
Orthos2

openSUSE Orthos2 Team

Apr 29, 2024

CONTENTS:

| | | |
|----------|------------------------------|-----------|
| 1 | Administrator`s Guide | 1 |
| 2 | User Guide | 21 |
| 3 | Orthos Client | 29 |
| 4 | Indices and tables | 35 |

ADMINISTRATOR'S GUIDE

These pages are partly very technical. You should read the User Guide before to understand the basic concept and the goals of Orthos. This document describes how to create and administrate groups and users, architectures, domains, machines, machine groups, serial console types, systems and enclosures, vendors and platforms. Furthermore, the administration of the Orthos server will be discussed.

1.1 Installation/Setup (Production system)

1. **Install the orthos2 package:**

```
$ zypper install orthos2
```

2. **Change nginx user to orthos:**

In `/etc/nginx/nginx.conf` add user `orthos orthos`; to the beginning of the file

3. **Enter the correct server name to the nginx server conf:**

In `/etc/nginx/conf.d/orthos2_nginx.conf` replace `127.0.0.1` in the `server_name` directive with the fqdn of the orthos2 server

4. Deploy code to database

```
orthos-admin migrate
```

5. Install fixtures (push data to database):

- a. In case of a fresh install - Load data from package data:

```
orthos-admin loaddata /usr/share/orthos2/fixtures/data/*.json /usr/share/  
orthos2/fixtures/taskmanager/*.json
```

- b. In case you want to load data from another orthos instance where you previously dumped data tables:

```
orthos-admin loaddata "dumped_table.json"
```

6. **Create a superuser**

```
orthos-admin createsuperuser
```

7. **Create html files from templates**

```
orthos-admin collectstatic
```

8. **start all services**

```
systemctl enable nginx
systemctl enable orthos2.socket
systemctl enable orthos2.service
systemctl enable orthos2_taskmanager

systemctl start nginx
systemctl start orthos2.socket
systemctl start orthos2.service
systemctl start orthos2_taskmanager
```

1.2 Server Configuration

Here the Orthos settings are defined and the Orthos server behavior is adjusted. The setting options are usually self-explanatory by their names.

1.2.1 `domain.validendings`

List of valid network domain endings. All FQDN's must match at least one of these. Multiple endings can be separated by a comma.

Leave empty if all domains should be accepted.

Default: `example.de, example.com`

Example: `example.de, example.com, example.bayern`

1.2.2 `mail.from.address`

Sender field of the emails sent by Orthos.

Default: `orthos-noreply@domain.de`

1.2.3 `mail.smtprelay.fqdn`

The SMTP server that is used to send mails. That should be a company-internal server.

Default: `relay.mail-server.de`

1.2.4 `mail.subject.prefix`

Subject prefix of the emails sent by Orthos. Each mail gets the prefix before the subject itself (e.g.: `[ORTHOS] Orthos password restored`).

A whitespace after the prefix is recommended.

Default: `[ORTHOS]<whitespace>`

1.2.5 `orthos.api.welcomemessage`

Orthos can show you a welcome message, this can be set here.

Default: blank

1.2.6 `orthos.bugreport.url`

The path to the online repo is set here. If errors occur in the Orthos code, they can be reported there.

Default: <https://gitlab.suse.de/orthos-maintainers/orthos2/issues>

1.2.7 `orthos.cli.url`

The path to the Orthos command line interface.

Default: <https://build.suse.de/package/show/Devel:Archteam:Orthos/orthos-cli>

1.2.8 `orthos.configuration.inline.begin`

It marks the beginning of a code inserted from Orthos configuration into a configuration files.

Example: `#—BEGIN ORTHOS SECTION –`

1.2.9 `orthos.configuration.inline.end`

It marks the ending of a code inserted from Orthos configuration into a configuration files.

Example: `#—END ORTHOS SECTION –`

1.2.10 `orthos.debug.dhcp.write`

Here the writing of the DHCP configuration can be activated or deactivated.

1.2.11 `orthos.debug.mail.send`

Here you can activate or deactivate the writing of Orthos Info E-Mails.

1.2.12 `orthos.debug.motd.write`

Here you can activate or deactivate that Orthos the motto of the day when installing a machine.

1.2.13 `orthos.debug.serialconsole.write`

Here the writing of a cscscreen configuration file on the screen server can be activated or deactivated via Orthos.

1.2.14 `orthos.debug.setup.execute`

Here you can activate or deactivate the writing of the machine installation files via Orthos.

1.2.15 `orthos.documentation.url`

The path to the Orthos documentation.

Default: <https://gitlab.suse.de/orthos-maintainers/orthos2/tree/development/docs>

1.2.16 `orthos.web.welcomemessage`

Here you can enter the Orthos welcome message.

Default: blank

1.2.17 `racktables.url.query`

Orthos retrieves the location of a machine via Racktables. It is important to have at least set `orthos_id#{{ id }}` at the end of the call, only then can Racktable find the appropriate machine.

Default: `https://orthos.arch.suse.de/cgi-bin/get_location_from_racktables.pl?orthos_id#{{ id }}`

1.2.18 `remotepower.default.password`

Default password for remote power access.

1.2.19 `remotepower.default.username`

Default user for remote power access.

1.2.20 `remotepower.dominionpx.password`

Password for remote Power Distribution Unit(Dominion PX) access.

Default: xxxxxxxx

1.2.21 remotepower.dominionpx.username.

User for remote Power Distribution Unit(Dominion PX) access.

Default: orthos

1.2.22 remotepower.ipmi.command

Path and command to power cycle over baseboard management controller (ipmitool).

Default: `/usr/bin/ipmitool -I lanplus -H {{ machine.bmc.fqdn }} -U {{ ipmi.user }} -P {{ ipmi.password }} power {{ action }}`

1.2.23 remotepower.ipmi.password

Password for remote power access over baseboard management controller.

Default: xxxxxxxx

1.2.24 remotepower.ipmi.username

User for remote power access over baseboard management controller.

Default: oroot

1.2.25 remotepower.sentry.password

Password for remote Remote Power Manager(sentry) access.

Default: xxxxxxxx

1.2.26 remotepower.sentry.username

User for remote Remote Power Manager(sentry) access.

Default: orthos

1.2.27 serialconsole.ipmi.password

Password for serial over LAN(SOL) over the baseboard management controller.

Default: xxxxxxxx

1.2.28 serialconsole.ipmi.username

User for serial over LAN(SOL) over the baseboard management controller.

Default: oroot

1.2.29 serialization.output.directory

Local directory where the machine object copies are stored after deleting a machine (see [Machines](./adminguide/machine.md) for more information).

Default: /tmp

Example: /usr/share/grave

1.2.30 serialization.output.format

Data format for the machine object copies after deleting a machine. Valid formats are json and yaml (see *Machines* for more information).

Default: json

Example: yaml

1.2.31 setup.execute.command

Here you can store a script that executes Orthos during installation.

Example: /srv/tftpboot/grub2/scripts/setup.py -mac {{ machine.mac_address }} -fqdn {{ machine.fqdn }} -arch {{ machine.architecture.name }} -default {{ choice }} -kernel-options “{{ machine.kernel_options }}” {% if machine.serialconsole %}-serial-console true -serial-baud {{ machine.serialconsole.baud_rate }} -serial-line {{ machine.serialconsole.kernel_device }}{% else %}-serial-console false{% endif %}

1.2.32 ssh.keys.paths

File path(s) to private SSH keys. Multiple paths can be separated by a comma. In production mode (running e.g. on Apache webserver), absolute paths should be used. Each SSH connection tries all keys until one of them matches.

Default: ./keys/orthos2-master-test

Example: /root/.ssh/id_rsa_cobbler_server, /root/.ssh/id_rsa_sconsole

1.2.33 ssh.scripts.local.directory

Local directory holding scripts determined for remote execution (e.g. for machine checks).

Default: ./scripts

1.2.34 `ssh.scripts.remote.directory`

Remote directory where scripts get copied before they get run on the remote system.

Default: `/tmp/orthos-scripts`

1.2.35 `ssh.timeout.seconds`

Set the SSH connecting timeout (in seconds).

Default: `10`

1.2.36 `tasks.daily.executiontime`

Time at which the daily tasks are started. Must be in 24h format.

Default: `00:00`

1.2.37 `virtualization.libvirt.images.directory`

Here stores Orthos the images for the virtual machines.

Default: `/mounts/users-space/archteam/orthos-vm-images`

1.2.38 `virtualization.libvirt.ovmf.path`

Here is the path for the KVM Support UEFI Binary defined.

Default: `usr/share/qemu/ovmf-x86_64-opensuse.bin`

1.2.39 `websocket.cscreen.command`

Local command which gets executed when a serial console gets requested. The service appends the hostname to the command (e.g. `/usr/bin/screen host`). The command can be anything returning a terminal (see websockets for more information).

Default: `/usr/bin/screen`

1.2.40 `websocket.port`

The port on which the WebSocket service is listening (see websockets) for more information).

Default: `8010`

1.3 WebSockets

1.3.1 Serial console in the web browser

Orthos provides the ability to use an interactive serial console in the web browser for each machine that has an entry for a serial console. The WebSocket communication protocol is used for this. To start the WebSocket service, the following command line must be executed (no HTTPS/TLS):

```
$ orthos-admin serialconsole-ws --start
Start websocket daemon on port 8010...
```

With HTTPS/TLS:

```
$ orthos-admin serialconsole-ws --start --wss\
  --cert <certificate>\
  --key <key>
Start websocket daemon on port 8010...
```

Example:

```
$ orthos-admin serialconsole-ws --start --wss\
  --cert /etc/apache2/ssl.crt/orthos.network.tld.crt\
  --key /etc/apache2/ssl.key/orthos.network.tld.pem
Start websocket daemon on port 8010...
```

Further configuraton information can be found in the [Administrator's Guide](#) (websocket.*).

When the service establishes a SSH connection to invoke a command on a remote server, a SSH key without password is recommended. The key path just needs to be added to the SSH command (-i <identity_file>).

The output is available in the machine view under the tab **Serial Console**.

This feature was tested with Chromium Version 62.0.3202.89 (64-bit) and Firefox 52.5.0 (64-bit).

1.4 Tokens

Usually no administration is required for tokens. When users use the command line client for the first time, they must authenticate themselves with the user / password, after which a token is written for the user. After the process, it is no longer necessary to log on via CLI, since this is then done via the token.

```
-----
|      CLI      | |-----|
| User/Password |---->----| |      |
|-----| | Orthos Server |
|-----| | User -> Token |
|      CLI      |----<----| |      |
| User / Token  |-----|
|-----|
```

1.5 Machines

Machines are the elementary objects in Orthos. They are entities like Servers, Desktop PCs, Virtual Machines (VM), Baseboard Management Controller (BMC), Remote Power devices etc. Machines can be administrative in order to serve as serial console, DHCP server etc. Administrative machines typically are productive and must not be touched. Users are able to reserve machine objects. The following figure shows the machine object organization in general:



1.5.1 Machine fields description

FQDN (required)

The fully qualified domain name must be DNS resolvable by the Orthos server. The domain must be valid according to the valid domain endings configured in Server Configurations (key domain.validendings).

Example: bach.arch.suse.de

Enclosure

The corresponding enclosure object. A enclosure object represents the physiscal chassis of a machine. If no enclosure object is selected, a new object will be created (using the machines hostname).

Example: bach

MAC address (required)

Media Access Control address (MAC) of the primary network interface. This address is used for e.g. DHCP configuration.

Example: 3C:A8:2A:10:74:C3

Architecture (required)

The architecture of the machine. The architecture should be equal to `uname -m` output of the Linux system running in the machine-to-be.

Example: `x86_64`

System (required)

The type of system of the machine.

Example: VM, BMC, BareMetal, LPAR, Desktop, RemotePower etc.

Serial number/Product code/Platform

Specified by the manufacturer. Mostly there is a sticker with the serial number and the product code on the machine's chassis, which has to be entered here.

Example: Serial number: `GPKLDV6120104` / Product code: `MAKL1K1PFB` / Platform: `Knights Landing`

Kernel option

Kernel options are passed via boot loader to the kernel. New values can be set here, or appended with a plus (+). You can find explanations of the kernel options in the GRUB 2 documentation.

Administrative machine

Important and productive administration servers inside the orthos network environment like DHCP, remote power switches, etc. "Do not touch"!

Example: RemotePower, SerialConsole, DHCP Server etc.

NDA Hardware

It marks machines from the Non-disclosure Agreement (NDA) program. Do not publish any details of these machines! If you exchange info like `lspci` or whatever hardware specifics, make sure that the bug you post this info in is marked private and people are aware of the sensitivity of the data you post/send.

Active

Machine is active and can be used from everybody. If deactivated (administrator rights needed), the machine is hidden for ordinary users.

Dedicated VM host

It marks whether the machine can be used as VM host in Orthos. VM guests can then be created on the machine and again managed by Orthos.

Delete automatically

Determine whether the VM Guest is deleted after the end of the reservation.

Max VMs

Maximum number of possible guest systems on the VM host. The number of VM guests depends on the hardware properties of the host.

Virtualization API

The solution that provides virtualization on the host system.

Example: libvirt etc.

Check connectivity

Determined whether and how Orthos checks the accessibility of the machine.

Example: Ping, SSH or SSH with login

Collect system information

Orthos can scan the machines and make them available to the system for information.

Example: dmesg, dmicode etc.

DHCPv4

How to handle the DHCPv4 server v4.

Example: exclude, write DHCPv4 record or ignore DHCPv4 request

DHCPv6

How to handle the DHCPv6 server v6.

Example: exclude, write DHCPv6 record or ignore DHCPv6 request

DHCP filename

Here you can store a machine-specific boot file for PXE and UEFI. See also the GRUB2 documentation.

1.5.2 SERIAL CONSOLE description**Type**

Access type to the serial console of the machine.

Example: Telnet, IPMI, free command etc

CScreen server

A cscreen server is a server on which the cscreen service is installed and entered.

Example: sconsole1.arch.suse.de

Baud rate

Serial console transfer rate.

Example: 115200, 57600, 9600

Kernel device

Kernel device on which the kernel outputs the serial signal.

Example: 0, 1 etc.

Management BMC

Here a BMC for serial over lan can be selected, it must be created similar to a machine.

Example: bahama-sp.arch.suse.de

Dedicated console server

A dedicated console server is an embedded device which is only for merging multiple consoles and then deploying. Access is via telnet. Access to the console runs via the CScreen server.

Example: sconsole3.arch.suse.de

Device

Kernel device through which the output for the serial console runs.

Example: ttyS0, ttyS1 etc.

Port

Network port for accessing the serial console.

Command

A free command can be entered here.

Example: telnet sconsole3.arch.suse.de 2008

1.5.3 REMOTE POWER description

Type

Access type to the RemotePower console of the machine.

Example: Telnet, IPMI etc.

Management BMC

BMC can be selected, it must be created similar to a machine.

Example: bahama-sp.arch.suse.de

Remote power device

Here a RemotePower device can be selected, it must be created similar to a machine.

Example: rpower1.arch.suse.de

Port

Network port for accessing the RemotePower.

Comment

Comment indicating the remote power device.

1.5.4 Delete a machine

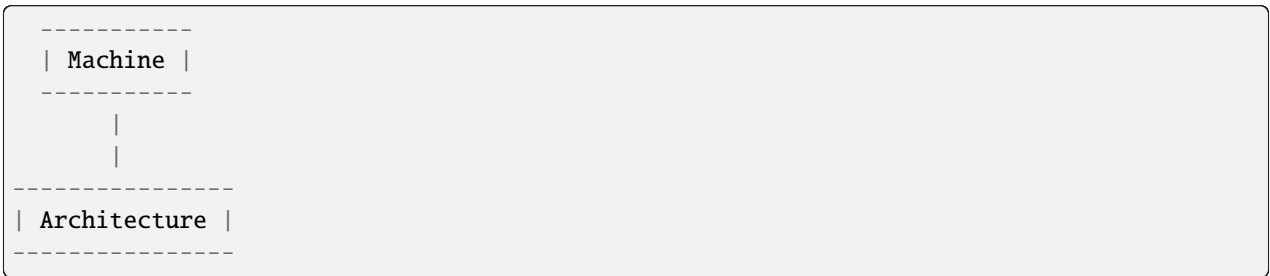
To delete a machine, choose from the machine list and press ‘Delete’ at the bottom of the machine view. All related information that is also deleted together with the machine object is displayed. Press Yes to confirm. For administrative reasons, a copy of each deleted machine object is stored in the form of a file. The format (JSON, Yaml) as well as the target directory can be set via the server configuration.

Further configuration information can be found in the *Administrator’s Guide* (`serialization.*`).

Note: When running in production mode, make sure the target directory (`serialization.output.directory`) can be written by the webserver user.

1.6 Architectures

Machine architectures can be set up, the most common ones are already preconfigured. Machine objects have an architecture that can then be mapped to the machine.



1.6.1 Architecture fields description

Name (required)

Name of the architecture.

Example: s390x, x86_64, ppc64le etc.

DHCP filename

Path to the boot files on the TFTP server.

Contact email

Email address to the person who is the contact person for an architecture.

Write DHCPv4

Enable/disable writing a DHCPv4 file for this architecture.

Write DHCPv6

Enable/disable writing a DHCPv6 file for this architecture.

1.7 Domains

Here you can set up a domain under Orthos, a domain must also be set up on a DNS server. That Orthos can install machines, it is absolutely necessary to specify the DHCP server, TFTP server and the architectures. Orthos is able to write the DHCP file for these domains. The FQDN of a machine object is bound to a domain.

| | | | |
|--------|-------|---------|--|
| | | ----- | |
| | | Machine | |
| | | | |
| ----- | | ----- | |
| Domain | ----- | FQDN | |
| ----- | | ----- | |

1.7.1 Domain fields description

Name (required)

The domain name must be a valid domain, this has to be stored on the DNS and the domain must be valid according to the valid domain endings configured in Server Configurations (key domain.validendings).

Example: Key: suse.de, suse.cz

DHCP Server

Responsible DHCP server for the domain.

Example: music.arch.suse.de

TFTP Server

TFTP server from which you are installing your machines. Here, for example, are: Grub2 files, Installation Scripts, Installation Images, Autoyast files and so on.

Example: music.arch.suse.de

Setup architectures

A domain can contain several architectures, for these a corresponding DHCP file is then written and the appropriate installation files and scripts are offered.

Example: x86_64, ppc64le, s390x

Setup machine groups

Here it is possible to add machine groups.

1.8 Systems and Enclosures

1.8.1 Systems

Systems are assigned to the machines accordingly, here the type of machine is determined. Most known systems are already stored in Orthos.

| | | |
|-------|-----------|---------------|
| ----- | | |
| | Enclosure | |
| ----- | | |
| | | |
| | | |
| ----- | | |
| | Machine | ----- System |
| ----- | | |

System fields description:

Name (required)

Exact name of the system.

Example: BMC, Desktop, LPAR PowerPC etc.

Virtual

Use this if the system a virtual system (VM Gast).

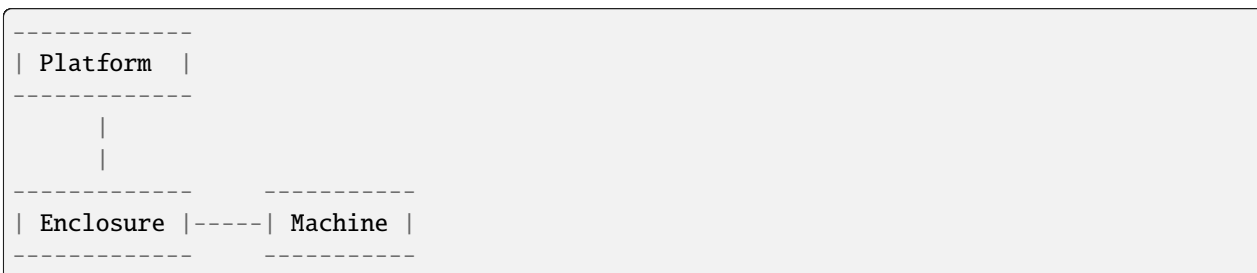
Administrative

Machines that are declared as administrative are machines that belong to the Orthos system and network management. These machines cannot be reserved by the user.

Example: RemotePower

1.8.2 Enclosures

Under Enclosures the case (unit) is defined. A parent name for a unit with several devices can be defined. Several machines can then be assigned to the enclosure. If you create a machine object and you do not specify an enclosure, the name of the enclosure is formed from the FQDN of the machine.



System fields description:

Name (required)

Name of the Enclosure.

Platform

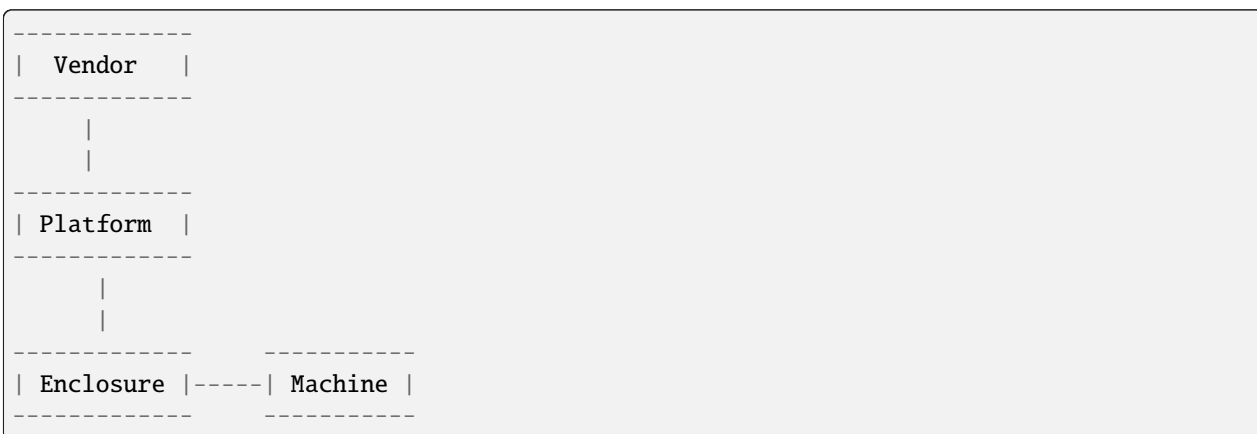
Here the machine platform is specified, this usually comes via the vendor.

Description

Since Enclosure is the generic term of a complete system, this should be as precise as possible.

1.9 Vendors and Platforms

Under Vendors you can enter the machine manufacturer, This serves together with the platforms to recognize a machine or machines of the same type. Most manufacturers are already stored in the Orthos system. With platforms you can set the machine family name (unit name). It goes hand in hand with the vendor, the platform name is determined by the vendor.



1.9.1 Vendor fields description

Name (required)

Name of the vendor.

Example: AMD, IBM, Raritan, Dell, Intel, SGI etc.

1.9.2 Platform fields description

Name (required)

Name of the platform specified by the manufacturer. On x86 systems you often get an idea via dmidecode command, e.g.: `` dmidecode -s system-product-name Latitude E7470 ``

But this info is often empty or wrong, especially on early development machines. Therefore it has to be filled manually. It should be a meaningful name, by which people who are familiar with products of the vendor have an idea what kind of machine this is (how old, features, etc.).

Vendor (required)

Here a vendor is specified and the connection between platform and vendor created.

Example: AMD, IBM, Raritan, Dell, Intel, SGI etc.

Cartridge/Blade

If the platform is a cartridge/blade system, a hook must be set here.

Description

More information about the platform.

1.10 Serial Console Type

Here you can set up the different ways to access a serial console. The best known ones like IPMI, ILO2, Telnet, libvirt etc. are already predefined. The serial console type can then be converted into a machine object under: Machine > SERIAL CONSOLE > Type can be set.

| | |
|-------------------------------------|-------|
| ----- | ----- |
| Machine ----- Serial Console Type | |
| ----- | ----- |

In the following are the fields for the Serial Console Type with explanations.

1.10.1 Name (required)

Name of the Serial Console Type.

1.10.2 Command

Command which is required to access the serial console. In Orthos you use variables/objects to create a serial console type. These are written in double curly brackets `{{ ... }}`.

Example: `virsh -c lxc+ssh://root@{{ machine.hypervisor.fqdn }}/system console {{ machine.hostname }}`

The objects are taken from the machine object or the respective entries at the machine. Such as, for example.

- Hostname = `machine.Hostname`
- BMC FQDN = `machine.bmc.fqdn`
- Hypervisor FQDN = `machine.hypervisor.fqdn`
- Console Server FQDN = `console_server.fqdn`
- IPMI User = `ipmi.user`
- IPMI Password = `ipmi.password`

1.10.3 Comment

Additional information to the Serial Console Type.

USER GUIDE

Describes how to use Orthos as a user, how to find, save and install machines for your purposes with the help of Web GUI and the Command Line Interface (CLI).

- *Introduction*
- *Architecture*
- *Web Client*
 - *Landing Page*
 - *Machine Page*
 - *Virtual Machines*

2.1 Introduction

Orthos is the machine administration tool of the development network in the ARCH team at SUSE. It is used for following tasks:

- locating a machine,
- getting the state of the machine,
- overview about the hardware,
- overview about the installed software (installations),
- reservation of the machines,
- generating the DHCP configuration,
- reboot the machines remotely and
- managing remote consoles.

This document describes the usage. For implementation specific notes, please refer to the “Developer’s Guide”. However, sometimes this document describes some aspects of the implementation simply to let the user better understand why things work as they do. All users of Orthos have a very technical background.

2.2 Architecture

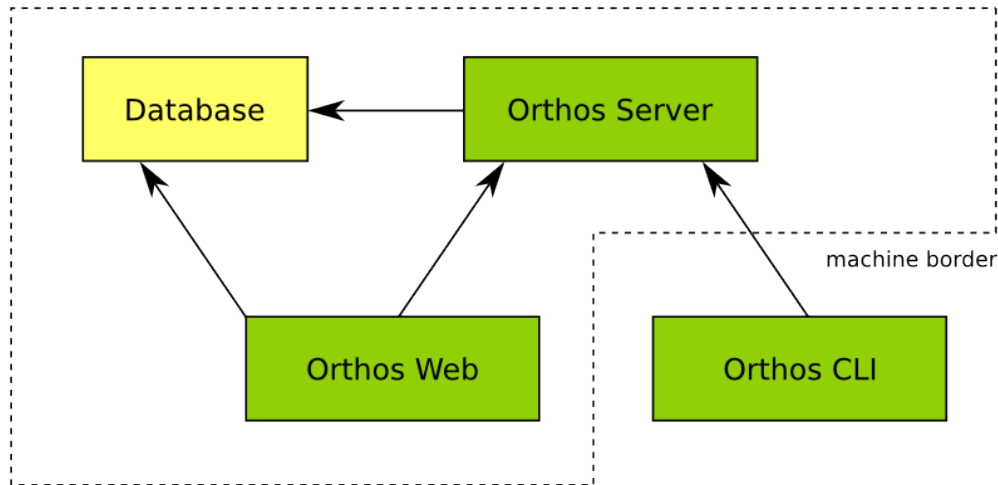


Image Overall Architecture shows the overall architecture. All data is kept in the database. The Orthos Server is the central component of Orthos. It has the following tasks:

- gathering machine data,
- communicating with the clients,
- executing jobs such as rebooting or installing the machine or syncing some files.

There are two clients: The command line interface and the web interface. While the web interface is able to deal with basic tasks such as getting a machine overview and reserving a machine, the CLI is able to deal with more complicated tasks such as executing queries and even edit the data.

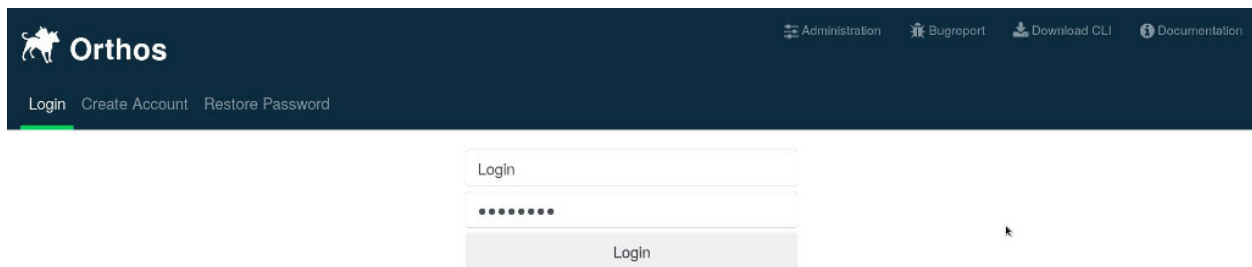
2.3 Web Client

Via the link to your webserver domain it is possible to use the Orthos Web Client. The Web Client is mostly self-explanatory, so only the most important things are explained. You can log in via the web client with an LDAP user account. This is a normal user and has no administrative rights. This means that it is not possible to create and delete machine objects yourself and so on. With this account it is possible to use machines.

You can:

- Search for suitable machines
- Reserve machines for you
- Install the machines you need with SLES, SLED, openSUSE Leap and openSUSE tumbleweed,...
- Perform various machine checks
- Get information about machines and test their accessibility
- Add annotations for each machine (Machine Features, Glitches, Hardware Upgrades, Bios Updates, etc.)
- Report Problems for each machine
- Get an overview of the reservation history

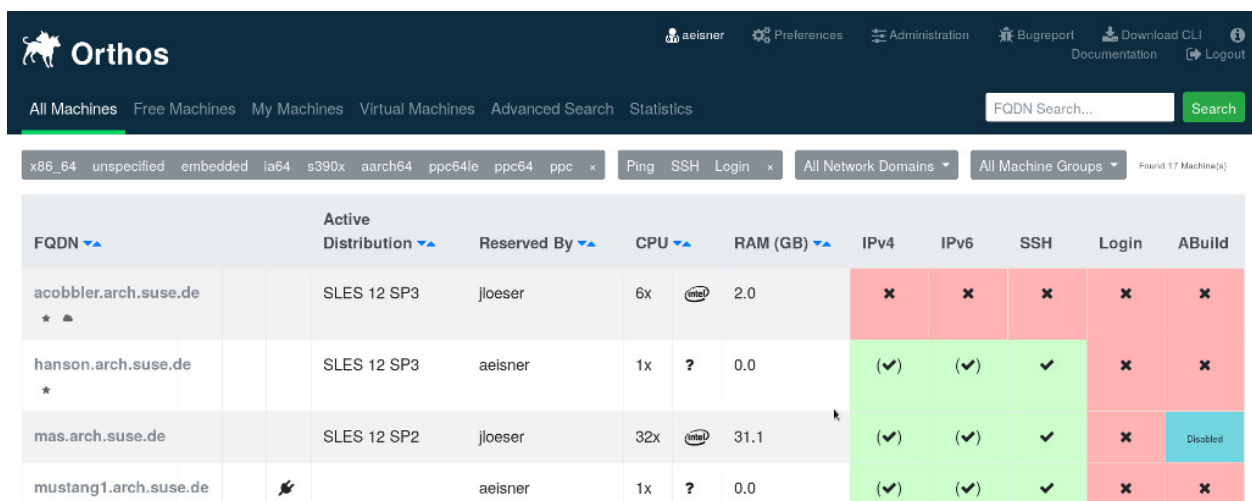
If administrative rights are required, they must be set up by an Orthos administrator. Below a screenshot of the login page.



The image shows the Orthos landing page login form. At the top, there is a navigation bar with the Orthos logo and links for Administration, Bugreport, Download CLI, and Documentation. Below the navigation bar, there are links for Login, Create Account, and Restore Password. The main content area contains a login form with a 'Login' label, a password input field (masked with dots), and a 'Login' button.

2.3.1 Landing Page

After logging in you will be redirected to the Orthos landing page. Here you will find a direct overview of all machines that are available in Orthos.



The image shows the Orthos landing page after login. The top navigation bar includes the Orthos logo, user 'aeisner', and links for Preferences, Administration, Bugreport, Download CLI, Documentation, and Logout. Below the navigation bar, there are tabs for All Machines, Free Machines, My Machines, Virtual Machines, Advanced Search, and Statistics. A search bar labeled 'FQDN Search...' with a 'Search' button is also present. Below the search bar, there are filters for architecture (x86_64, unspecified, embedded, ia64, s390x, aarch64, ppc64le, ppc64, ppc), availability (Ping, SSH, Login), network domains, and machine groups. A table displays the machine overview with columns for FQDN, Active Distribution, Reserved By, CPU, RAM (GB), IPv4, IPv6, SSH, Login, and ABuild. The table shows four machines: acobbler.arch.suse.de, hanson.arch.suse.de, mas.arch.suse.de, and mustang1.arch.suse.de. The status of each machine is indicated by a color-coded background: red for unavailable, green for available, and blue for disabled.

| FQDN | Active Distribution | Reserved By | CPU | RAM (GB) | IPv4 | IPv6 | SSH | Login | ABuild |
|-----------------------|---------------------|-------------|-----|----------|------|------|-----|-------|----------|
| acobbler.arch.suse.de | SLES 12 SP3 | jloeser | 6x | 2.0 | ✗ | ✗ | ✗ | ✗ | ✗ |
| hanson.arch.suse.de | SLES 12 SP3 | aeisner | 1x | 0.0 | ✓ | ✓ | ✓ | ✗ | ✗ |
| mas.arch.suse.de | SLES 12 SP2 | jloeser | 32x | 31.1 | ✓ | ✓ | ✓ | ✗ | Disabled |
| mustang1.arch.suse.de | | aeisner | 1x | 0.0 | ✓ | ✓ | ✓ | ✗ | ✗ |

Here different possibilities are available to get to machines or to get an entire machine status.



The image shows the Orthos landing page after login. The top navigation bar includes the Orthos logo, user 'aeisner', and links for Preferences, Administration, Bugreport, Download CLI, Documentation, and Logout. Below the navigation bar, there are tabs for All Machines, Free Machines, My Machines, Virtual Machines, Advanced Search, and Statistics. A search bar labeled 'FQDN Search...' with a 'Search' button is also present. Below the search bar, there are filters for architecture (x86_64, unspecified, embedded, ia64, s390x, aarch64, ppc64le, ppc64, ppc), availability (Ping, SSH, Login), network domains, and machine groups. A table displays the machine overview with columns for FQDN, Active Distribution, Reserved By, CPU, RAM (GB), IPv4, IPv6, SSH, Login, and ABuild. The table shows four machines: acobbler.arch.suse.de, hanson.arch.suse.de, mas.arch.suse.de, and mustang1.arch.suse.de. The status of each machine is indicated by a color-coded background: red for unavailable, green for available, and blue for disabled.

- All Machines: Overview of all machines that are available in Orthos.
- Free Machines: Overview of all machines that are available in Orthos.
- My Machines: Overview of all Orthos machines reserved under your name.
- Virtual Machines: Overview of all virtual machines. (Host/Gast).
- Advanced Search: Advanced machine search.
- Statistics: Statistics about the machines located in Orthos.



The image shows the Orthos landing page after login. The top navigation bar includes the Orthos logo, user 'aeisner', and links for Preferences, Administration, Bugreport, Download CLI, Documentation, and Logout. Below the navigation bar, there are tabs for All Machines, Free Machines, My Machines, Virtual Machines, Advanced Search, and Statistics. A search bar labeled 'FQDN Search...' with a 'Search' button is also present. Below the search bar, there are filters for architecture (x86_64, unspecified, embedded, ia64, s390x, aarch64, ppc64le, ppc64, ppc), availability (Ping, SSH, Login), network domains, and machine groups. A table displays the machine overview with columns for FQDN, Active Distribution, Reserved By, CPU, RAM (GB), IPv4, IPv6, SSH, Login, and ABuild. The table shows four machines: acobbler.arch.suse.de, hanson.arch.suse.de, mas.arch.suse.de, and mustang1.arch.suse.de. The status of each machine is indicated by a color-coded background: red for unavailable, green for available, and blue for disabled.

- x86_64, unspecified, embedded, ia64, s390x, ppc64le etc: Sorting by architecture
- Ping, SSH and Login: Sort by availability status.
- All Network Domains: Sort by network domain.
- All Machine Groups: Overview and sorting of machines in the respective machine group.

2.3.2 Machine Page

Each machine (machine object) has its own page here you have the possibility to get more detailed information about the machine, to request the status of a machine, to have machines scanned, to install the machine with a new OS, to open error tickets and to write annotations to the machines.

hanson.arch.suse.de NUC Server Room 2 Extension > NUC SRV2Ext-A-9 > 16,17

Overview CPU Network Installations PCI USB SCSI Miscellaneous Reservation History

General Information

| | |
|---------------|---------------------------------|
| FQDN | hanson.arch.suse.de |
| Architecture | x86_64 |
| IPv4 | 10.161.16.38 |
| IPv6 | FE00:113:5Fcd:8000:10:161:16:33 |
| System Type | Basic/Full |
| Comment | - |
| Product Code | 778452-821 |
| Serial Number | 7M244034Q |
| NBA Hardware | |
| ID | 2704 |

System Information Joiner

| | |
|------------------------|-------------------|
| MAC Address | 9C:5E:54:E2:82:8D |
| CPU Cores / Threads | 1 / 1 |
| RAM (GB) | 0.0 |
| BIOS Version | |
| IPMI Supported | x |
| EPI Booted | x |
| VM Capable | x |
| Dedicated VM Host | x |
| Installed Distribution | SLES 12 SP3 |
| Kernel Options | |

Status Last check: 30 July 2018, 04:00 (CEST)

| IPv4 | IPv6 | SSH | Login | ABuild |
|------|------|-----|-------|--------|
| | | | | |

Reservation Current reservation data

| | |
|--------|--------------------------------------|
| Owner | netwin |
| Period | 1. Nov. 2017, 08:20 (CET) - infinite |
| Reason | TESTING cobble |

Platform Hardware / Operating System

| Endosure | System |
|----------|---------------------|
| Vendor | HP |
| Platform | ProLiant DL160 Gen9 |

Annotations WARNING: INTERNAL USE ONLY! - WARNING: OPERATIONAL DATA - INTERNAL USE ONLY

Actions

External Ticketing:

Rescan Status:

Rescan All:

Rescan Installations:

Rescan Network Interfaces:

Swap Machine:

Report Problems:

Machine Scans, Infos, Actions and Annotations

Overview CPU Network Installations PCI USB SCSI Miscellaneous Reservation History

- Overview: The most important information about a machine, information about the status, possibility to scan the machine, to reinstall, to report errors and to write annotations.
- CPU, Network, Installation, PCI, USB, SCSI, Miscellaneous and Reservation History: Detailed information on the Subitems.

Status Last check: 30 July 2018, 04:00 (CEST)

| IPv4 | IPv6 | SSH | Login | ABuild |
|------|------|-----|-------|--------|
| | | | | |

- IPv4 / IPv6: Ping status of a machine IPv4 and IPv6.
- SSH: Orthos tries if it would be possible to establish an SSH connection.
- Login: If a connection with SSH is possible, Orthos tries if a login is also possible.

The scan behaviour of Orthos can be defined by an administrator for the respective machine object.

Annotations

Machine Features, Glitches, Hardware Upgrades, Bios Updates, etc.

Upgrade Kit V2.0 (admin 2018-03-13)

Upgrade Kit V1.2 (admin 2018-03-13)

Add Annotation (no Bugreports)...

Save

Additional machine information should be entered here. For example, upgrades, hardware configuration changes, etc.

Actions

Reserve Machine

Rescan Status

Rescan All

Rescan Installations

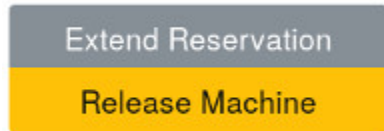
Rescan Network Interfaces

Setup Machine

Report Problem...

- **Reserve Machines:** Here it is possible to reserve a machine under your name. In general, make sure that machines are only reserved for as long as you actually need them. A maximum of 90 days is planned. Please remember that other users may also need the machine. If you need a machine for a longer period of time, only an Orthos administrator can make reservations under your name for longer time periods, up to infinite for constant machine assignment.
- **Rescan Status:** Rescan the status information of a machine.
- **Rescan All:** Rescan all information of a machine.
- **Rescan Installations:** Rescan the installation status of a machine.
- **Rescan Network Interfaces:** Rescan the machine network interfaces.

- **Setup Machine:** Here you can install your machine according to your needs. You have the possibility to install SLES, SLED, Opensuse Leap, Opensuse and Tumbleweed. During the installation you have several options: install, install ssh install ssh auto, install auto etc.
- **Report Problem:** If you unexpectedly encounter a problem with the machine, you can create a support ticket here.



- **Release Machine:** This field is only for machines that are reserved under your name. Here you have the possibility to release the machine for other users.

2.3.3 Virtual Machines

In Orthos it is possible that you work with virtual machines. You can work with a virtual machine as well as with a bare metal machine. You can use the Power Cycle and access the console.

 A screenshot of the Orthos2 web interface showing the 'Virtual Machines' section. The interface includes a top navigation bar with the Orthos logo and user 'aisner'. Below the navigation bar, there are tabs for 'All Machines', 'Free Machines', 'My Machines', and 'Virtual Machines' (which is selected). A search bar labeled 'FQDN Search...' is on the right. Below the tabs, there are filters for architecture (x86_64, unspecified, embedded, ia64, s390x, aarch64, ppc64le, ppc64, ppc) and actions (Ping, SSH, Login). A table lists the virtual machines with columns: FQDN, Active, Distribution, Reserved By, CPU, RAM (GB), IPv4, IPv6, SSH, Login, and ABuild. One machine is listed: 'mas.arch.suse.de' with distribution 'SLES 12 SP2', reserved by 'jloeser', CPU '32x intel', RAM '31.1', and status 'Disabled'.

| FQDN | Active | Distribution | Reserved By | CPU | RAM (GB) | IPv4 | IPv6 | SSH | Login | ABuild |
|--------------------|--------|--------------|-------------|-----------|----------|------|------|-----|-------|----------|
| mas.arch.suse.de + | | SLES 12 SP2 | jloeser | 32x intel | 31.1 | (✓) | (✓) | ✓ | ✗ | Disabled |

Under Virtual Machine select a VM host and click on the plus (+) to create a VM guest.

mas.arch.suse.de

| | |
|--------------------------------------|---|
| System: | VM KVM |
| Architecture: | x86_64 |
| Memory (MB): | 2048 |
| | <small>Value between 512MB and 16384MB.</small> |
| Image: | None |
| Disk size (GB): | 10 |
| | <small>Value between 10GB and 50GB; applies only if no image is selected.</small> |
| NICs (max. 5): | 1 |
| Use UEFI boot: | <input type="checkbox"/> |
| Enable VNC: | <input checked="" type="checkbox"/> |
| Parameters: | <input type="text"/> |
| | <small>e.g. '--cdrom /dev/cdrom'</small> |
| <button>Add Virtual Machine</button> | |

After Add Virtual Machines the VM host is created and made available under the My Machines

ORTHOS CLIENT

3.1 Command Line Interface (CLI)

3.1.1 Basics

The Orthos CLI is a simple interactive shell. You can perform following tasks with the Orthos CLI:

- execute complex queries such as “which machines have more than 4 CPUs” and are reserved by Manfred
- reserve or release a machine
- trigger installation of a machine
- view detailed machine information (such as dmidecode, lspci, lsmod or hwinfo)
- turn power on/off or just reboot a machine
- edit machine data
- Sync machines with a cobbler server
- Trigger machine data scans
- ...

3.1.2 Install the CLI

Python 3 is recommended.

```
zypper in orthos-client
```

In the Orthos Web Client you will find also the download link [Download CLI](#) to the CLI, here you have the possibility to download the suitable client for your distribution(.rpm). The button is in the top right corner.

3.1.3 Connecting the CLI to the Orthos Server

The CLI has several options how you can set up the connection to the Orthos server and adapt it to your needs.

```
you_machine:~ # orthos --help
usage: orthos [-h] [-H HOST] [-P PORT] [-U USER] [--password PASSWORD]
              [--token TOKEN] [-D] [-L FILE] [--no-pager] [-p] [-F IFS]
              [-q] [-v] [--timezone TZ]
```

(continues on next page)

(continued from previous page)

Orthos command line interface.

optional arguments:

| | |
|--------------------------------------|--|
| <code>-h, --help</code> | show this help message and exit |
| <code>-H HOST, --host HOST</code> | use the hostname specified on the command line instead of the one in the config file |
| <code>-P PORT, --port PORT</code> | use the port specified on the command line instead of the one in the config file |
| <code>-U USER, --user USER</code> | use the username specified |
| <code>--password PASSWORD</code> | Better use <code>--token</code> instead (Create a token in the preferences section of the user on the web interface of the server) |
| <code>--token TOKEN</code> | use this token for automatic authentication (e.g. for scripting); <code>-U/--password</code> options will be ignored |
| <code>-D, --debug</code> | write debugging output |
| <code>-L FILE, --logfile FILE</code> | use that together with <code>-D</code> to log the debug output in a file rather than the console |
| <code>--no-pager</code> | do not use pager when showing results |
| <code>-p, --plain-output</code> | print plain output (e.g. for scripting) |
| <code>-F IFS, --ifs IFS</code> | set internal field separator (only useful in combination with <code>-p</code> ; default is <code>\$OIFS</code>) |
| <code>-q, --quiet</code> | makes command line client quiet |
| <code>-v, --version</code> | print version output |
| <code>--timezone TZ</code> | set the local time zone (default is "Europe/Berlin") |

Example for the connection to the Orthos Server:

```
your_machine:~ # orthos -H https://orthos-next.arch.suse.de
(orthos 2.0.0)
```

3.1.4 Orthosrc configuration file

The orthos package installs an empty configuration file in `/usr/share/doc/packages/orthos-clientorthosrc`. Once you connected to the orthos server, it is recommended to copy this file to your local config directory: `~/.config/orthosrc` and adapt the file to your needs.

3.1.5 Orthos Command Examples

HELP

Provides help and shows you an overview of the available Orthos CLI commands.

```
(orthos 2.0.0) help
```

Commands are:

| | |
|------------|---|
| INFO | Retrieve information about a machine. |
| POWER | Power cycles a machine. |
| REGENERATE | Regenerate machine-related or service files. |
| RESCAN | Rescan a machine. |

(continues on next page)

(continued from previous page)

| | |
|--------------------|---|
| SERVERCONFIG | Show server configuration. |
| RESERVATIONHISTORY | Show reservation history of a machine. |
| ADD | Adds information to the database. |
| RELEASE | Release machines. |
| QUERY | Retrieve information about a machine. |
| RESERVE | Reserve machines. |
| SETUP | Automatic machine setup. |
| DELETE | Removes information from the database. |
| ALIAS | Define own aliases. |
| AUTH | Request authorisation manually. |
| EXIT | Exit program. |
| HELP | Provides help. |

(orthos 2.0.0) help info

Command to get information about a machine.

Usage:

INFO <fqdn>

Arguments:

fqdn - FQDN **or** hostname of the machine.

Example:

INFO foo.suse.de

(orthos 2.0.0) help power

Command to power cycle machines **or** the get the current status.

Usage:

POWER <fqdn> <action>

Arguments:

fqdn - FQDN **or** hostname of the machine.

action - Specify new power state. Actions are:

```

on                : Power on.
off               : Power off via SSH. If didn't succeed, use remote power.
off-ssh           : Power off via SSH only.
off-remotepower   : Power off via remote power only.
reboot            : Reboot via SSH. If didn't succeed, use remote power.
reboot-ssh        : Reboot via SSH only.
reboot-remotepower : Reboot via remote power only.
status            : Get power status.

```

Example:

POWER foo.suse.de reboot

(orthos 2.0.0) help reserve

Reserves a machine.

Usage:

(continues on next page)

(continued from previous page)

```
RESERVE <fqdn>
```

Arguments:

```
fqdn - FQDN or hostname of the machine.
```

Example:

```
RESERVE foo.suse.de
```

```
(orthos 2.0.0) help alias
```

Define or display aliases. The command can be called without any arguments, then it displays all available aliases. If it's called with one argument, then it displays the definition of a specific alias. If it is called with more than two arguments, then you can define new aliases.

To execute an alias, type the alias name with a leading '@'.

Usage:

```
ALIAS [alias] [*args]
```

Arguments:

```
alias - Alias name.
*args - Valid command string.
```

Example:

```
ALIAS
ALIAS allmachines query name, ipv4 where name =~ foobar
ALIAS allmachines
```

```
@allmachines
```

QUERY

Retrieve all kind of information about a machine or general orthos data. This is a very powerful command. It is built up similar to a SELECT SQL database statement. In fact it ends up in querying the underlying orthos server database:

```
query foo, bar where binary_attribute and int_attribute > XY and char_attribute =~ "STRING"
```

-> This will search for and show the results (also attributes): foo and bar where the condition after where matches.

Examples:

```
query fqdn, cpu_physical
query fqdn where cpu_model =~ Intel
query fqdn where cpu_model =~ Intel OR !efi
```

More complex queries:

```
# Show full names and installed distributions of all machines which
# are not reserved (!res_by), which do run and have an orthos ssh key
# installed and therefore could be nightly scanned (status_login)
# which are not administrative, x86_64 machines and do have more than 7
# CPU cores:
query fqdn, inst_dist where !res_by and status_login and !administrative and
architecture = x86_64 and cpu_cores > 7
```

Use alias(es) for more complex queries: To permanently define and use above command as an alias (auto-stored in ~/.config/orthosrc), make sure to not use any quoting, just do:

```
alias x86_free_running query fqdn, inst_dist where !res_by and status_login and !
↪administrative and architecture = x86_64 and cpu_cores > 7
```

To use above defined alias (tab completion working...):

```
@x86_free_running
```

Valid operators are:

| | |
|----------|--|
| !<field> | not (binary fields only) |
| == = | exactly equal |
| =~ | contains |
| =* | startswith |
| != | unequal |
| > < | greater or less than (number fields only) |
| >= <= | greater equals or less equals (number fields only) |
| AND | logical conjunction |
| OR | logical disjunction |

3.1.6 Orthos Variables and Objects

The Orthos Client has many objects that you can query from the machine objects in Orthos. The names are self-explanatory and can be used for queries as described under the *query* command above or you use the TAB completion feature to see available *query* attributes.

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`