, you must create a tab bearing the name of the group on the left-hand side of the screen. This tab gives you a shortcut to access the content of the group without through the groups list.

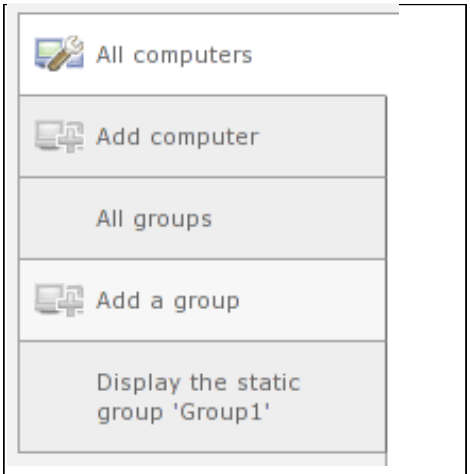
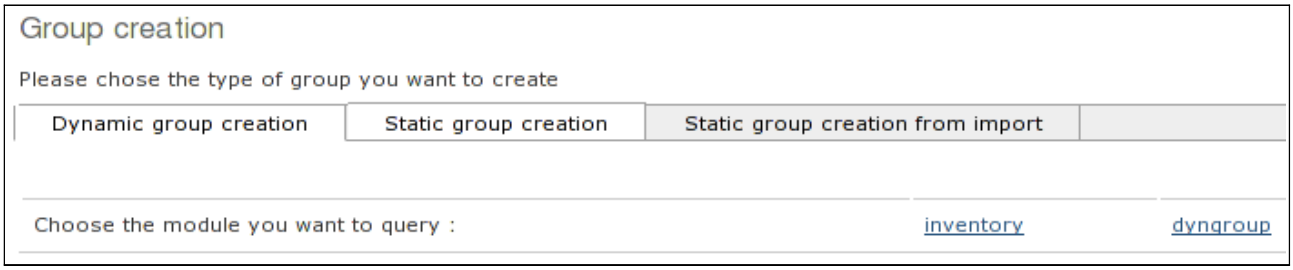


Illustration 8: Group access shortcut tab

2. Dynamic groups

A dynamic group lets you create a group of machines using criteria from the inventory database (**inventory**) and the group database (**dyngroup**).



In this case, the machines are not chosen manually, but according to (search) criteria to be defined in the query.

When a dynamic group is created, it is saved in Query Mode or Result Mode.

Queries versus results

There are two adjacent concepts:

- Dynamic groups in Query Mode
- Dynamic groups in Result Mode

Queries are true dynamic groups, for which the result is never saved: the query is made each time. The main advantage is obvious: the result uploaded is always up to date with respect to the inventory.

Results are semi-dynamic groups. The query is always memorised, and so is the result. The main advantages are greater calculation speed (the result is already stored in memory) and greater malleability (a machine can be deleted from a result, but not from a query).

A dynamic group in query mode can be changed to result mode, and vice versa.

Note: when all machines have been deleted from a result, it becomes a query.

Creating a dynamic group

On the dynamic group creation page, select the module to be queried (**inventory** or **dyngroup**) to display the criteria on which our query can operate.

The **inventory** module supplies the following fields *by default*:

- Software/ProductName: name of the software
- Registry/Value/DisplayName: name of the machine
- Hardware/OperatingSystem: operating system
- Hardware/ProcessorType: processor in the machine
- Drive/TotalSpace: hard disk space

The **dyngroup** supplies the following field:

- groupname: name of an existing Pulse 2 group

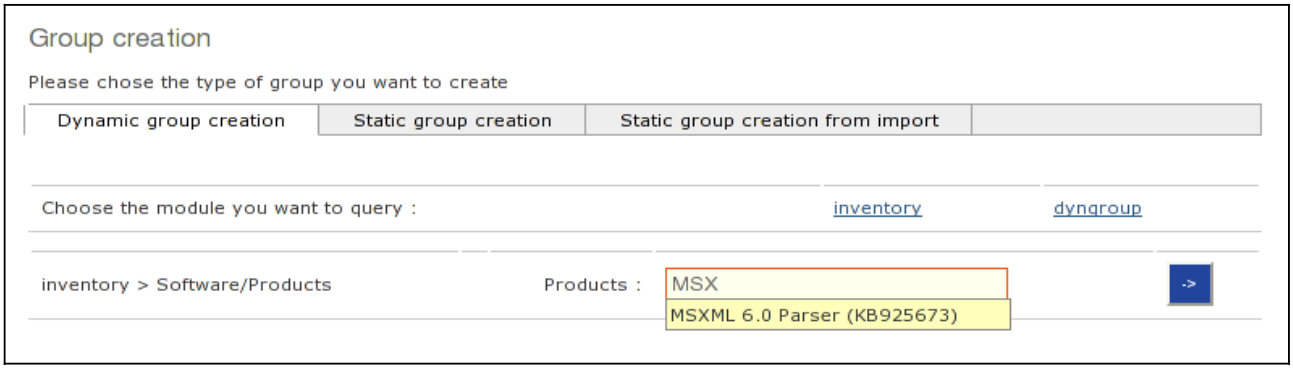
A dynamic group is created by defining a certain number of searches from the fields listed above.

The final query can be consolidated by a Boolean expression between the various searches selected.

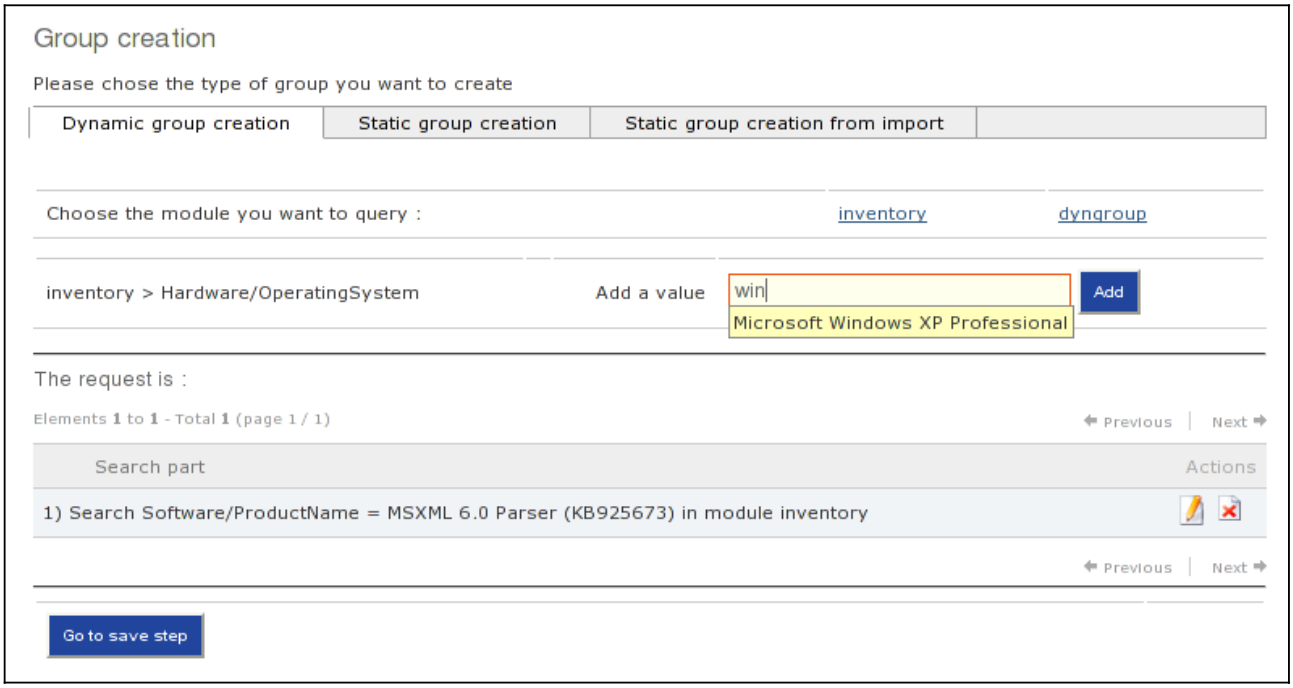
Example

We will search machines running Windows and having MSXML 6.0 software.

We can start by searching the MSXML software. When we enter the value (at least three characters), we are given a list of choices.



In the same way, we add a criterion on the OS by selecting the Hardware/OperatingSystem field:



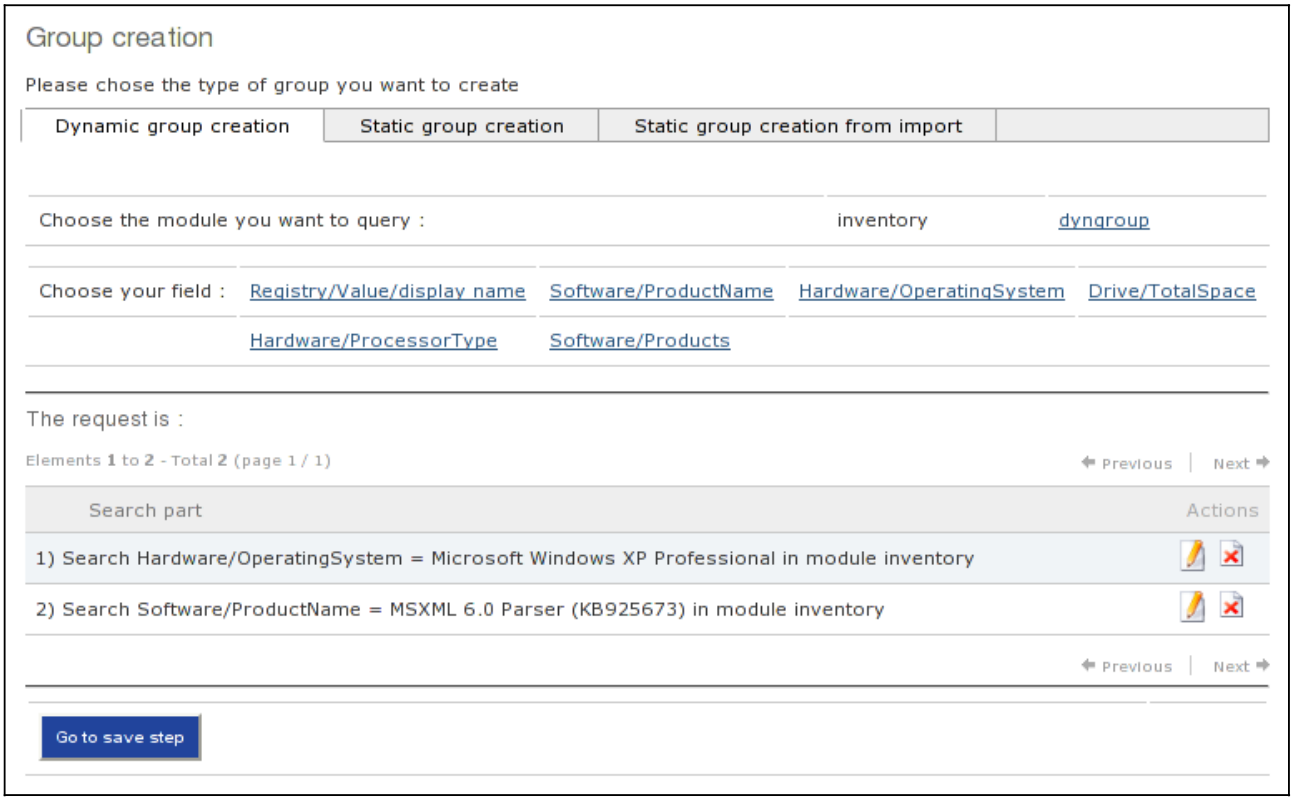
Note that a search can be deleted or edited by clicking the appropriate icon in the action column.

*Using a wildcard **

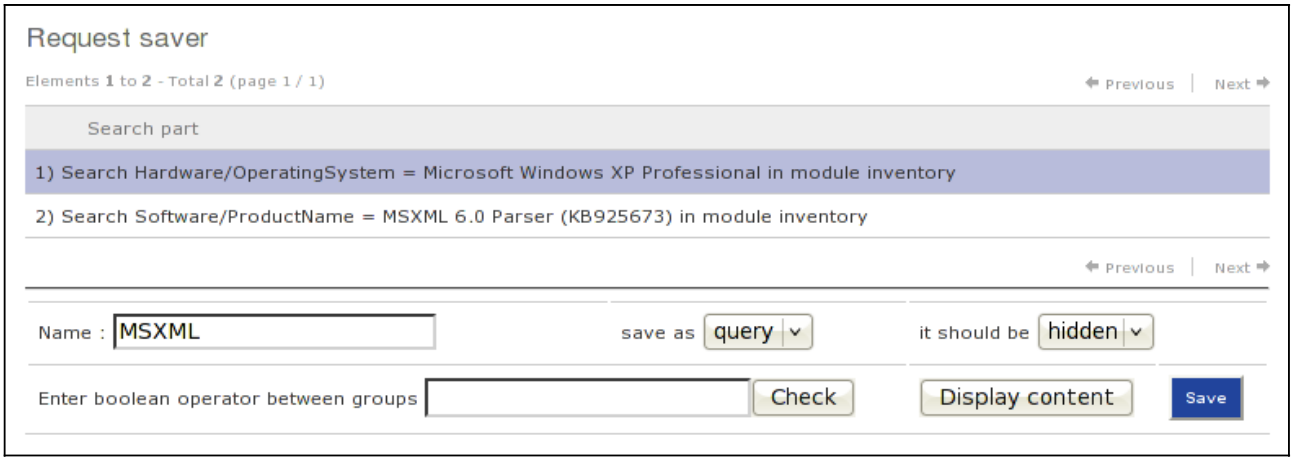
Note that the wildcard * is supported and can be used to perform a more general search on a criterion. For example, the following value can be specified for the OS field: **Microsoft***

In that case, all versions of the Microsoft OS will be uploaded (XP, 2000, NT4 etc.).

The detail of the searches we have added is continuously displayed.



When our searches have been defined, we can go on to the next stage: saving the group.



The choice between Query Mode and Result Mode must be made on this screen.

There is also the possibility of creating a tab for the group, or not, by making it visible or hidden.

The query can be consolidated by a Boolean expression.

3. Boolean Expressions

When creating a dynamic group, if (and only if) several searches have been input, the rules to be applied between the result lists can be specified (one list per search).

The possible operators are AND, OR and NOT.

- The AND (or intersection) operator returns elements which are present in all the lists it has operated on.
- The OR (or union) operator returns all elements which are present in at least one of the lists it has operated on.
- Both of these operators make sense only if several lists have been run past them.
- The NOT operator applies to just one list. It returns all possible elements except those in the list it has operated on.

By default, an AND is applied between all the lists, i.e. only machines present in all result lists will be included in the final result.

Every search has an identifier within the group (specified in the far left column). This identifier allows you to build the result consolidation Boolean expressions.

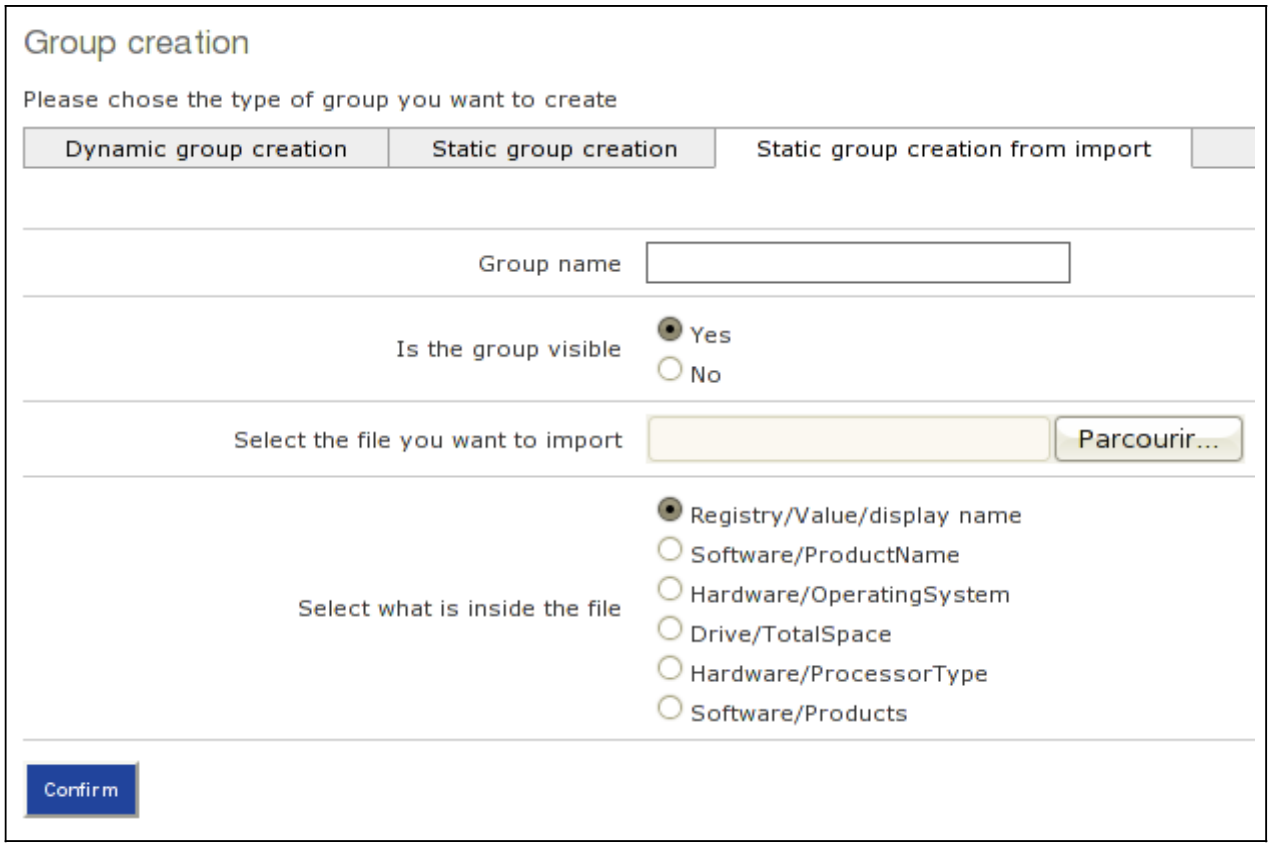
The syntax of this expression uses prefixed notation, i.e. the operator is placed before the parameters. An operator (AND or OR) must encompass all the parameters. Expressions such as the following were therefore obtained:

- AND(1, OR(2, 3))
- OR(4, AND(2, 1), 3)
- AND(NOT(1),2)

4. Creating a static group from an import

From the group creation interface, there is a screen to let you build a static group from a file containing a certain type of data.

The import file must contain a unique list of data.



The data contained in the file are limited to one type from the list of fields offered.

By default:

- One or more software names,
- One or more machine names,
- One or more OS names,
- etc.

5. Sharing a group

When a user creates a group, it belongs to that user and is visible to that user only. An interface lets you share your groups with other users.

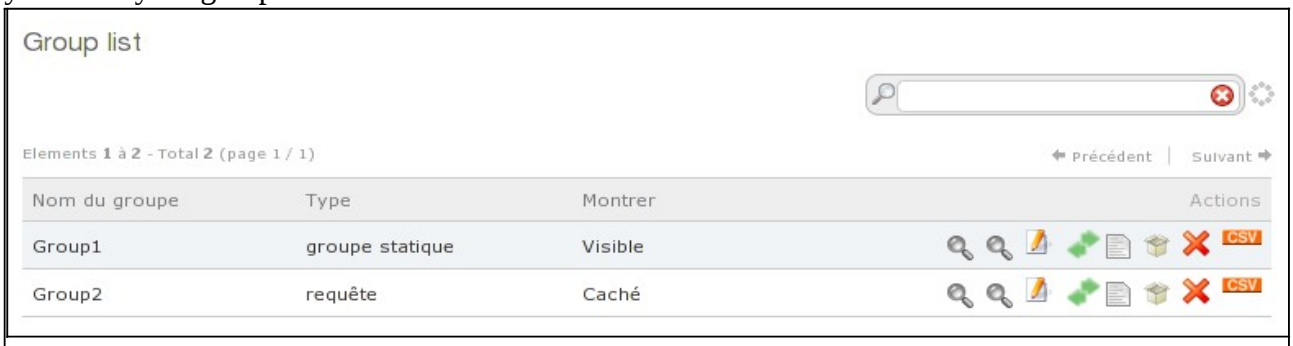
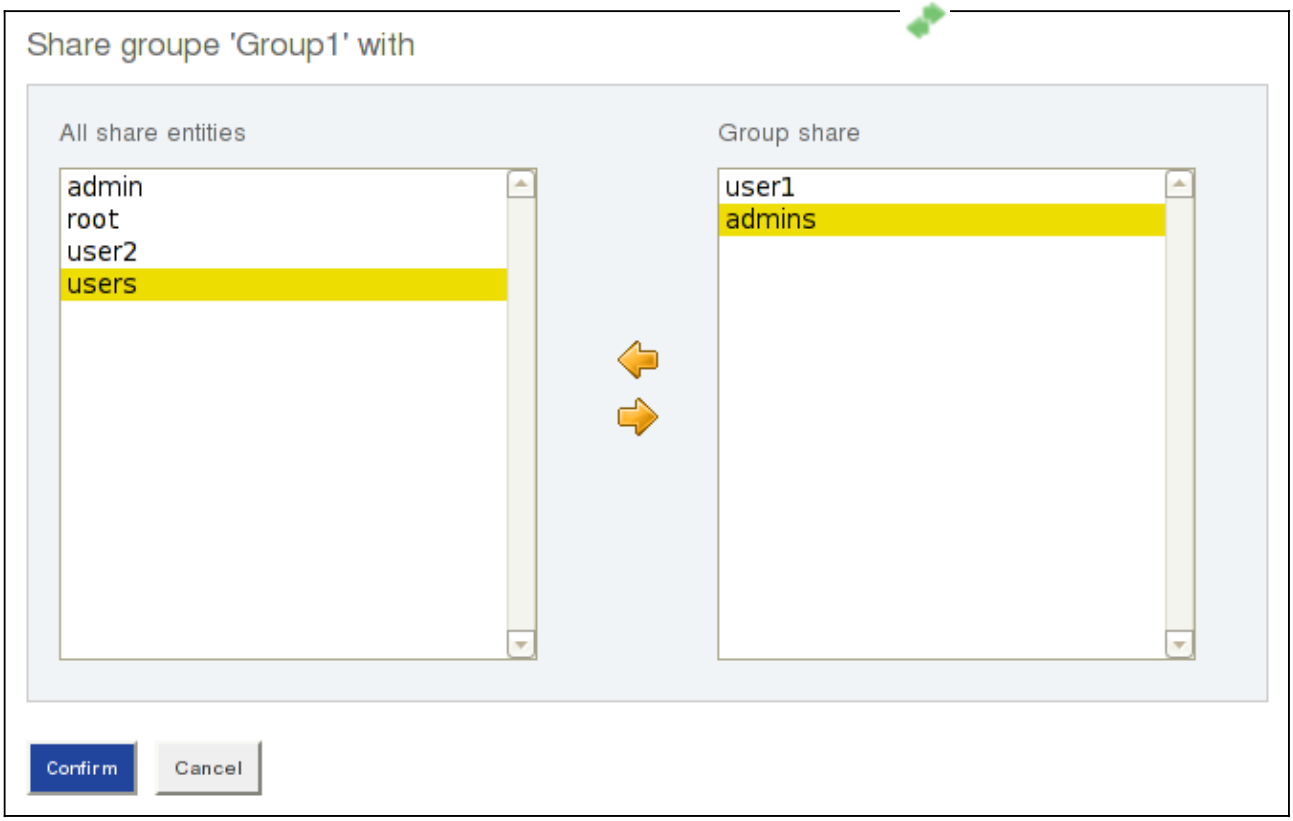


Illustration 9: List of groups

From the list of groups, this interface can be accessed using the icon

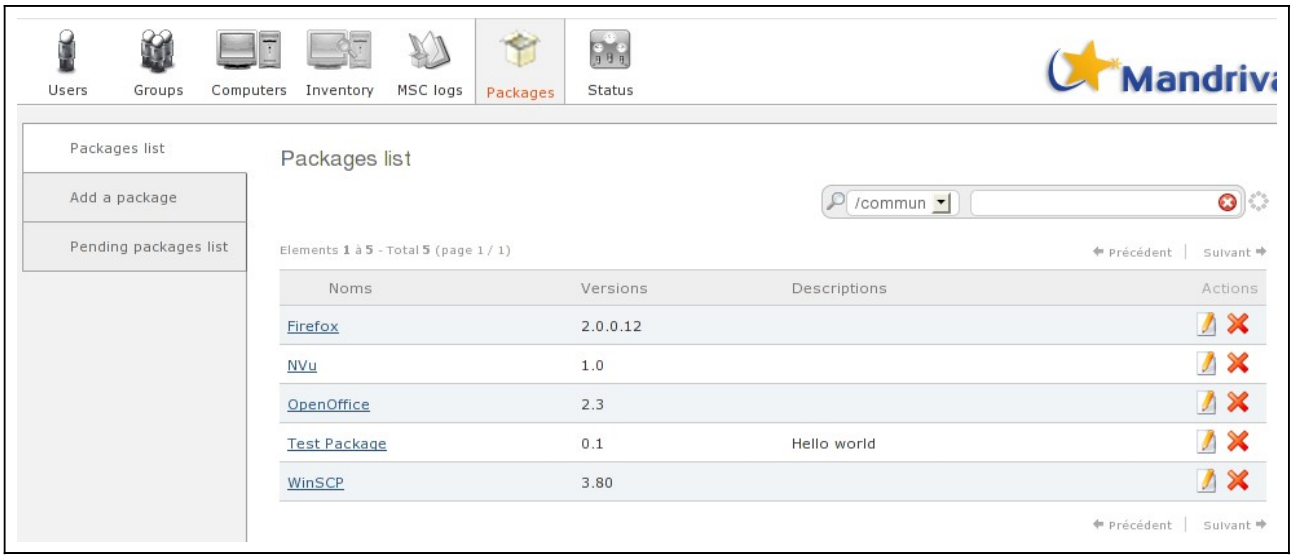


The group can be shared for a user or a group of users.

In the above example, Group1 will be shared with user1 and with the 'admins' group. The users with whom the group is shared have the same rights as the user who created the group, expect to delete the group.

Package Management

The package management interface is accessible from the Packages module.



Packages are filed in one or more repositories. (/commun above)

1. Pulse 2 package concept

Two conditions must be fulfilled for a package to be installed correctly from Pulse 2:

- The package must be installed in silent mode (no information or window displayed on screen)
- The package must return a correct return code (0 for a success, any other code for an error)

As a general rule, a package is installed in silent mode by applying additional options to the installation software.

For example, silent deployment of an InnoSetup package is always performed as follows:

```
setup.exe /verysilent
```

2. Adding a package

A directory containing the package installation files is loaded manually (scp, ftp, smb, etc.) onto the package server, in a directory intended for this purpose (according to the configuration of pulse2-package-server, by default: /tmp/package_tmp/put1/ and /tmp/package_tmp/put2/ respectively for the put1 and put2 APIs).

When the files have been loaded onto the server, the package must be added from the Pulse 2 interface via the 'Add package' tab.

For example, the package created belongs to the *nvu* software. The files loaded onto the server are:

```
nvu/                                    # folder
 |--nvu-1.0-win32-installer-full.exe     # app. Install. binary
 `--install.bat                          # package installation script
```

From the Pulse2 package management module, the package is added in two stages. Firstly, certain information about the package must be filled in:

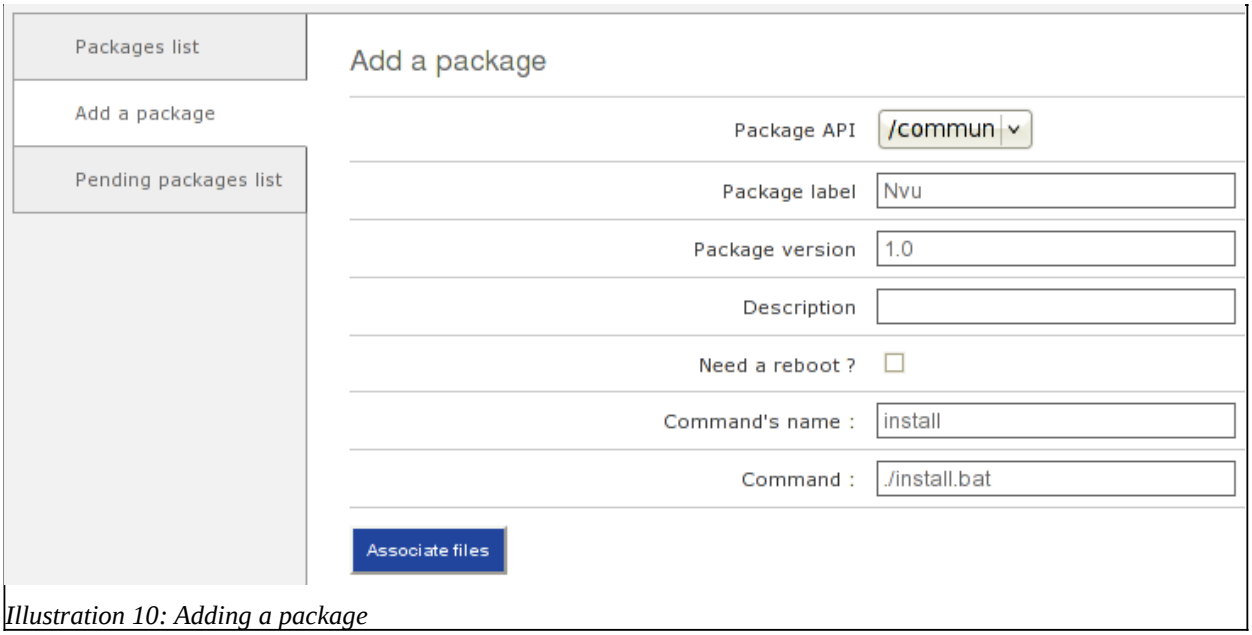


Illustration 10: Adding a package

The information requested is used to characterise the package for Pulse 2:

- Choice of package API (package repository, which can be different depending on the OS),
- The label of the package will be the name displayed in the Pulse 2 interface,
- The version will be the value displayed in the Pulse 2 interface,
- The description is optional and is also displayed in the interface,
- You may specify whether the computer should be rebooted after deployment of the package,
- The 'name of command' field is not yet handled,
- The installation command describes how the package should be installed on the target machine(s). A bat (or sh) script is generally used; in this case, the command will be of the following type: **./nameofscript.bat** (where nameofscript.bat is a bat script present in the files of the package previously loaded).

When this information has been entered, the next screen lets you associate the files of the package

that have already been loaded onto the server to constitute the final package. Simply select the folder containing the files of the package to finish adding the package.

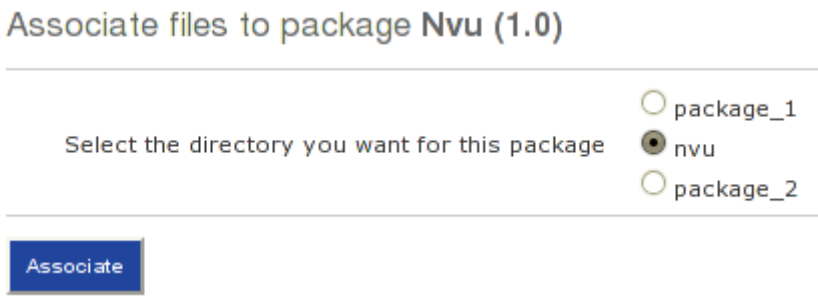


Illustration 11: Associate files

If you have several package servers, the package will be replicated on all configured package servers. As long as the new package is not replicated, it is placed in the queue:

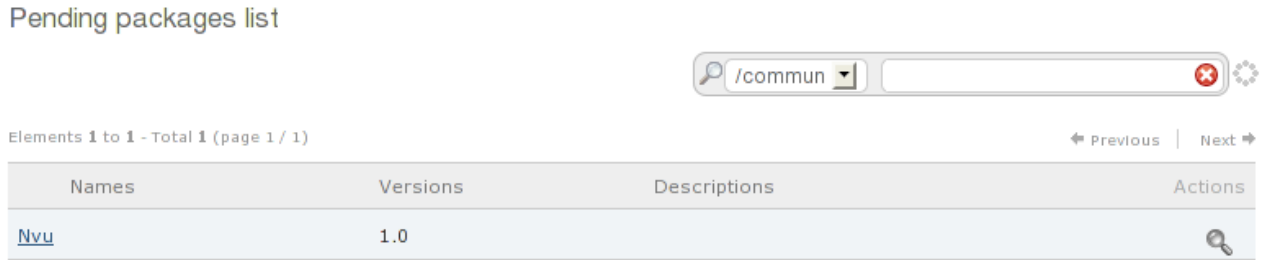


Illustration 12: Pending packages list

3. Editing a package

The package editing function does not let you modify the contents of a package (the installation files) or its package API (repository).

You can, however, rename the package or change the package installation command.

4. Summary of package creation procedure

1. Create a bat or sh script to launch the software installation in silent mode.
2. Load the application installation files with the script on the package server into a file (e.g., using scp), and assign appropriate rights to the files.
3. Declare the package in the 'Packages' module of the Pulse 2 console and associate the previously loaded folder with it.
4. The package appears in the list of packages when all the package mirrors are synchronised (only 1 repository by default).

Deploying Packages

1. Deployment

Package deployment can be performed on a machine or a group of machines.

- To deploy a package on a machine, proceed via the list of computers ('All computers') in the Computers module.
- To deploy a package on a group of machines, proceed via the list of groups ('All groups') in the Computers module.

In either case, deployment is launched in the same way, using the deployment icon  :

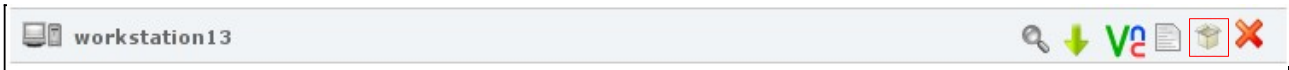


Illustration 13: Deploying packages on a machine

Illustration 14: Deploying packages on a group

The deployment page lists the packages available for the group or machine:

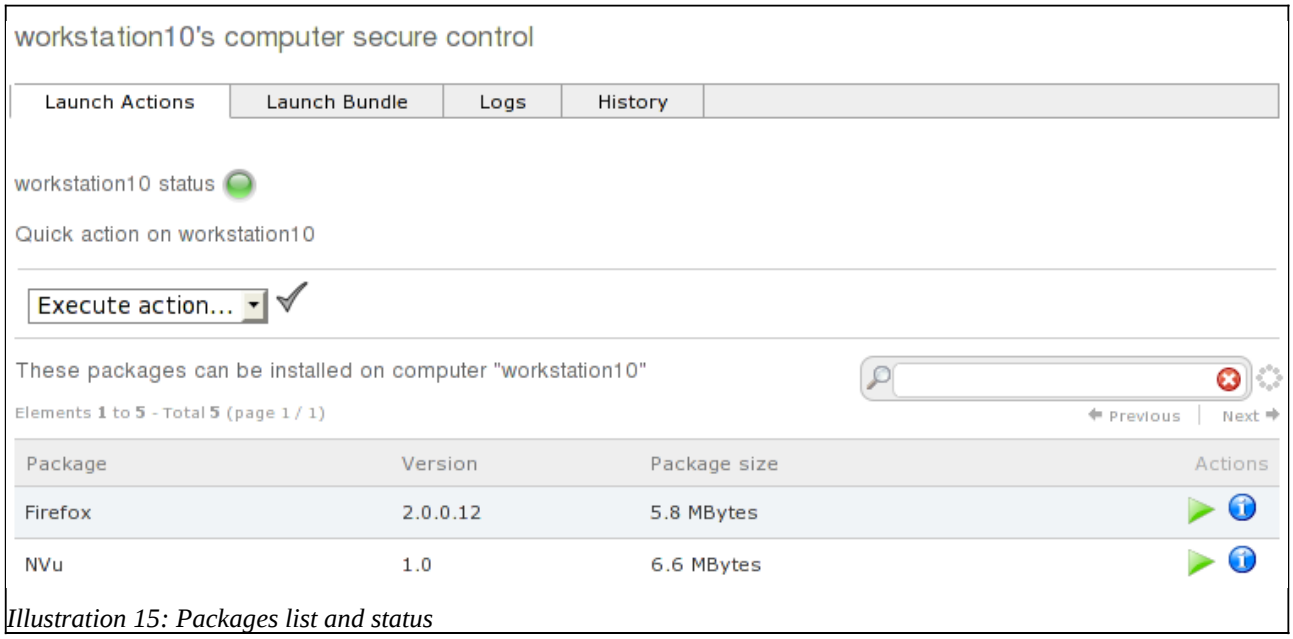


Illustration 15: Packages list and status

When only one machine is selected, an indicator displays the machine status:

- red: no ping
- orange: ping, but no SSH
- green: ping and SSH ok

If 'ping' fails but the machine is accessible via SSH, the indicator remains red, but package deployment will work.

This information is provided as an indication only, and must not be applied absolutely to the letter (for example, the ssh probe time could be longer than the maximum time allocated to the test).

1.1. Quick deployment

From the list of packages, the icon is used to launch an immediate (one-shot) deployment.

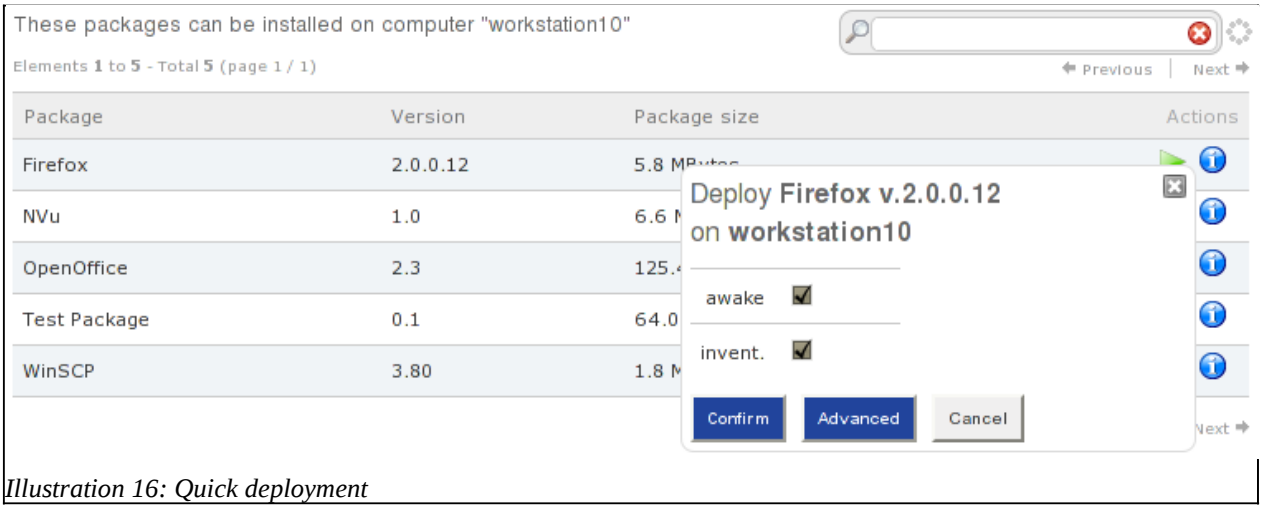


Illustration 16: Quick deployment

A popup lets you activate or not activate WOL and inventory uploading.

Deployment starts as soon as the user clicks the 'Confirm' button.

1.2. Advanced deployment

Click the 'Advanced' button in the fast deployment popup to set several parameters.

You can change parameters as follows:

- Specify parameters for the installation script
- Perform a WOL if the connection fails
- Program the start of deployment
- Program the deployment end date (aborts all unfinished deployments)
- Program the time slots during which deployment can occur (deployment interval)
- Start the script (default value: **yes**)
- Delete uploaded files after a successful command (**yes**)
- Carry out an inventory after a successful command (**no**)
- Shut down the machine after a successful command (**no**)
- Number of connection attempts
- Time between two connections (minutes)
- Maximum bandwidth (b/s)
- Deployment by local proxy

Deployment interval

The deployment interval lets you specify the time slots during which deployment can be performed.

- The time slots are separated by commas (,)
- A time slot contains two limits separated by a hyphen (-)
- A limit is a string with the format HH or HH:MM or HH:MM:SS.

Examples:

```
2-4                     : deployment will be from 2am to 4am
3:00-5:00,18:00-20:30   : from 3am to 5am and from 6pm to 8.30pm
```

Example of discontinuous deployment

An administrator wants to deploy a package for a period of 3 days and only from 1am to 6am. The deployment should begin the first day at midnight and ends two days later at 7am.

In this case:

```
Beginning of the deployment : 23/04/2009 00:00
End of deployment           : 23/04/2009 07:00
Deployment interval         : 01-06
```

Deployments that are being discontinued at the end of the deployment (here, 6 am) are placed in "re-scheduled" state. They are automatically restarted when the next deployment interval begins (1am the following day).

When the end date of deployment is reached, the deployments are being interrupted and placed in the "Failed" status.

1.3. Deployment of package bundles

A **bundle** lets you deploy several packages on a machine or group. The user can set the dependencies (order of installation) between the packages in the bundle.

Start deployment of a bundle by selecting the packages in the bundle. A bundle cannot yet be saved; it must be redefined at each deployment.

After choosing the packages to be contained in the bundle, specify the installation order of the packages in the appropriate screen.

Taking an example with three packages:

- *package1*
- *package2*
- *package3*

First case: the following order of installation is chosen:

1. *package1*
2. *package2*
3. *package3*

package2 will be deployed after *package1* has been correctly deployed. *package3* will be deployed after *package2* has been correctly deployed. If the deployment of *package1* fails, *package2* and *package3* will not be deployed.

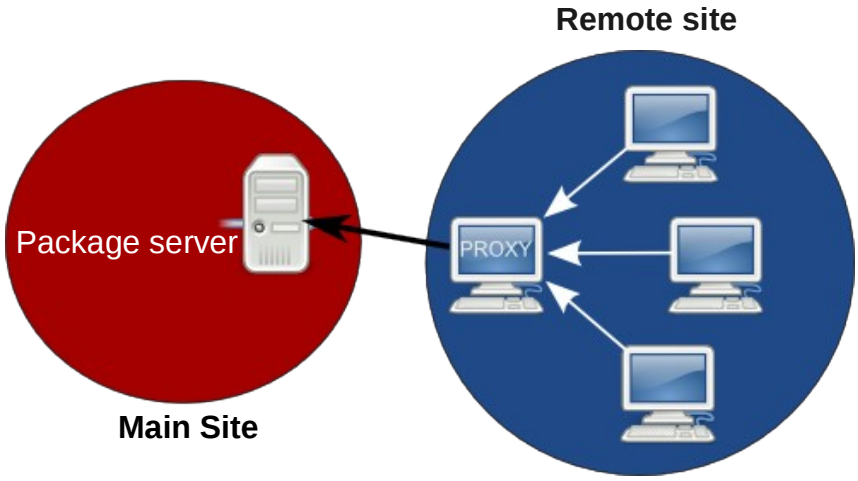
Second case: the following order of installation is chosen:

1. *package1*
1. *package2*
2. *package3*

package1 and *package2* will be deployed with equal priority. If, and only if, both of these packages are correctly deployed, then *package3* will be deployed.

1.4. Deployment by local proxy

This type of deployment is used for remote sites with a few machines where it is not possible to install a package server. Local proxy mode minimises the use of bandwidth between the remote site and the main site.



During a classic deployment, each machine concerned obtains the package to be deployed from a package server. In local proxy mode, only one machine (the proxy) downloads the package, and the other machines download the package from this proxy machine.

Deployment by proxy is performed on a group of machines containing only the machines of the remote site. To deploy a package by local proxy, simply select the appropriate option on the advanced deployment form.

When the user confirms deployment, there is an additional step to select the possible local proxy machines. At least one machine must be chosen. Pulse 2 will use the first machine it is able to contact as the local proxy.

A deployment using local proxies can only be performed on a group of computers in advanced mode :

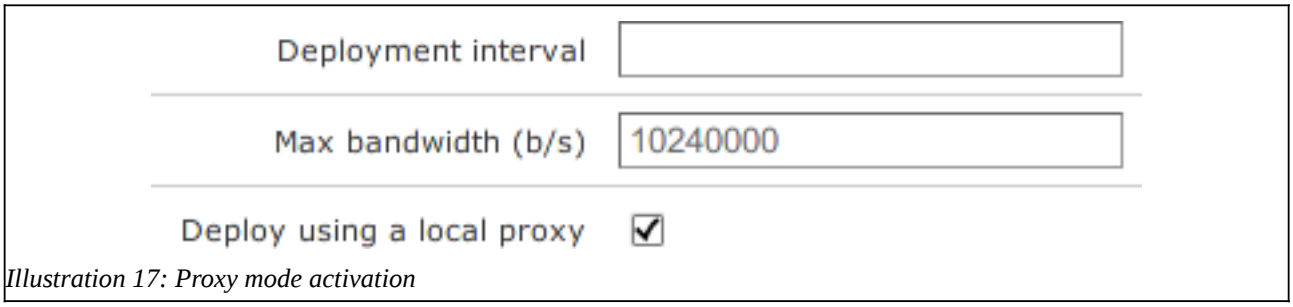


Illustration 17: Proxy mode activation

During a deployment using local proxies, the group should contain only machines belonging to the same remote site.

The next screen allows you to specify the proxy mode (single or multiple) :

- In simple mode Pulse 2 will use the first machine, if it is unavailable, the second machine from the list, etc ...
- In multiple mode different proxies are used in parallel.

The recommended method is the multiple mode.

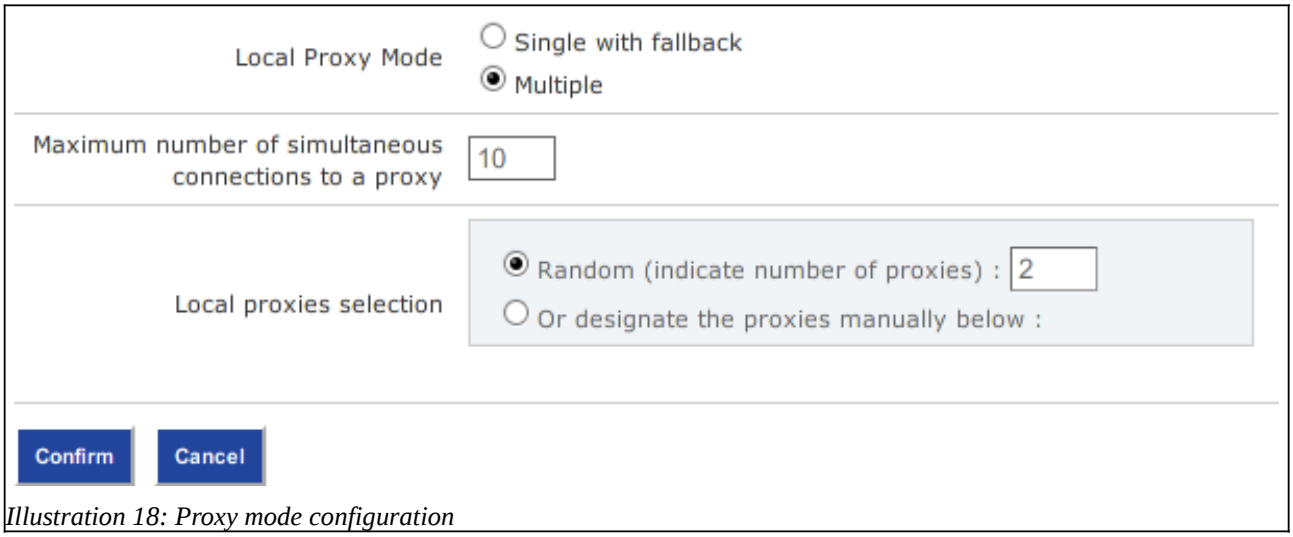


Illustration 18: Proxy mode configuration

The interface allows you to specify (manually or automatically) the number and potential candidate computers to the role of local proxy and the maximum number of simultaneous clients using a proxy. The deployment is recorded as usual after clicking OK.

Do not exceed 10 simultaneous connections by proxy !

2. Monitoring a deployment

When a deployment is launched (whether on one machine or a group of machines), the user is automatically transferred to the deployment detail page. Click the 'Logs' tab to monitor the deployment in real time. Deployment is monitored at three main phases: sending of the package, installation, and deletion of the installation files. There are also additional optional WOL and inventory uploading phases.

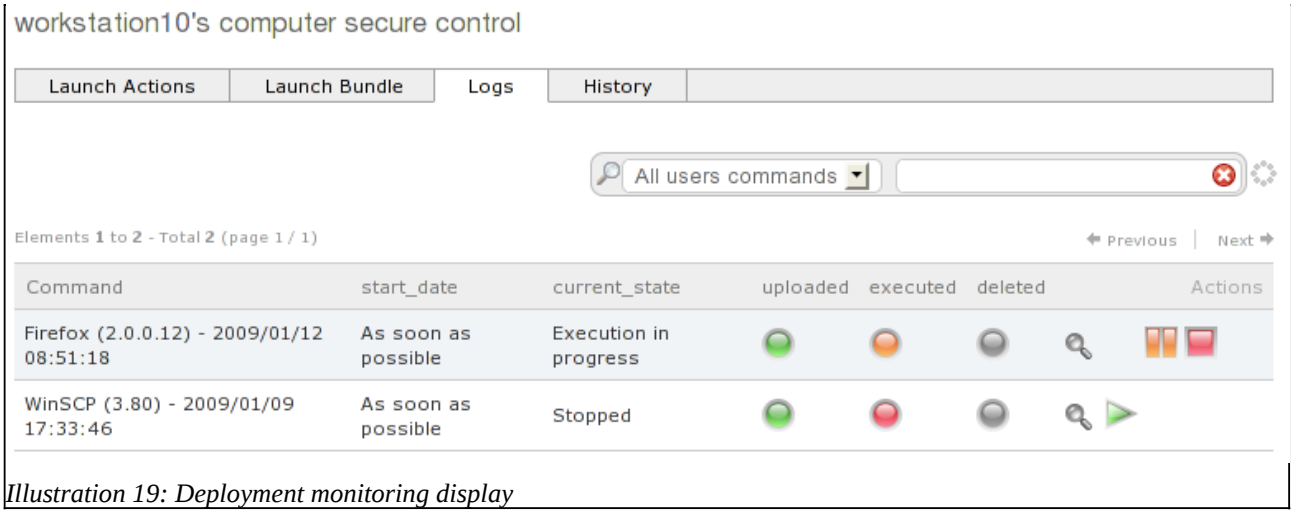


Illustration 19: Deployment monitoring display

The phase indicators are updated every 30 seconds on the interface (you can also refresh the page to force an update). When a deployment is complete, it can be seen in the 'History' tab. All deployments in progress can be seen in the 'Logs' tab.

Deployment status

- **WOL in progress**: deployment is in Wake On LAN phase
- **Sending in progress**: deployment is in sending phase
- **Execution**: deployment is in execution phase
- **Deletion in progress**: deployment is in deletion phase
- **Inventory in progress**: deployment is in inventory phase
- **Completed**: deployment has been successfully completed
- **(Sending, Execution or Deletion) failed**: deployment has failed and used up all of its attempts (3 by default)
- **Scheduled**: deployment is programmed and no attempt has yet been made
- **Rescheduled**: deployment has failed in one of the three deployment phases, but will retry later because some unused attempts remain
- **Aborted**: the user has interrupted deployment manually via the corresponding action
- **Pause**: the user has paused deployment manually via the corresponding action.

Monitoring the deployment phases

- orange: phase in progress
- green: phase successfully completed
- red: phase failed

Actions on deployments

The  button pauses deployment. Deployment is interrupted at the end of the current phase. For example, if execution is in progress and deployment is paused, the deployment will stop at the end of the execution phase.

A  button is used to resume deployment.

The  button halts deployment completely.

2.1. Deployment details

The icon gives access to the detailed deployment log.

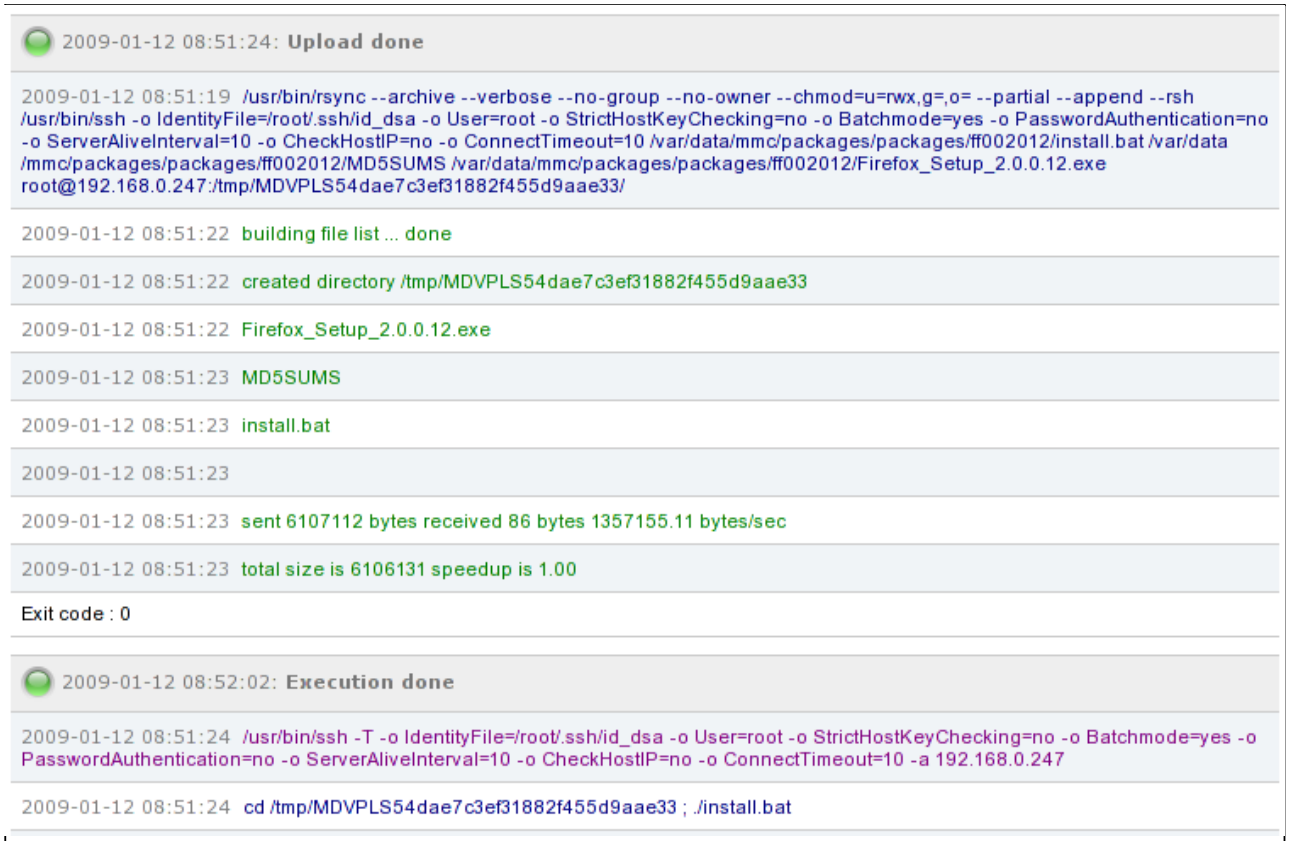


Illustration 20: Deployment log

The deployment log lists the three main phases and all commands executed during deployment.

The text responses from the commands executed are uploaded into the log. These responses are written in red if an error appears to have occurred (stderr). Successfully executed commands are shown in green (stdout).

Commands executed by the launcher are in blue.

Each command sent by the launcher is time-stamped (date HH:MM:SS) as an aid to deployment monitoring.

2.2. Application deployment history

The history of completed or current deployments can be viewed for a particular machine or group via the Computers module.

To view all the deployments performed on a system, you must use the MSC module. Using this module, application deployments can be viewed by status (Completed, In progress, Failed, etc.) on the whole network.

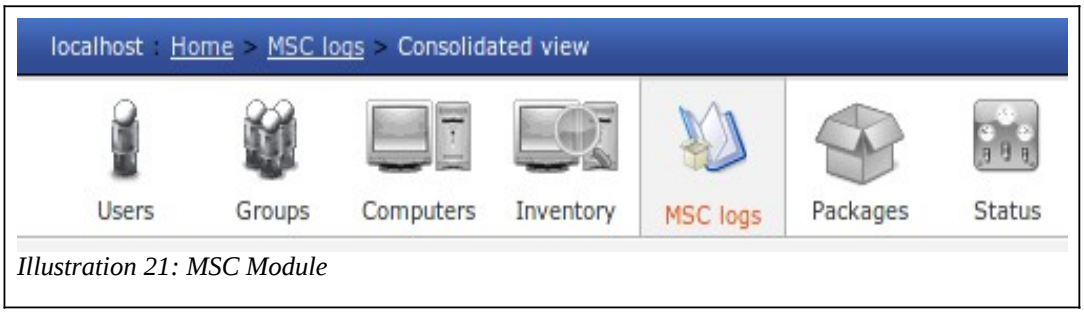


Illustration 21: MSC Module

In this module, several views are available:

- consolidated views of the deployments
- all commands
- the waiting commands
- the currently running commands
- the terminated commands
- commands sorted by status

To well understand the displayed information on these pages, two notions must be introduced:

- **command** or **action**: in Pulse2's interface, a command is equivalent to a package's deployment on a computer.
- **Deployment**: a deployment contains one or more commands. For example, a deployment of a bundle creates as much commands than the number of package in the bundle.

Consolidated deployment view

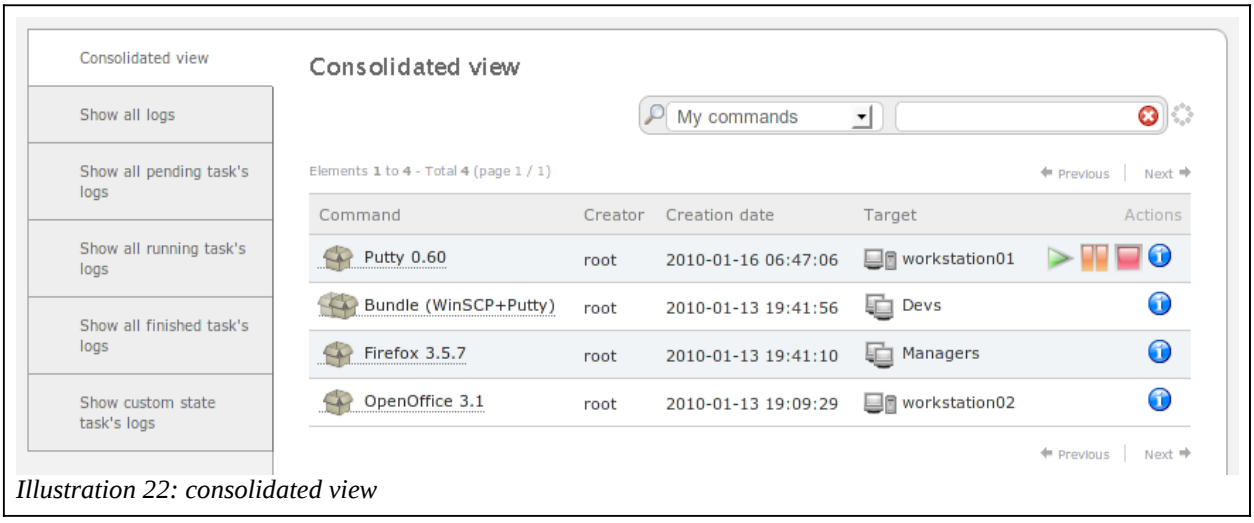


Illustration 22: consolidated view

In contrast with the other available views, the consolidated view displays a simple overview of the last deployments. Each line represents the deployment of a package or a bundle on a computer or a group of computers.

The icons let you quickly know what has been doployed and the target of the deployment:

 a package

 a bundle

 a computer

 a group of computers

The action  lets you display the status of the deployment. By clicking on the name of the deployment (the package or the bundle), you can analyse the various commands of the deployment.

Other views

The other views of the MSC module display the commands that have been executed by Pulse2 on your network according to the desired state.

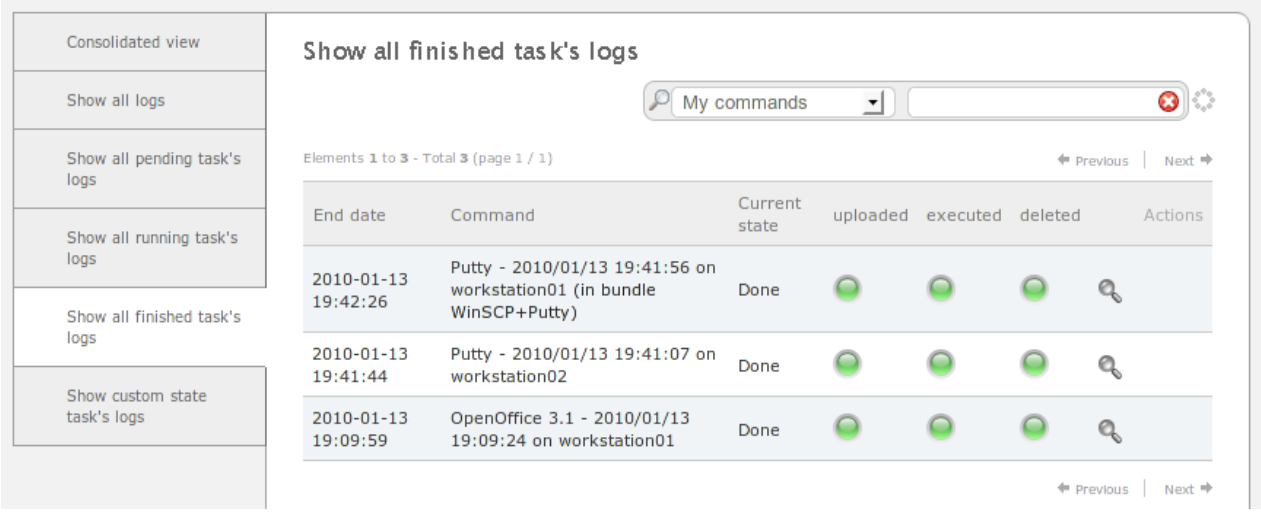


Illustration 23: MSC: Terminated actions

Every line describes a command. In consequence, it is difficult to know if, e.g., this command is part of the deployment on a group or to link the other commands of the same deployment. To have a more global vision, you can use the consolidated view.

2.3. Installation reports

When packages are being deployed on a group or during deployment of a bundle of packages, the classic indicators (diodes) are not present and would not be useful, given the amount of information to be represented (several packages and several machines).

For these two types of deployment, Pulse 2 provides a detailed installation report accessible using the icon. It lists precise information concerning your deployment.

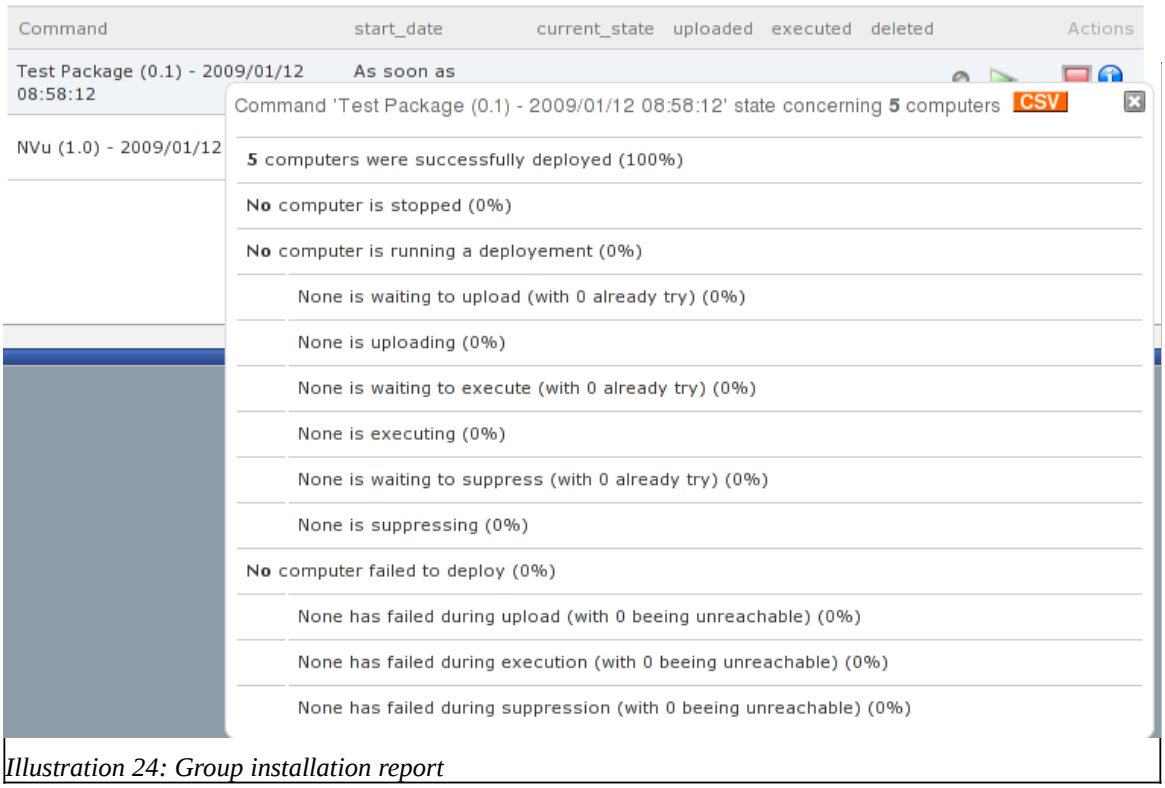


Illustration 24: Group installation report

Note that these information can all be exported into a CSV file for the purpose of writing reports or other external documents.

Each information line can also be exported into a CVS of the form:

```
Computer's UUID, FQDN, current state, start time, end time
```

The resulting file can, for example, let you create a group containing all computers in a failure state on a specific command using the feature that creates a static group by importing a file.

3. Quick Actions

A list of Quick Actions is available for a group or machine. This feature lets you execute simple commands (without sending a file) on a machine or group of machines. After confirmation, the action is carried out immediately; it cannot be programmed.

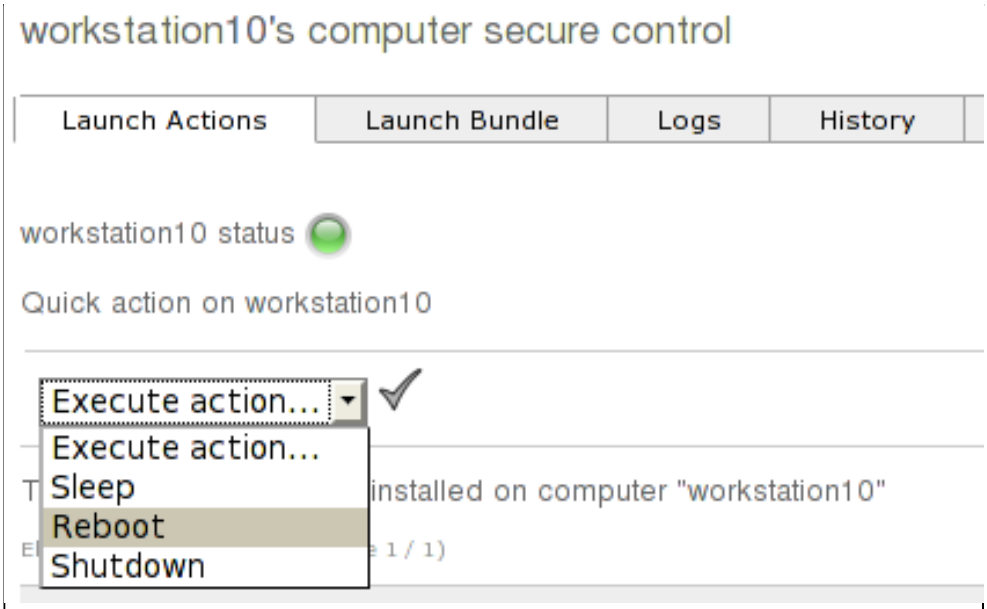


Illustration 25: Quick Actions

Imaging module

1. Overview

1.1. Features

This pulse2 module occupies an important place in a computer's life cycle because it lets you:

- create masters of an operating system;
- instantiate the operating system of a computer from a mater;
- do a first particularization of this operating system using a post-install step
- save the operating system of the computer during a PRA, for example.

1.2. Definition

A "rescue image" is a copy of an hard drive at a given date. Thanks to Pulse2, a rescue image can be created and then restored. We can use the word "image" as a short version of "rescue image".

A "master" is a copy of an hard drive at a given date specially prepared to be used as a base for the installation of the rest of the network.

2. Imaging server and entities

2.1. Imaging server

An imaging server is a server computer whose role is to provide save/restore features for the clients computers, and to stock these images.

This Pulse2 component has to be installed on all geographical places administrated by Pulse2 and requiring Imaging features. Each imaging server communicates with the central site where the Pulse2 administration console (web MMC interface) is installed.

Technically speaking, a imaging server has the following services:

- Pulse2 package service;
- Pulse2 imaging server;
- NFS File System;
- TFTP service;
- DHCP service, if there's no other one on the network.

2.2. Entities

An entity usually corresponds to an administrative division and/or geographical localization.

Each computer contained in the the Pulse2 inventory belongs to an entity, by default the root entity.

2.3. Link between entity and imaging server

The client computers of an entity communicate with the imaging server contained in their entity for the imaging functionalities. In fact, to have correct performances for the hard drive imaging , the client computers have to be on the same LAN than the imaging server.

Note well that it's not possible to have more than one imaging server by entity.

Case of a mono-site computer network

This is the simplest case. It is computer network contained in a single entity, the imaging server is simply a complete Pulse2 installation.

The imaging server is simply linked with the root entity.

Case of a multi-site computer network

For a multi-site install, we would have an imaging server on each secondary site, and one complete Pulse2 server on the main site controlling the secondary sites.

The administrators of each entities connect themselves on the MMC web interface of the Pulse2 server for the administration tasks.

In case of loss of access from the remote sites to the main server, save and restore features are still available, but it's not possible to add computers in the imaging module anymore. Moreover, some features of the Pulse2 administration console are not available anymore.

2.4. Role of the packages server on the imaging server

The packages server, in addition with its role of management of the packages available for deployment, maintains the list of available disk images on the imaging server, and the list of computer subscribed to the imaging server.

Technically speaking, the link between an imaging server to an entity corresponds to a linkage of the packages server with an entity.

Note: unlike packages, disk images cannot be synchronized between the various sites. This feature will be available in the future.

3. Computer's profiles

3.1. Profile notion

A profile defines type of install for a set of computers.

For example, the "server-lamp" profile can correspond to:

- an imaging menu letting you install a base master Mandriva Enterprise Server 5;
- an application bundle installing Apache, MySQL and Python

For example, the “windows-workstation” profile could correspond to:

- an imaging menu letting you install a base master Windows XP, with a post-install script letting you move Windows in a domain;
- the Firefox package;
- the Microsoft Office package.

A profile lets you restrict the possible actions on the computers on which it is assigned.

A profile can also be seen as a object selection provided by the packages server.

3.2. Profiles and packages servers

The objects contained in a profile (packages, imaging, …) are provided by the packages server. Thereby, a profile is always linked to one (and only one) packages server.

Only the computers that belongs to the entities on which the packages server applies can be added to the profile.

3.3. Profile constituents

Elements of the startup menu of the imaging module can be associated with a profil.

In the next Pulse2 versions, it is expected to be able to associate these elements with a profil:

- one or more packages;
- one list of quick actions;
- one or more bundle(s).

3.4. Pulse2 profiles management in the MMC

The profiles management is similar to the group management. It is available in the “Computers” tab in the interface.

Profile creation

Click on the “Add a profile” tab in the left menu. The profile must first be linked with an imaging server.

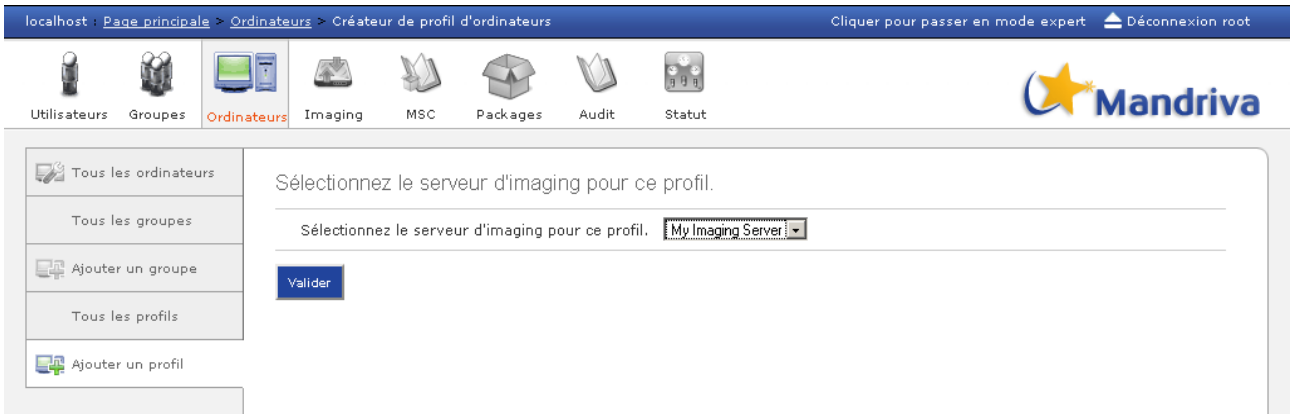


Illustration 26: Profile creation: linkage to an imaging server

Then, its creation is similar to a group creation (see 4).

Notes:

- a computer cannot be contained in more than one profile at a time;
- unlike a group, a profile cannot be dynamic.

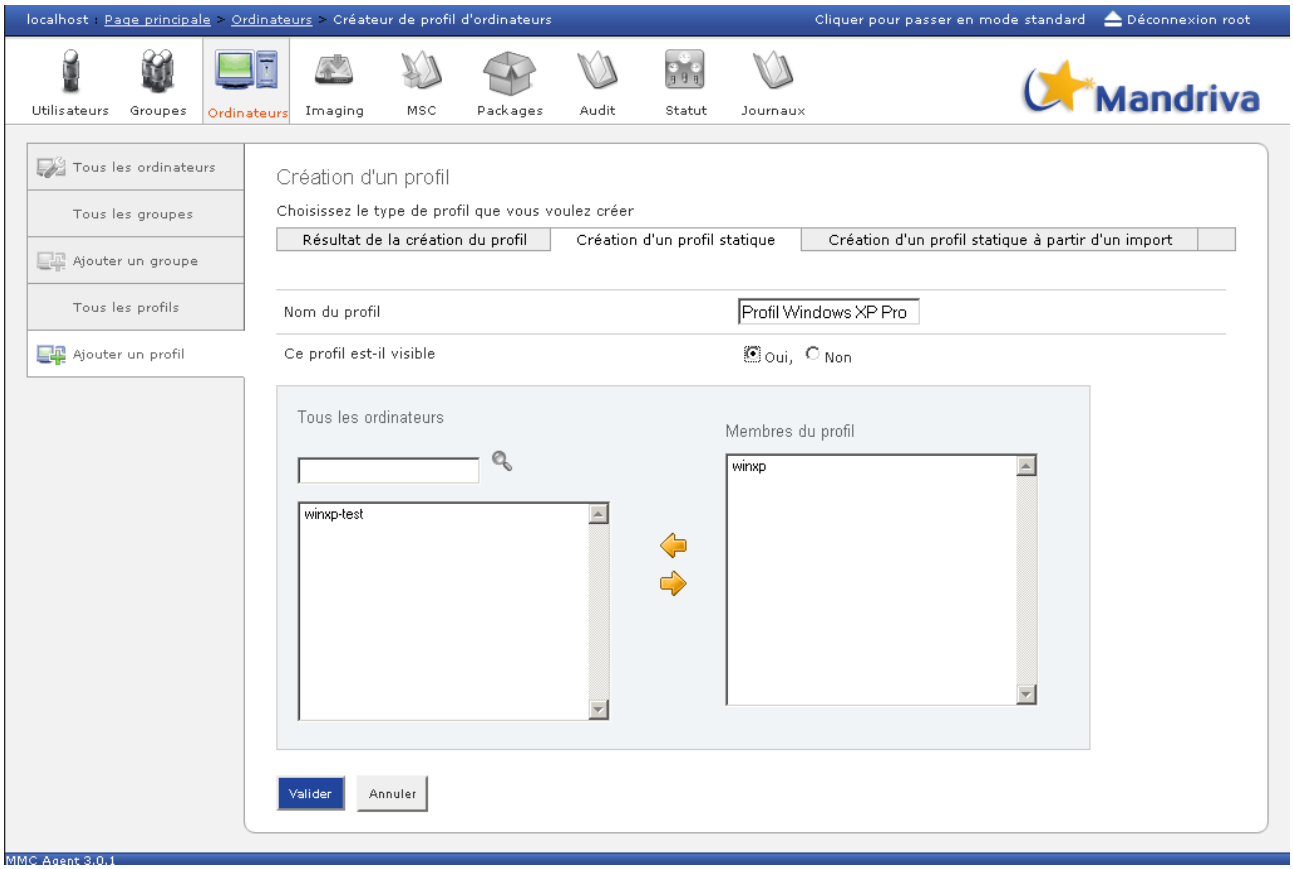


Illustration 27: Creation of a static profile

Profile deletion

To delete a profile, you only need to display the list of profiles and to click on the red cross in front of the profile to delete.

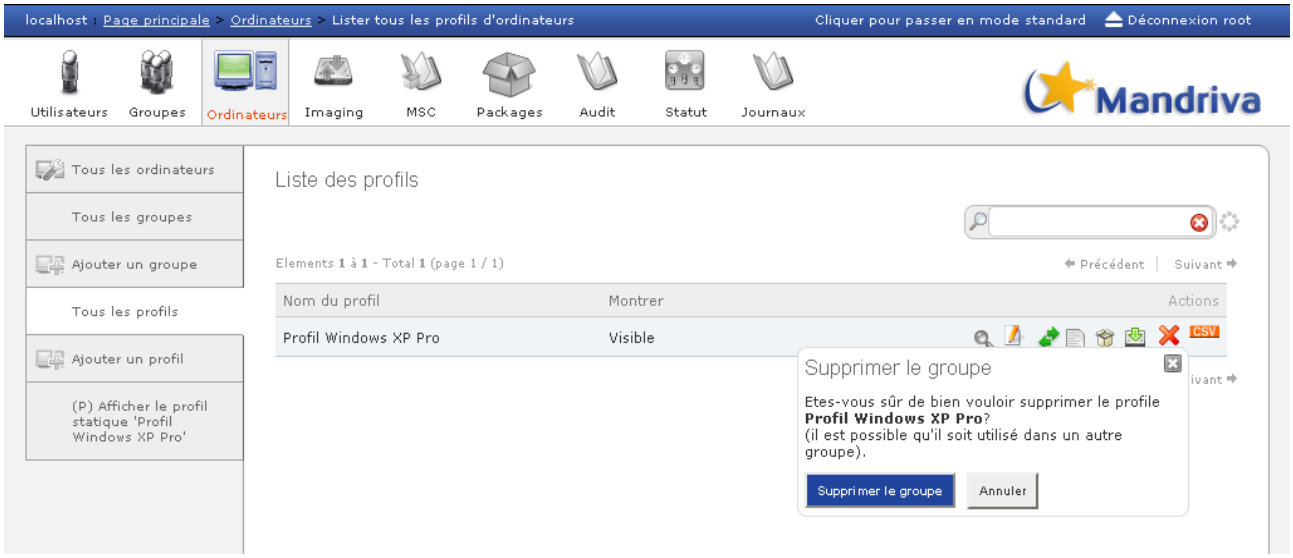


Illustration 28: Profile deletion

If the profile used to define an imaging menu, then the computers belonging to this profile receive a copy of of the profile menu.

4. Imaging servers management

The “Imaging” tab lets you link the imaging servers to their entity and then to configure them.

The list at the top of the page lets you choose the imaging server with which you want to interact.

Every time a change is done, the synchronization button appears at the top of the screen. The synchronization sends a request to the imaging server to make it do actually do the requested changes.

4.1. Imaging server state

The “Imaging server state” displays the state, some statistics and the logs of the imaging server.

4.2. Default startup menu of an imaging server

When a computer is subscribed in the imaging module, a copy of the default startup menu of the imaging server is attributed to it.

It is possible to modify this default menu using the “Default startup menu” tab:

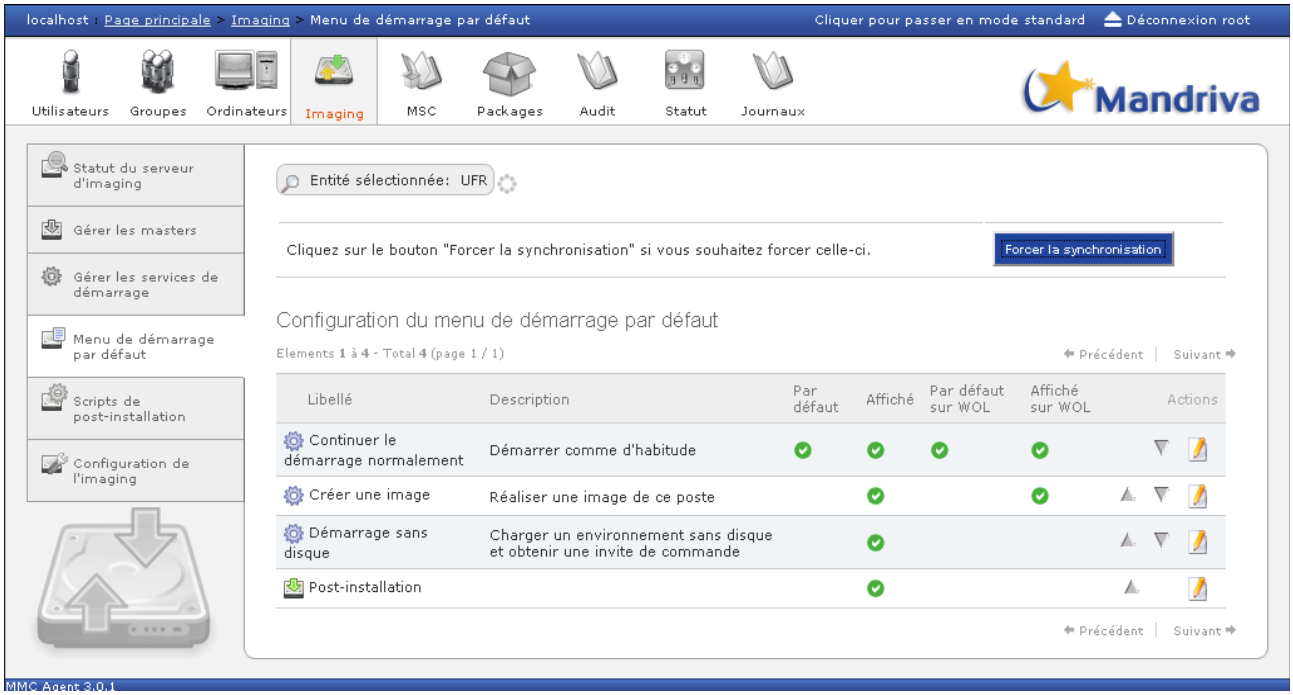


Illustration 29: Default startup menu

- Label: designate the name of the proposed actions. Note that you can change the text of the label by clicking on it.
- Description: Contains a short description for each action.
- Default: We choose the action that will be executed if no selection is made by the user. This column must have one ticked cell.
- Displayed: Lets you choose the choices that will be displayed in the user menu. There must at least one ticked cell.
- Default on WOL: The default choice after a wake of the computer by Wake On LAN.
- Displayed on WOL: Indicates if this entry is displayed after a Wake On LAN.

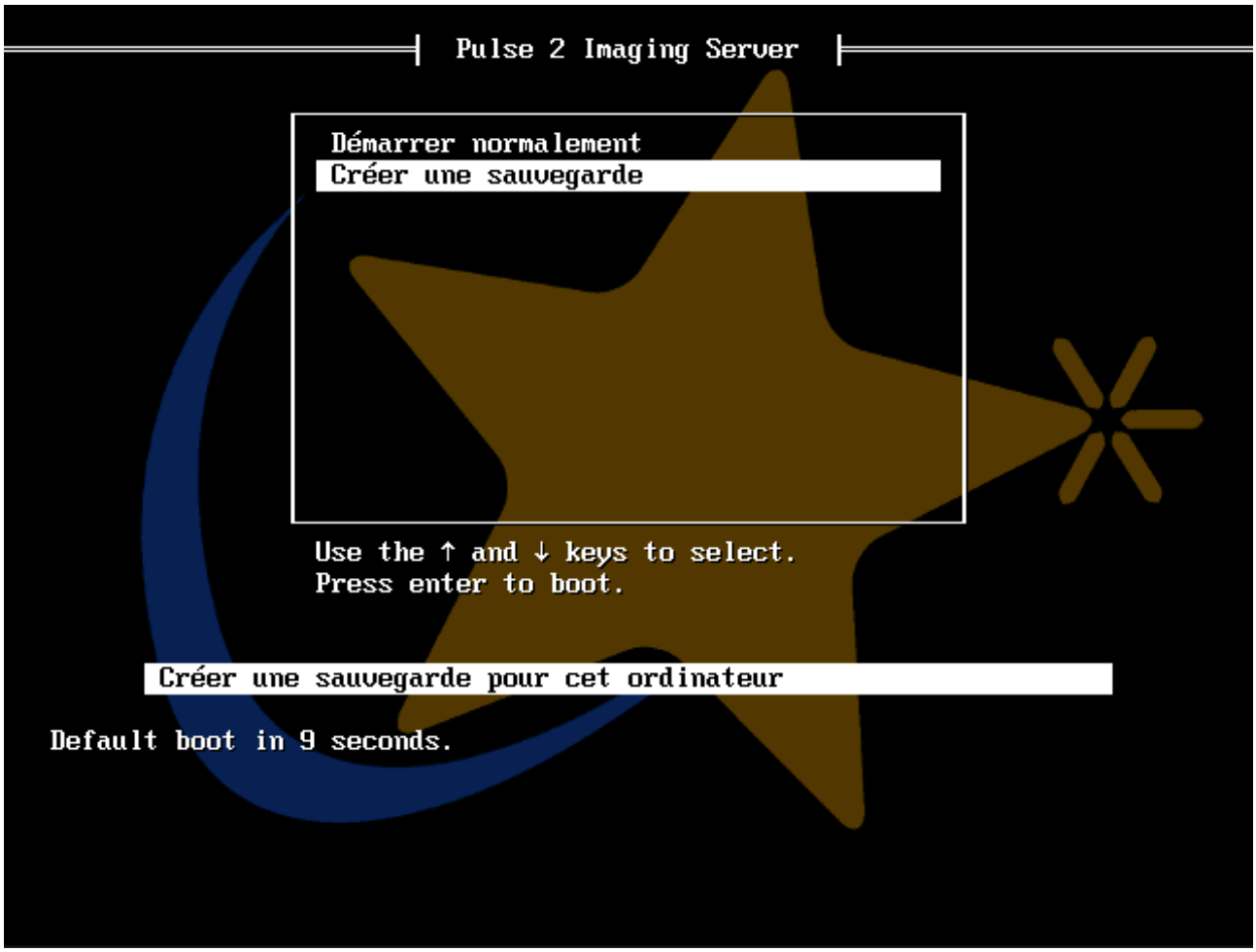


Illustration 30: Example of a basic network startup menu of a Pulse2 client

Once the modifications are done, do not forget to click on the “Synchronize” button so that the changes are applied. When you click on that button, you define the default startup menu that will be assigned to the new client computers when they are subscribed to the imaging module.

The “Imaging configuration” lets you modify some parameters of the default startup menu for the imaging server.

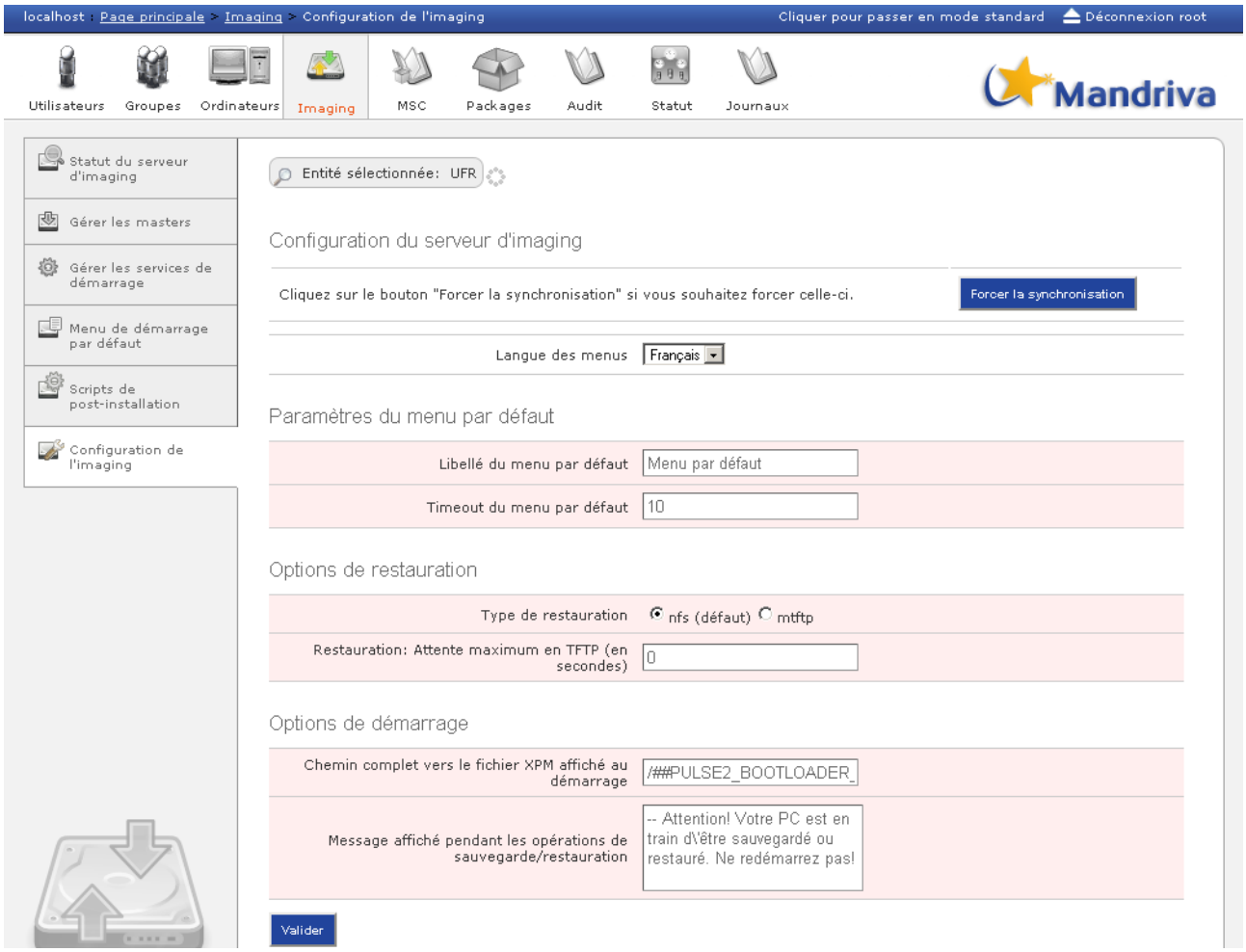


Illustration 31: Imaging server configuration in expert mode

The only available parameter in standard mode is the selection of the startup menu language. This parameter is applied to all menus that will be displayed by the client computers handled by the selected imaging server. This lets you change the language depending on the site it is installed on, for example in a case of a network spread on multiple countries.

The other parameters are available in expert mode. When a new computer is registered in the imaging module, its menu is instantiated by default with these parameters:

- Computer's menu's parameters:
  - the menu label
  - the menu timeout (in seconds)
- Restoration options
  - restoration type (NFS (by default) or MTFTP for multicast)
  - the restoration timeout for the MTFTP mode
- Target options
  - kernel parameters
  - parameters required for the backup or the restoration

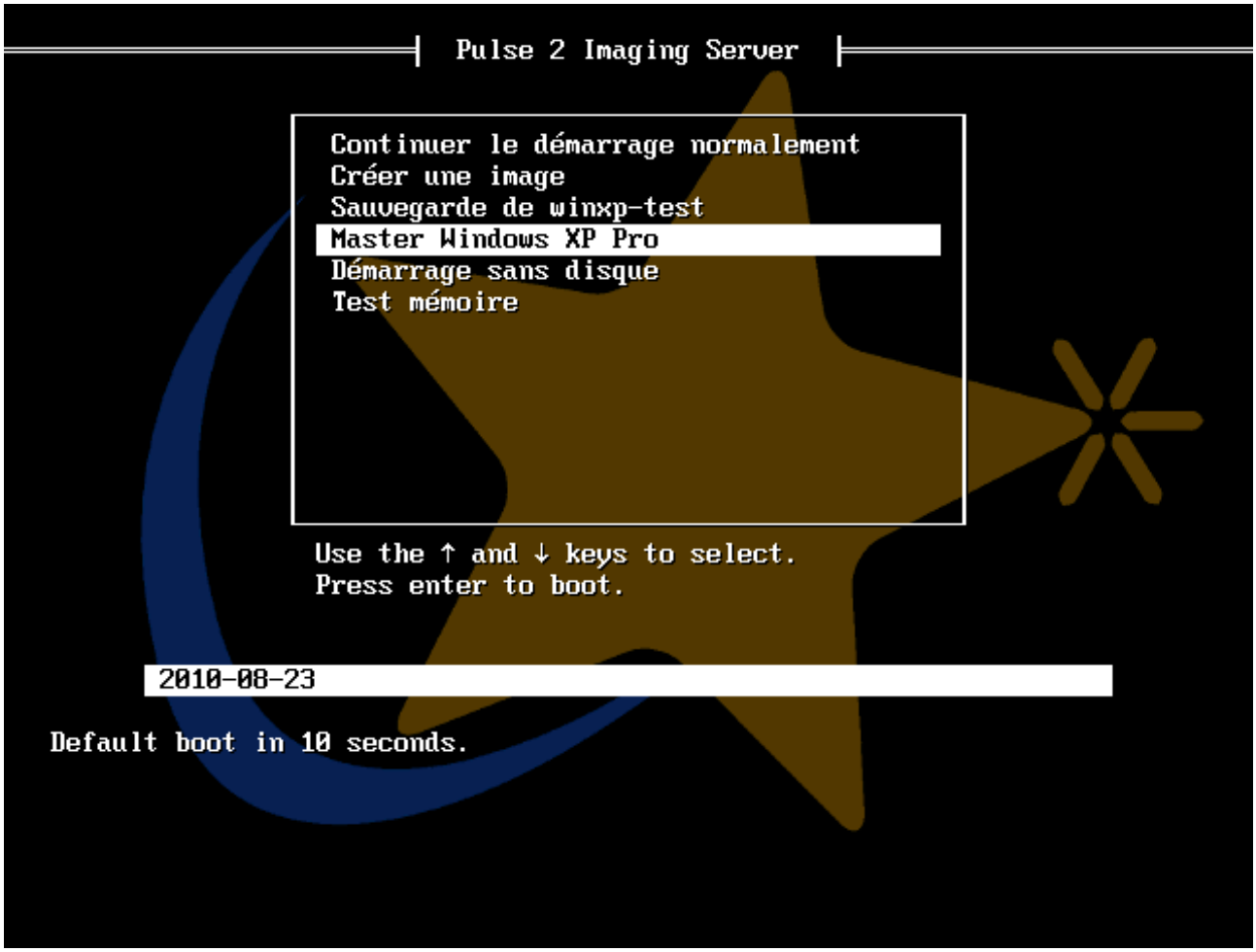


Illustration 32: Example of a advanced startup menu of a Pulse2 network client

4.3. Global masters management

This page displays the list of available masters on the imaging server local to the entity of the computer.

When a master is created, it is available for all the computers of the imaging server on which it is saved.

For each master, it is possible to:

- add it to the default startup menu of the new computers;
- prepare a auto-restore ISO;
- edit it: label and description, post-install options
- delete it, if it's not used.

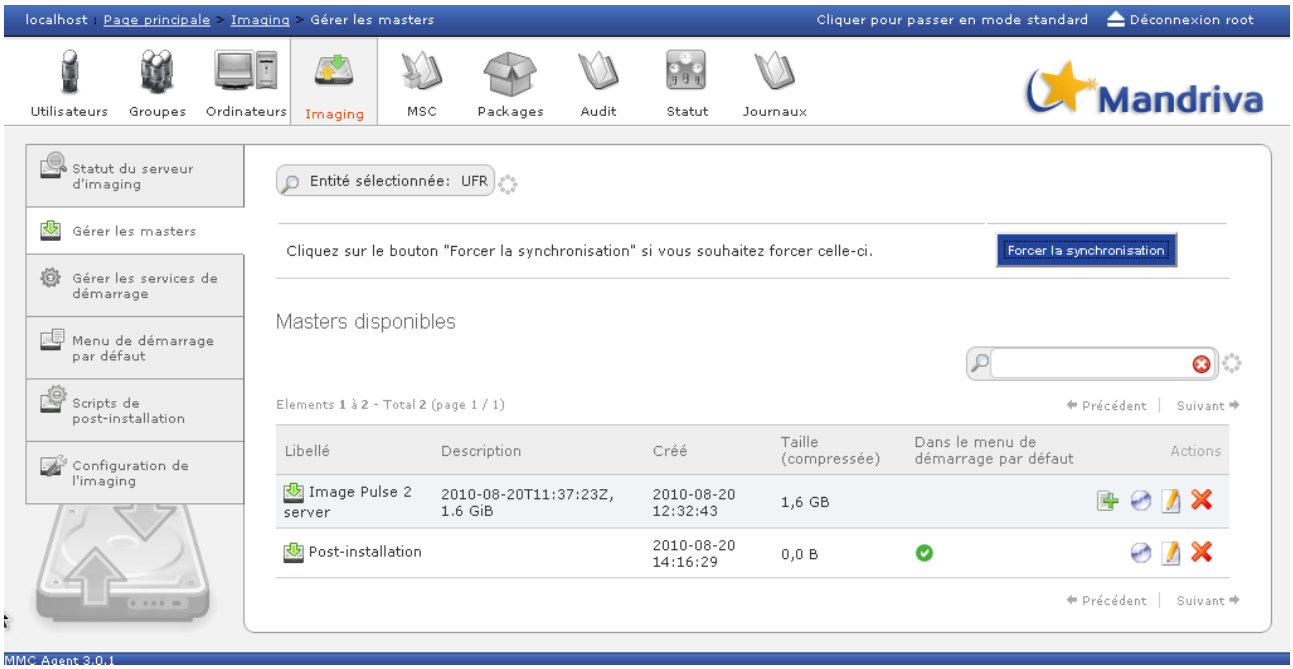


Illustration 33: Masters management

4.4. Post-install

This tab lets you handle the available post-install scripts

Base post-install scripts

The provided scripts offer functionalities covering the most common deployment scenario:

- Date: write the current date in the “date.txt” file that will be placed at the root of the first partition of the first disk. Used for a simple test.
- Sysrep: make a copy of the sysrep.inf file on the first partition of the first disk and change the NetBios name present in the file by the one declared in Pulse2.
- SID: re-initialize the SID and the NetBios name of a Windows system.
- Shutdown: shutdown the system after the restoration.
- Debug: debug shell
- Partition resize: extend the only FAT or ext2 partition on the whole disk.
- NTFS resize: extend the only NTFS partition on the whole disk.
- Agent pack: install the Pulse2 agent pack for Windows
- ICH 5 sync: synchronization of the 1st and 2nd disk on the fake RAID cards based on the ICH5 chipset.

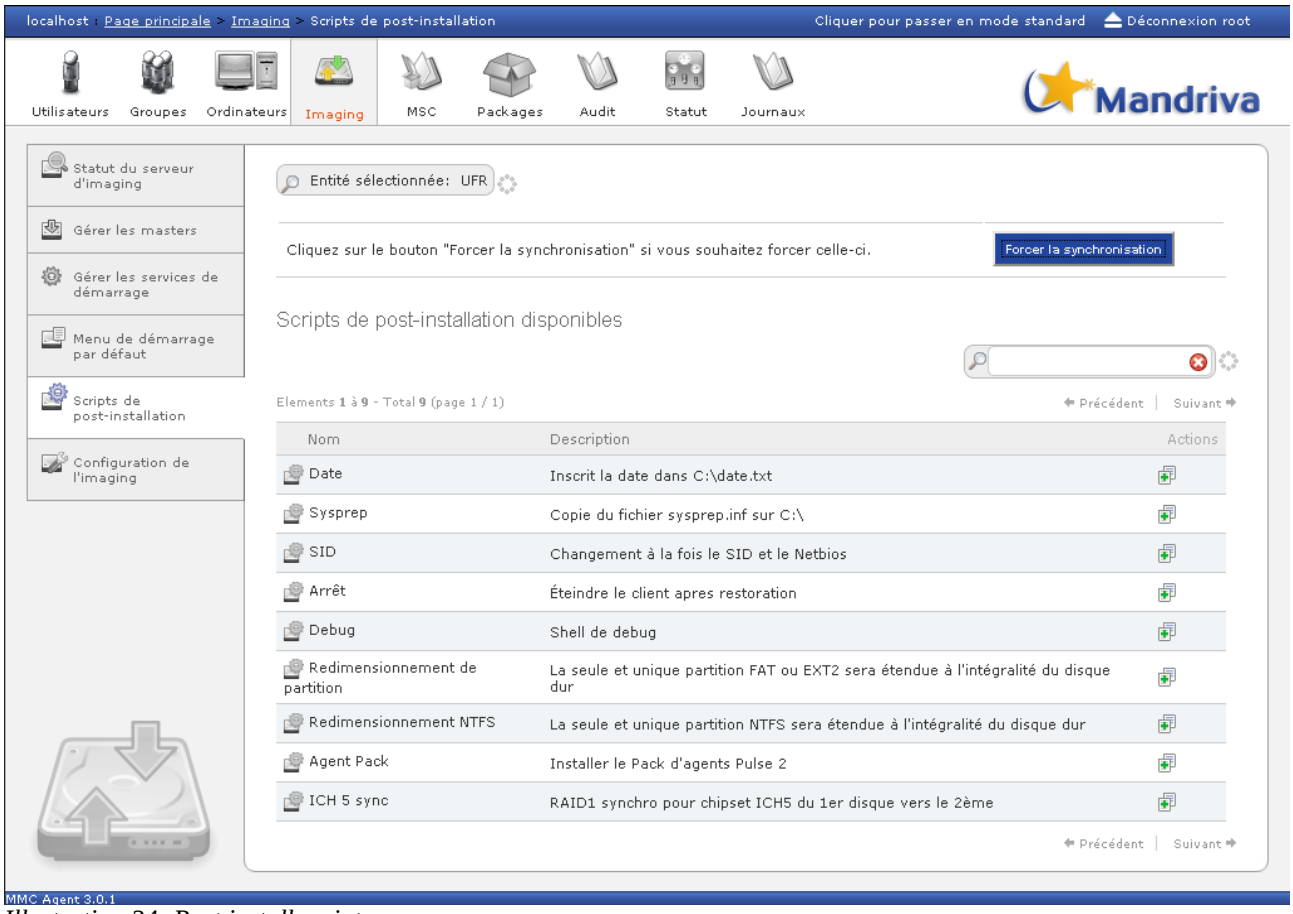


Illustration 34: Post install scripts

Post-install scripts creation

The default scripts are only basic examples. To explore the infinite possibilities of the post-install system, all scripts can be duplicated, this means that a new script can be defined based on an already existing script.

5. Imaging client use

5.1. DHCP server configuration

To be able to start over the network, the client computer has to know where the data needed for the startup are located. This configuration consists only in the addition of DHCP options on the on the addresses range of the computers concerned by Pulse2.

These options are:

- "filename" equivalent to */bootloader/pxe_boot*. This option is often available under the reference "Option DHCP 66, Boot Server Host Name".
- "next-server" equivalent to the IP address of the Pulse2 server. This option is often available under the reference "Option DHCP 66, Boot Server Host Name".

```
# changer 192.168.X.XXX par l'IP du serveur
next-server 192.168.X.XXX;
filename "/bootloader/pxe_boot";
```

5.2. Configuration of the client computer

PXE lets a computer boot over the network.

PXE is an open industrial standard developed by some software and hardware vendors. PXE enables a network card to become a boot medium. This implies the addition of the network card in the usual list of boot medias, like hard drives, CD/DVD ROMs or USB drives that load the operating system or the installation software on the computerr.

Network cards that do not support PXE are rare. However, it is often necessary to edit the BIOS parameters, for example the “boot network” (enable), and eventually the virus check option of the BIOS (which detects the PXE ROM as a virus).

5.3. Client’s subscription

There are several ways to add a client in the Pulse2 Imaging module.

List of computers of the MMC web interface

The “Computers” tab of the MMC web interface displays the list of computers subscribed in the inventory base, whatever the backend used (native of GLPI). If the computer’s inventory contains only one MAC address, it can be subscribed to the imaging module by clicking on the action icone available in the computers list.

Note that in the case of of the native backend, it is possible to manually add a computer from the “Add a computer” page of the ”Computers” tab, without using the inventory.

Network boot menu

In the startup menu of the client computer, chose the menu entry named “Add as a Pulse2 client”, and fill the computer’s name.

Several cases are then possible:

- the inventory base contains a computer having the client’s MAC address. The subscription will be accepted only if the computer’s name is identical to the one in the inventory base;
- the inventory database does not contain any computer with this MAC address. If the Pulse2’s native backend is being used, the client computer is first saved in the inventory database then subscribed to the imaging module. If the backend being used in GLPI, the subscription

is refused.

Note: this means that if Pulse2 is associated with the GLPI inventory database, the computers gave to be created in GLPI first before they can be subscribed in the imaging module.

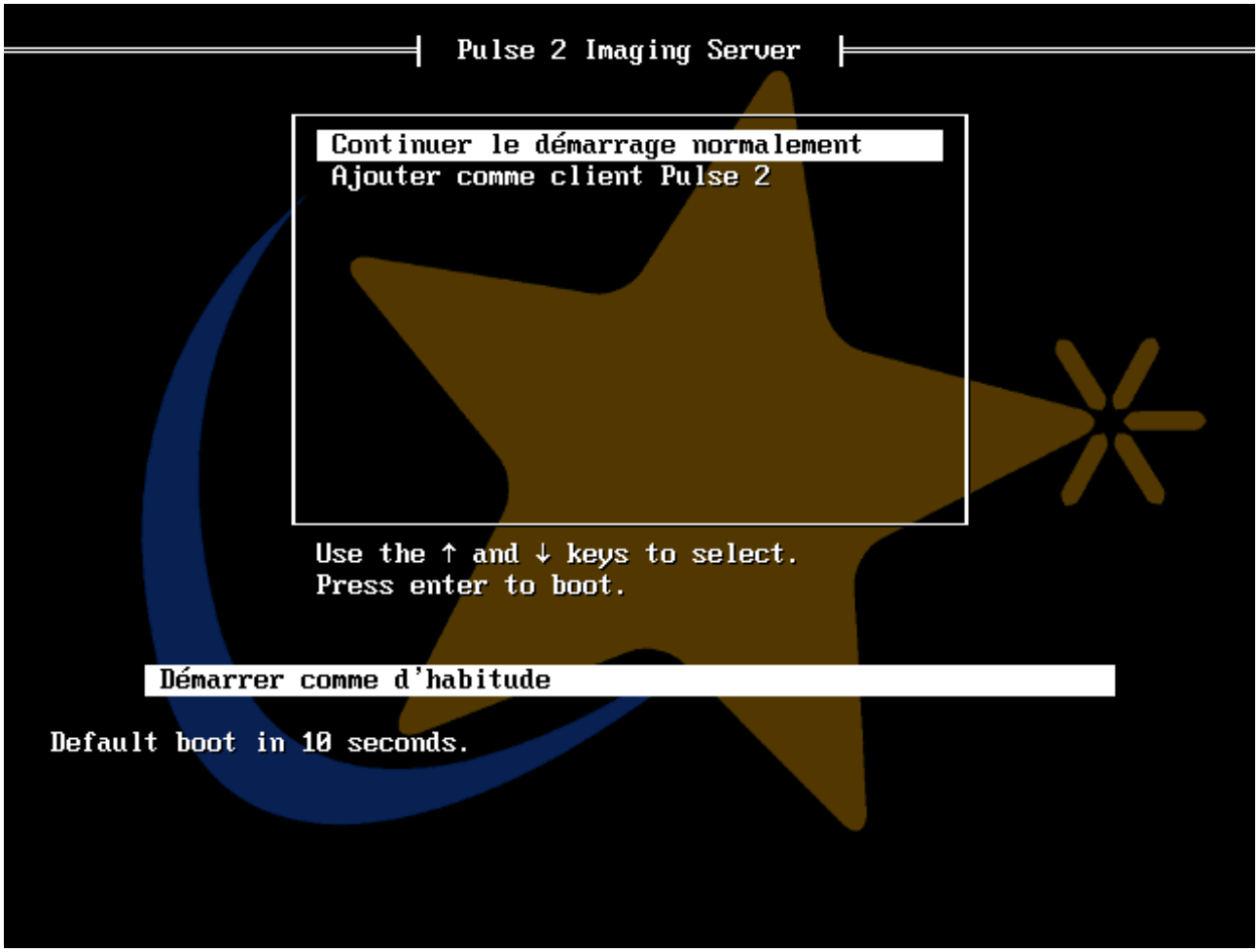


Illustration 35: Expmle of a network boot menu: subscription of a Pulse2 client

5.4. Subscription of a computer in a profile via the network boot menu

Using the network boot (cf. Network boot menu), it is possible to subscribe a client computer and to add it in a profile at the same time.

To do that, the save line syntax must be composed of a profile name and the computer's name in the following way: profile_name:computer_name

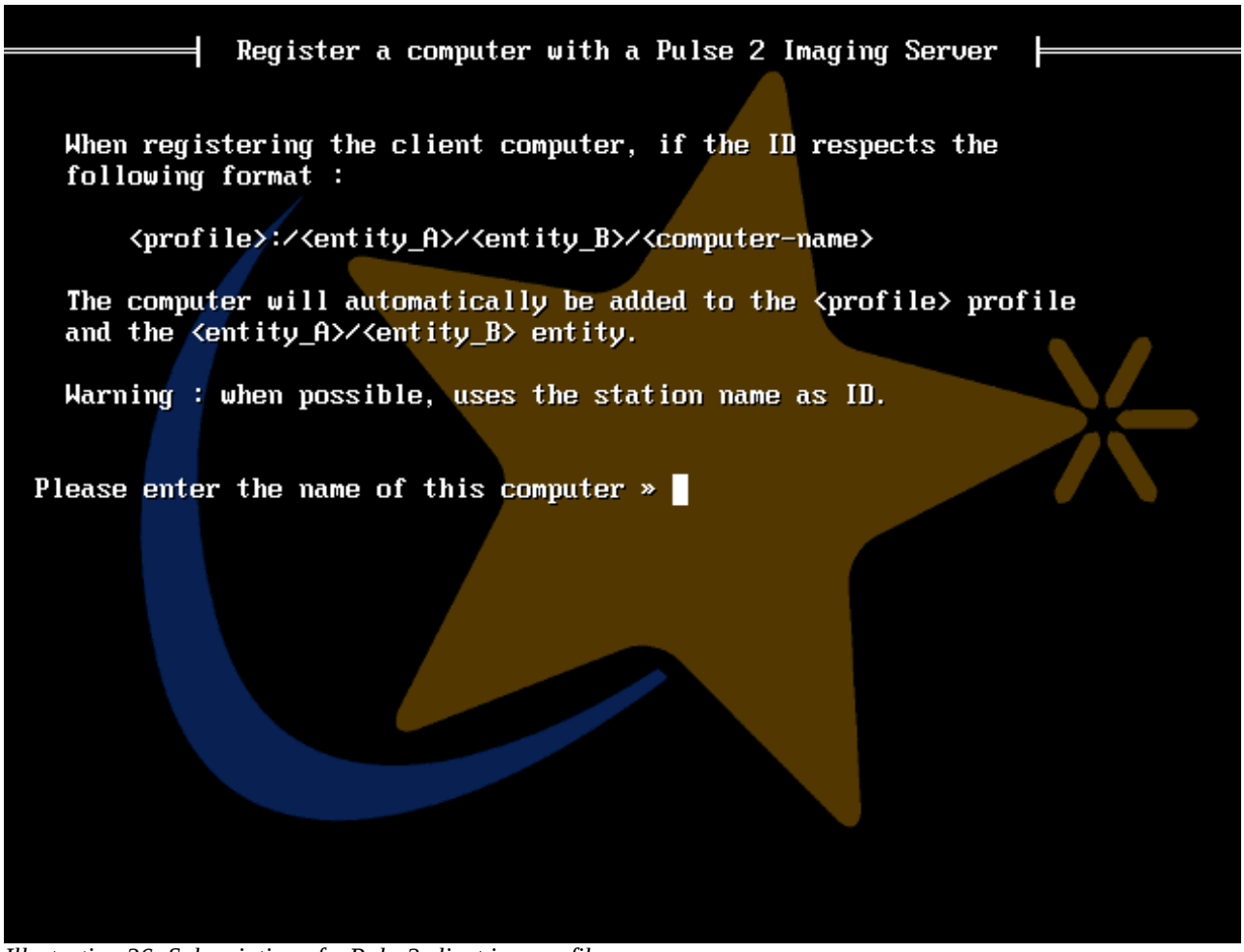


Illustration 36: Subscription of a Pulse2 client in a profile

5.5. Boot menu of a computer in a profile

When a computer is part of a profile, it is impossible to change the content of the boot menu or of the boot menu options for that computer: it is in read only. However, the user can add the rescue images of the computer to the menu if some are available.

For that part of the personalized network boot it is possible to:

- sort the element in the menu using the arrows. Note that the profile elements will always be displayed first in the consolidated boot menu;
- remove an element of the menu;
- edit a menu element, but only to change its states “Displayed” and “Displayed on WOL”.

Finally, advanced and specific parameters can as well be set in advanced mode using the “Menu configuration” tab of the computer:

- the Linux kernel boot parameters and its “initrd”
- disks and partitions to exclude from the saves and restorations

6. Management of a client computer

Once a computer has been subscribed to the imaging module, its menu and its imaging parameters can accurately be configured:

- the “boot menu” tab lets you modify the entries order in the menu and to select the entries that will be displayed and launched by default;
- the ”Images and masters” tab lets you manage the list of rescue images of the computer and to add rescue images or masters to the boot menu;
- the “boot services” tab lets you add service entries in the boot menu. An entry to start a memory test can be added, for example;
- the “imaging logs” tab lets you consult the save and restoration logs of the computer;
- the tab “menu configuration”.

6.1. Boot menu

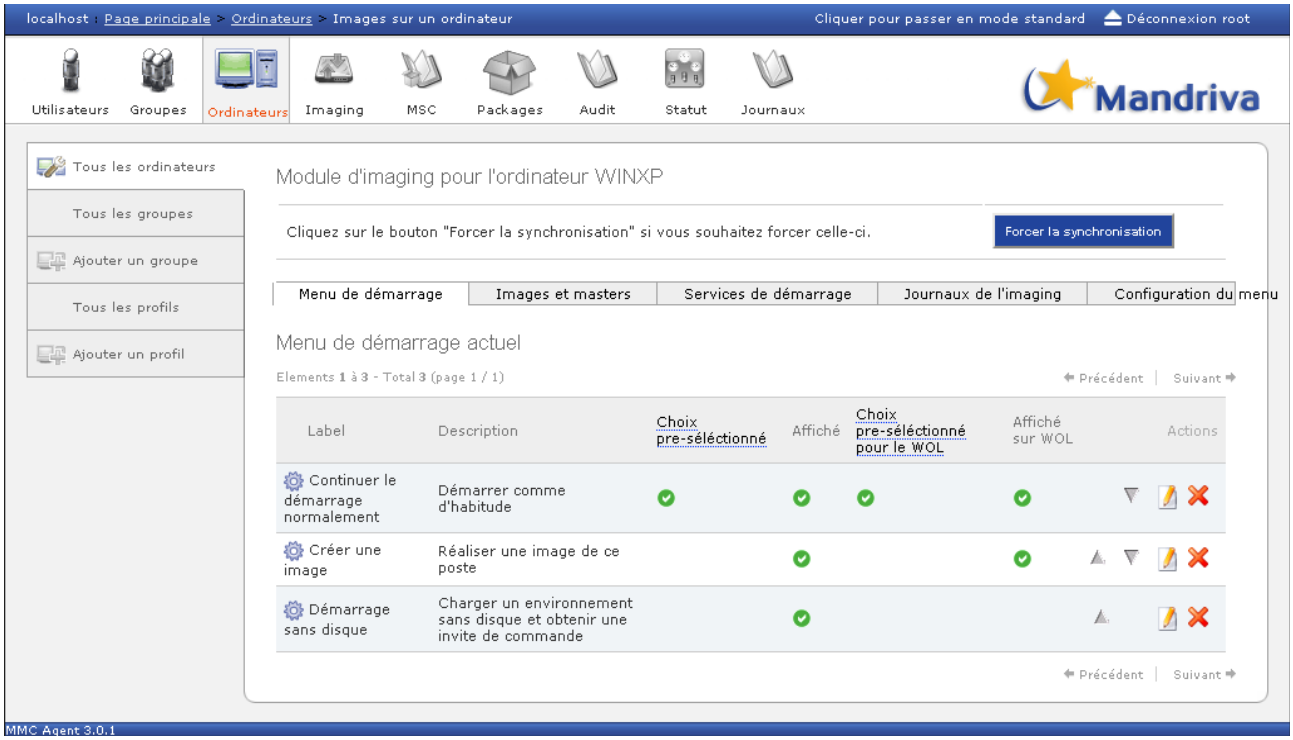


Illustration 37: Boot menu of a client computer

The parameters of the boot menu of the computer are more precise from the computer’s configuration:

- possibility to hide the menu;
- possibility to edit the GRUB command line
- possibility to deactivate the disk checks (to force the restoration of an image even if

the disk’s size is too small)

- possibility to update the NT boot loader if the disc changed
- possibility to use the Linux command line

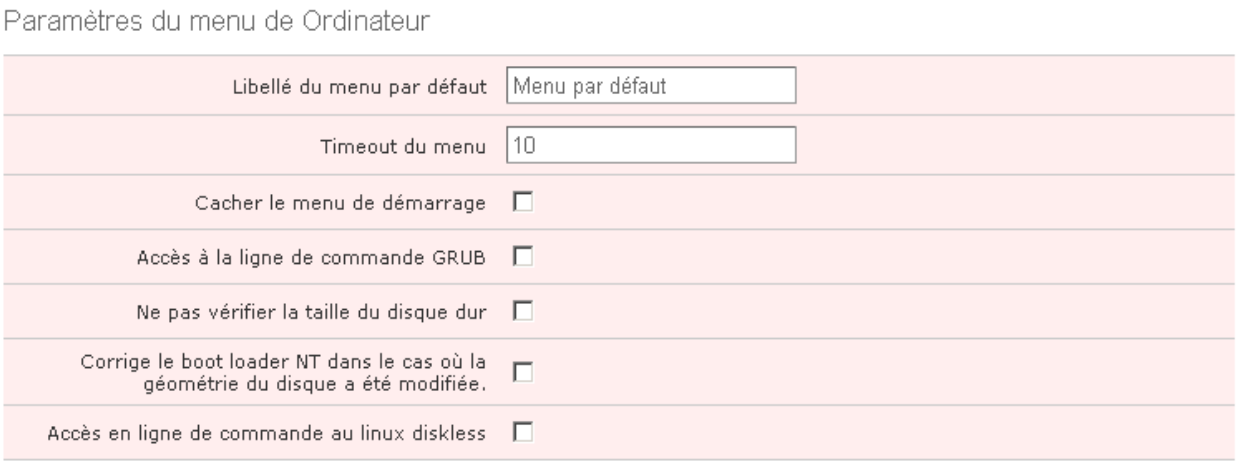


Illustration 38: Computer’s menu parameters

6.2. Images and masters for a computer

This page displays the list of rescue images of the computer and the available masters on the local imaging server.

For each image it is possible to:

- edit it: label and description;
- delete it

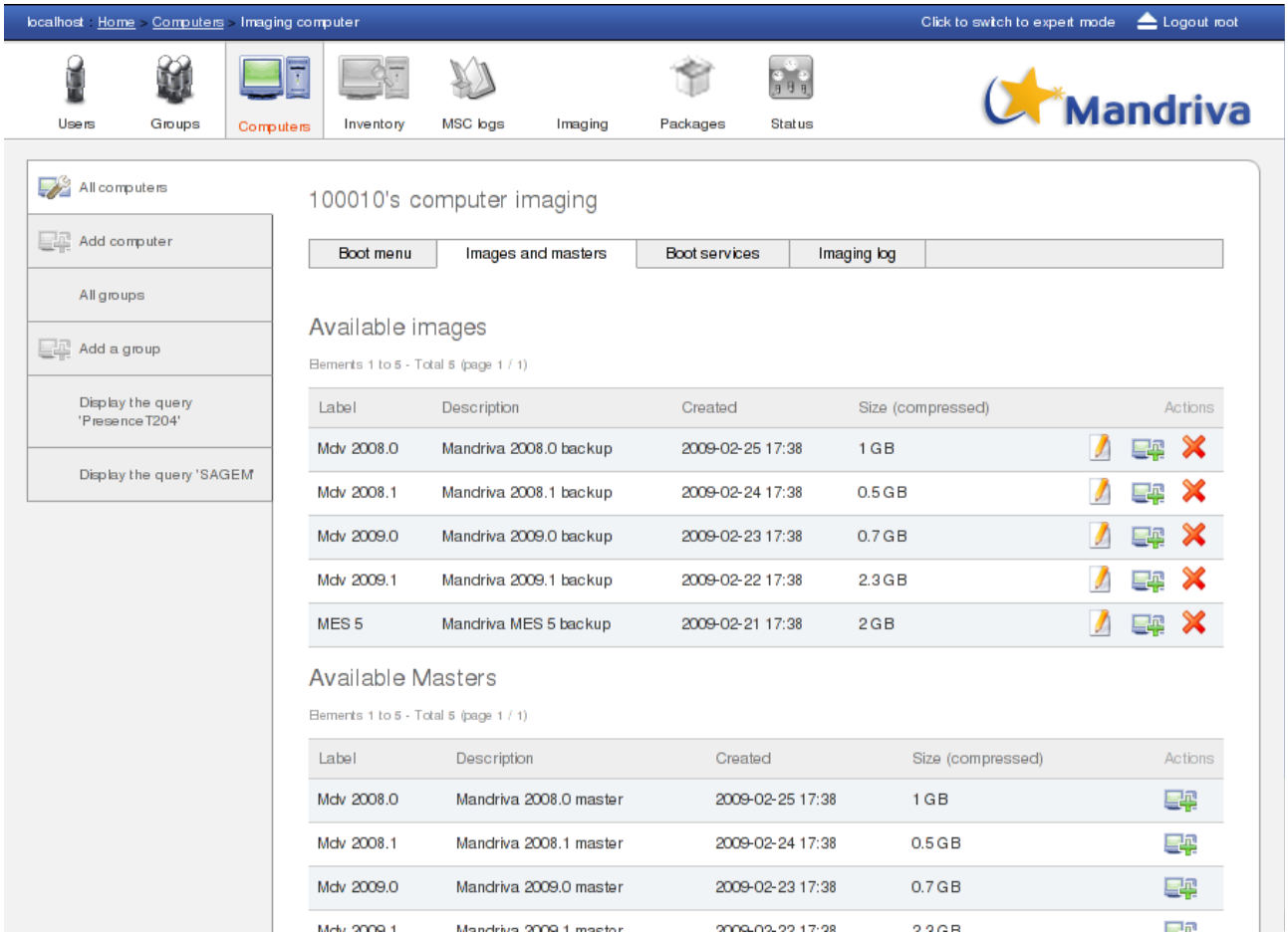


Illustration 39: Images and masters

On the masters, it is possible to ask for the launch of one or more post-install scripts at the end of the restoration.

6.3. Imaging logs

The imaging logs lets you visualize the last events associated with the imaging module for this computer.

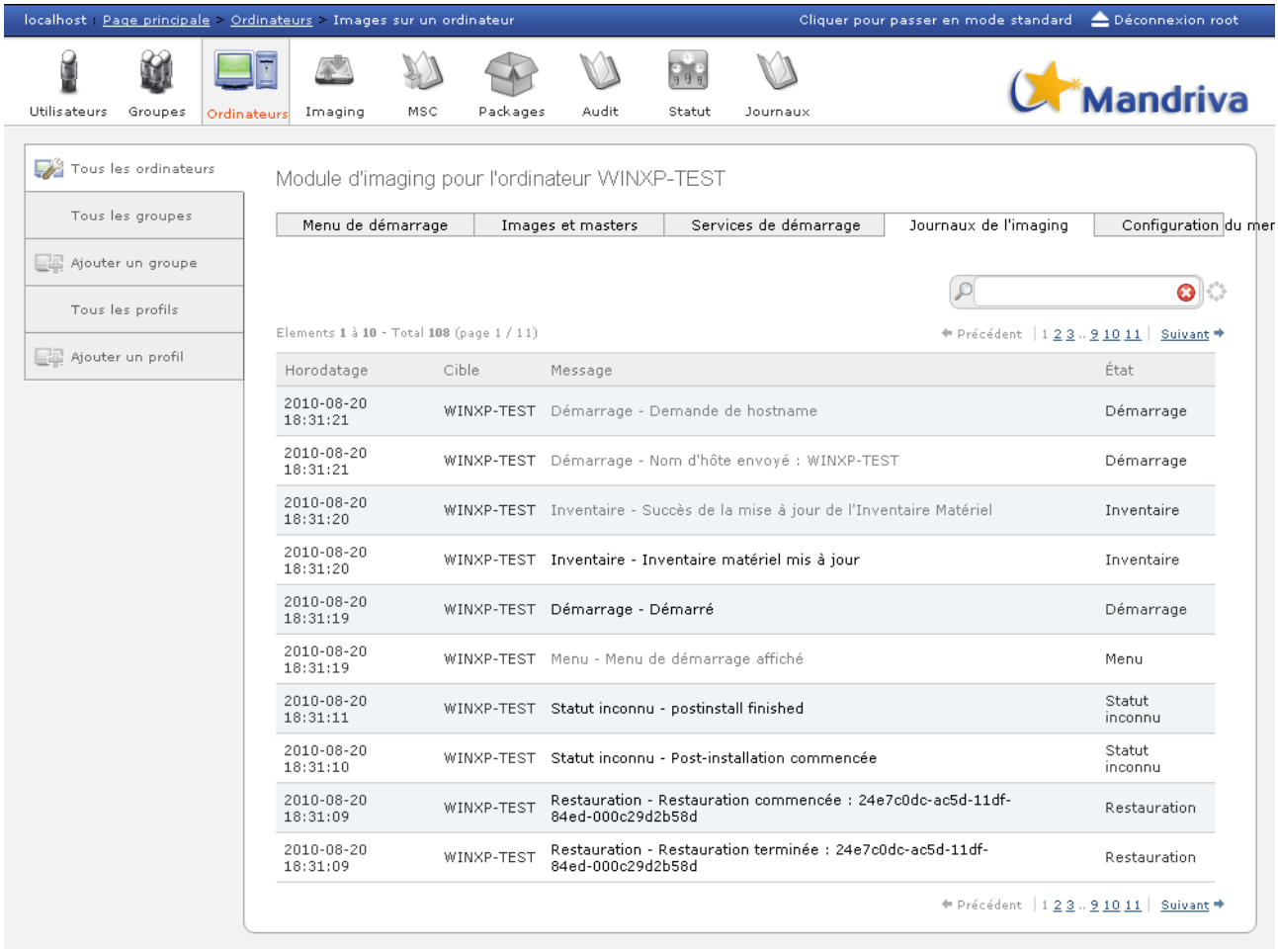


Illustration 40: Imaging logs

Inventory Management

The inventory of all computers in the system can be accessed via the Inventory module.

You can also view the inventory for a group of machines or a single machine using 

The inventory module lets you view the hardware and software characteristics of the computers.

List of categories:

- General: model and Serial Numbers of machines
- Hardware: processor, OS, RAM
- Software: company / name / version for each application installed
- Network: TCP/IP parameters and MAC address
- Controller: computer chipsets
- Registry (optional): Windows registry keys

Each inventory can be exported to a CSV file using the following button: 

Remote Diagnostics

The remote diagnostic function allows a user to retrieve a file from a particular folder in the machines. This folder is the same for all machines. The first file contained in the folder will be downloaded. It is generally a log file or a debug file, and can be generated by a Quick Action, for example.

Illustration 41: Remote diagnostics page

Single-click the remote diagnostics icon to go to the remote diagnostics page, which offers the following options:

- Launch retrieval of the remote diagnosis to the Pulse2 retrieval zone
- A list of available retrieved remote diagnoses. The possible actions in this list are to delete a remote diagnosis from the Pulse2 retrieval zone and to download a remote diagnosis that has already been retrieved onto the viewing console
- Indicators for the current retrievals (with the option to cancel them).

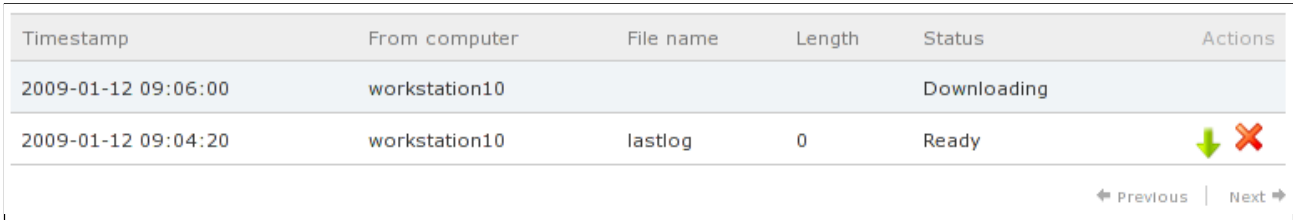


Illustration 42: Downloading of a file onto the target machine is in progress.

Remote Viewing of Computers

This feature lets you view the screen of a target machine remotely using VNC protocol. Note that the VNC connection is secure between Pulse 2 and the target machine because it is encapsulated in an SSH connection.

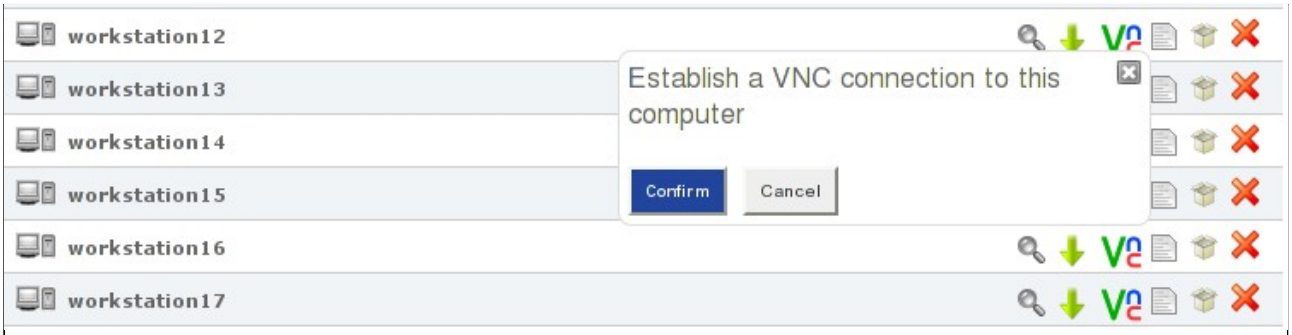


Illustration 43: The remote control icon.

Click 'Confirm' to open a connection with the target machine.

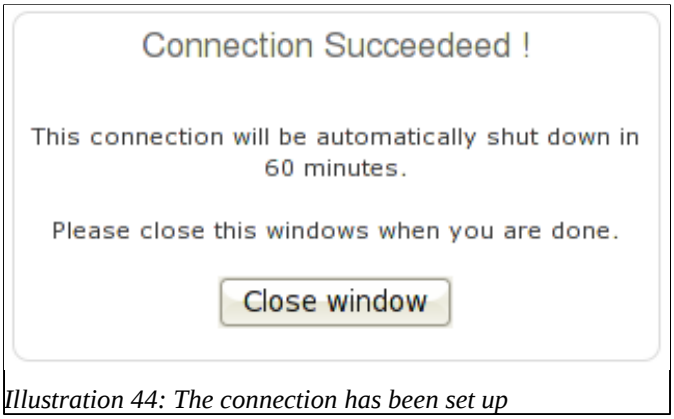


Illustration 44: The connection has been set up

A Java applet appears, inviting you to view the screen of the target machine remotely. If the following screen appears, click 'Execute'.

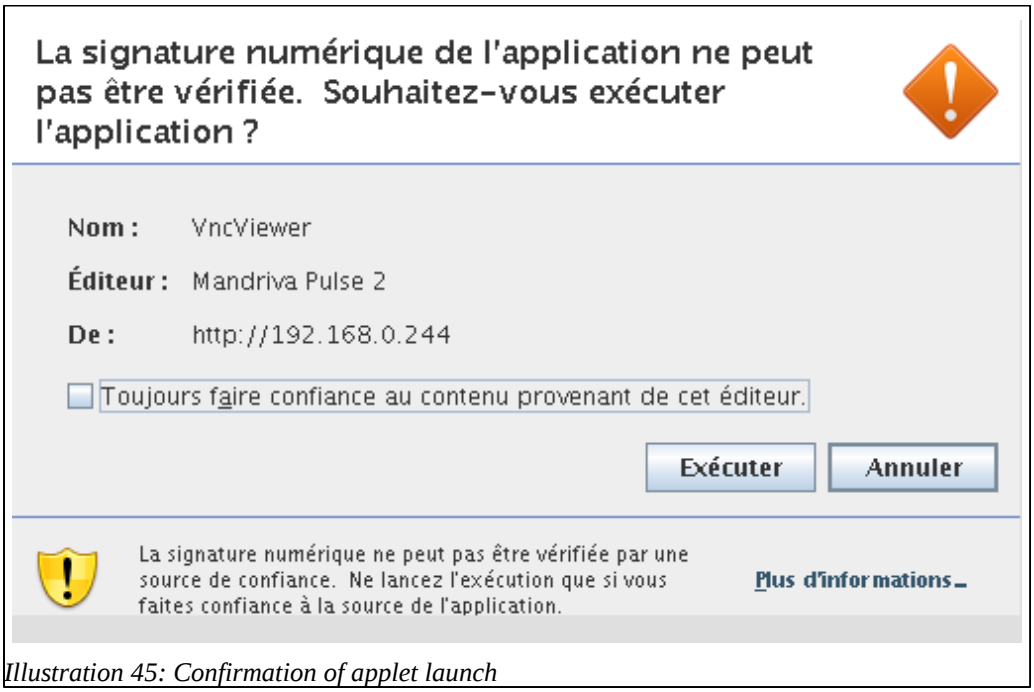
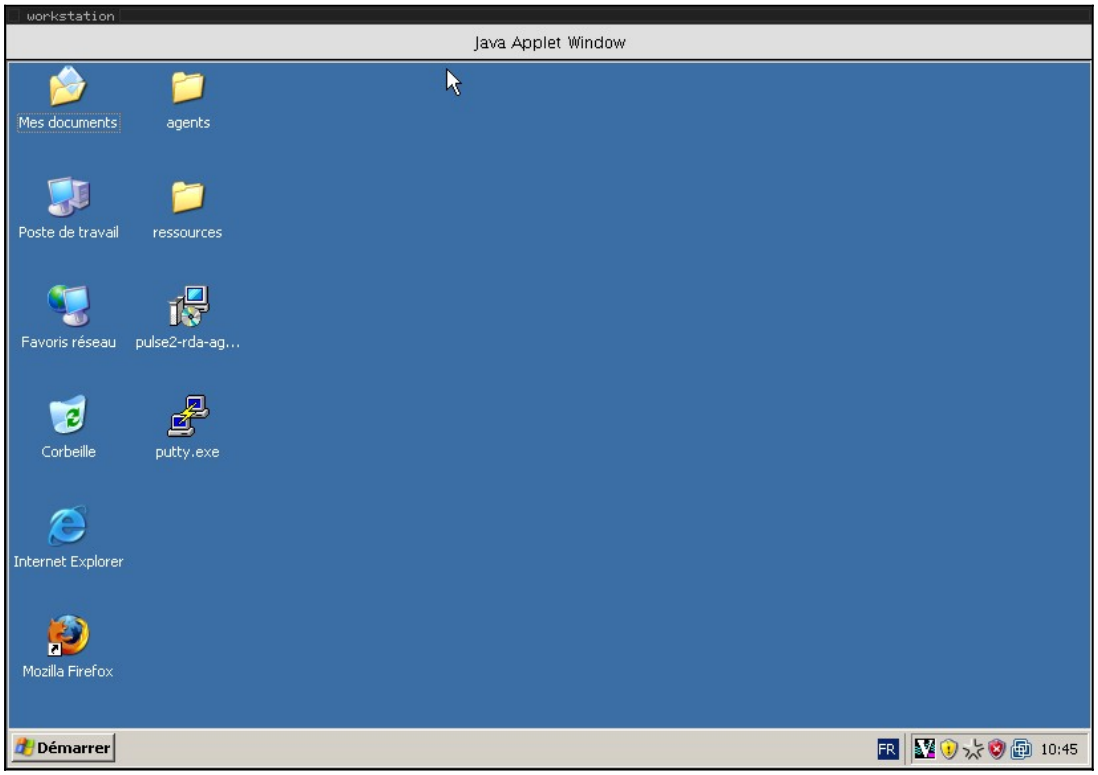


Illustration 45: Confirmation of applet launch

The user then has access to the desktop of the target machine.



Index of illustrations