DICOS OneClick Extension

This a to-PDF-converted version of the original HTML/CSS Documentation

DICOS GmbH

Copyright © 1998 - 2025 DICOS GmbH Kommunikationssysteme

Table of contents

Copyright © 1998 - 2025 DICOS GmbH Kommunikationssysteme

Legal Notice

This documentation and the related products are protected by copyright. They are part of a license agreement and may only be used in accordance with the terms of the contract. No part of this document may be reproduced or transmitted in any form or by any means, or reproduced or modified, or translated into any other natural or machine language. This excludes the creation of a backup copy for personal use. The transfer to third parties of the information given to you is only permitted with the prior written permission of DICOS GmbH Kommunikationssysteme.

1. Introduction

> The DICOS OneClick Extension serves as a GUI frontend for the DCSWatch and ScriptLauncher products in Spectrum OneClick, adding specialized views and renderers.

Version: 2.6.4 | Compatible with: Spectrum 24.3.2

Copyright © 1998 - 2025 DICOS GmbH Kommunikationssysteme

1.1 Quick Access

Installation

How to install the OneClick Extension standalone or with Spectrum

Update

How to Uninstall an old version

Configuration

How to Configure the application

Reference

What (URL-)Endpoints are available, what files are used in the configuration

Script Launcher

How to define a script, configure parameters, and set up permissions

Architecture

How the three-tier architecture works and why it is designed that way

Troubleshooting

Common issues and how to resolve them

DCSWatch

How DCSWatch integrates with OneClickExt for watch management

Copyright © 1998 - 2025 DICOS GmbH Kommunikationssysteme

2. Explanation

2.1 Architecture

This page describes the three-tier architecture of OneClickExt, explaining what each component does, why the system is structured this way, and how the tiers communicate with each other.

2.1.1 Overview

Spectrum Standalone

```
flowchart LR
    user([User])

    subgraph oneclick["OneClick Server"]
        oc[OC-Client]
        sl[Scriptlauncher]
    end

    subgraph ss_host["SpectroSERVER"]
        daemon[OcExt-Daemon]
        ss[SpectroSERVER]
        scripts[Scripts]
    end

    device[Device]

    user -->|HTTP| oc
    oc <--> sl
    sl <-->|RMI| daemon
    daemon <--> ss
    daemon --> scripts
    scripts -->|SSH| device
    ss -->|SSH| device
```

The daemon is typically installed on the SpectroSERVER host (see Installation with Spectrum). It can also run on a dedicated separate host, in which case the daemon is installed using the standalone guide and configured to connect to SpectroSERVER via `oneclickext.mls.primary` and `oneclickext.ss.host`.

```
flowchart LR
    user([User])

    subgraph sl_server["Scriptlauncher Server"]
        sl_servlet[Script Server]
    end

    subgraph scriptserver["Scriptserver"]
        daemon[OcExt-Daemon]
        scripts[Scripts]
    end

    device[Device]

    user -->|HTTP| sl_servlet
    sl_servlet <-->|RMI| daemon
    daemon --> scripts
    scripts -->|SSH| device
```

No Spectrum installation required. Scripts are invoked via the Script Server HTTP interface. See Standalone Installation.

2.1.2 Why a separate server daemon?

The Spectrum CORBA API requires a Spectrum username and runs only on the server host. Rather than embedding CORBA access inside the client applet (which runs in each user's browser session), OneClickExt uses a long-running daemon process that holds the single CORBA connection. The client communicates with this daemon over RMI.

Copyright © 1998 - 2025 DICOS GmbH Kommunikationssysteme

This design has several consequences:

- The Spectrum username (`oneclickext.ss.user`) is configured once on the server, not per user.
- Script execution happens on the server host, not in the browser.
- The daemon can reconnect to Spectrum after a CORBA failure without any client-side action.
- Multiple OneClick Console users share the same daemon and its thread pool.

2.1.3 The two modules

OneClickExt ships with two separately-licensed functional modules:

Module	License code	What it provides
ScriptLauncher	`sl`	Launch shell scripts on managed devices from the Spectrum OneClick Console or a Script Server
DCSWatch	`wt`	Manage Spectrum watch objects: Create DcsWatch, Copy Watch(es), Delete DcsWatch(es), Set DcsWatch Attributes

Both modules are implemented in the same daemon and client. Which ones are active depends on the license.

2.1.4 The client

The client runs inside the Spectrum OneClick Console as a Java Swing application, loaded via JNLP. It connects to the server daemon at startup using RMI.

The RMI connection details are passed to the client as JVM system properties set in the JNLP launch file (`$SPECROOT/tomcat/webapps/spectrum/oneclickext.jnlp`). The installer sets these automatically:

```
oneclickext.rmi.host=<host>
oneclickext.rmi.port=<port>
```

If the primary server is unreachable, the client automatically tries a secondary server configured via:

```
oneclickext.rmi.secondary.host=<host>
oneclickext.rmi.secondary.port=<port>
```

This failover is attempted once at connection time: the client tries primary first, then secondary. If neither responds, the extension menu items are not shown.

When a script item in the menu config specifies a host and port directly (using the `<item name="ScriptName_<hostname>_<port>">` syntax), that address overrides the system properties and both servers are bypassed.

2.1.5 The server daemon

The daemon is a standalone Java process (`OneClickExtServer`) that:

1. Creates a local RMI registry and binds itself to it.
2. Reads its configuration from `oneclickext.props` .
3. Connects to SpectroSERVER via CORBA.
4. Maintains a thread pool (`OneClickExtExecutor`) for running scripts concurrently.

Script execution is queued: incoming `runScript` RMI calls add a `ScriptRunner` to a blocking queue, and worker threads pull from that queue. The number of worker threads is controlled by `oneclickext.exec.threads` .

Configuration hot-reload

The additional properties file list (`oneclickext.additional.props`) is re-evaluated before every script call. This means new scripts can be added or existing ones reconfigured by editing the additional props file, without restarting the daemon. Core settings (RMI ports, SpectroSERVER connection) require a restart.

Copyright © 1998 - 2025 DICOS GmbH Kommunikationssysteme

2.1.6 The Script Server (optional)

The Script Server is a Tomcat servlet (`ScriptServer`) deployed separately from the daemon. It provides HTTP endpoints that allow scripts to be invoked from a web browser or external system without requiring the full Spectrum console.

The Script Server connects to the daemon over RMI (configured via `sl.rmi.host` and `sl.rmi.port`) and delegates all script execution to it. It also handles output history and syslog notifications independently.

See the Script Server reference for configuration details.

Copyright © 1998 - 2025 DICOS GmbH Kommunikationssysteme

3. Guides

3.1 Troubleshooting

This page describes common problems encountered when installing, configuring, and running OneClickExt, along with their causes and solutions. It covers connectivity issues, script execution failures, and configuration loading problems.

3.1.1 Log files

The daemon uses Log4j 2, configured via `$SPECROOT/DICOS/OneClickExt/conf/log4j.xml` (standalone: `$DCSROOT/DICOS/OneClickExt/conf/log4j.xml`). By default, log output is written to:

```
$SPECROOT/DICOS/OneClickExt/logs/oneclickext.log
```

The log file rolls when it reaches 10 MB, keeping up to 10 rotated files (`oneclickext-1.log`, `oneclickext-2.log`, etc.).

To increase verbosity for debugging, change the log level to `DEBUG` in `conf/log4j.xml`:

```
<Root level="DEBUG">
```

Restart the daemon after changing the log configuration.

For the Script Server (Tomcat), logs appear in `$SPECROOT/tomcat/logs/` (standalone: `$DCSROOT/webtomcat/logs/`).

3.1.2 Daemon connectivity issues

The client menu items are missing

The OneClickExt menu entries only appear if the client can successfully connect to the daemon over RMI at startup. If the connection fails silently, nothing appears in the menu.

Check:

- The daemon is running on the server host.
- The RMI port (default: 13689) is reachable from the Spectrum console host. Firewall rules between the console host and the server may block it.
- The `oneclickext.rmi.host` and `oneclickext.rmi.port` properties are set correctly in `$SPECROOT/tomcat/webapps/spectrum/oneclickext.jnlp`. The installer writes these automatically from the values you entered during `setup.sh`.
- The daemon log shows `Server started` without errors.

If the extension loads but individual script entries are missing, they have not yet been added to `custom-menu-config.xml`. See Defining a Script for how to configure script menu entries.

Daemon does not start after Spectrum restart

If the OneClickExt daemon is not running after a SpectroSERVER restart, the idb auto-ticket may have been stopped with `cmdC localhost 2 ONECLICKEXT` (STOP_AUTO_TICKET). This flag persists through Spectrum restarts and prevents the daemon from being brought up automatically by `processd`.

Verify the daemon is not running:

```
ps -fu spectrum | grep OneClickExtServer
```

If no process is found, start the daemon manually:

```
cd $SPECROOT/DICOS/OneClickExt/bin
./OneClickExt.sh &
```


Copyright © 1998 - 2025 DICOS GmbH Kommunikationssysteme

Check the log to confirm startup was successful:

```
tail -20 $SPECROOT/DICOS/OneClickExt/logs/oneclickext.log
```

The log should end with `Server started` and `connected to MOM`.

Extension not loading after installation

If the `CHECKONECLICKEXT` ticket is running but the extension still does not load, a pre-patched JNLP file is available as a fallback at:

```
$SPECROOT/tomcat/webapps/spectrum/oneclick.jnlp.ext
```

Copy it over the original as a one-time manual step:

```
cp $SPECROOT/tomcat/webapps/spectrum/oneclick.jnlp.ext \
   $SPECROOT/tomcat/webapps/spectrum/oneclick.jnlp
```

Warning

This manual copy does not survive a Spectrum upgrade. If `oneclick.jnlp` is regenerated, the step must be repeated.

RMI connection refused

```
Server rmi://<host>:<port>/OneClickExt cannot be contacted
```

Causes:

- The daemon is not running. Start it.
- Wrong host or port in the client configuration.
- A firewall blocks the RMI port and the RMI connection port. Both ports must be open: `oneclickext.rmi.port` (registry) and `oneclickext.rmi.connport` (object communication).

CORBA connection fails

The daemon log shows:

```
waiting for connection to MOM...
unable to connect to MOM!
```

Check:

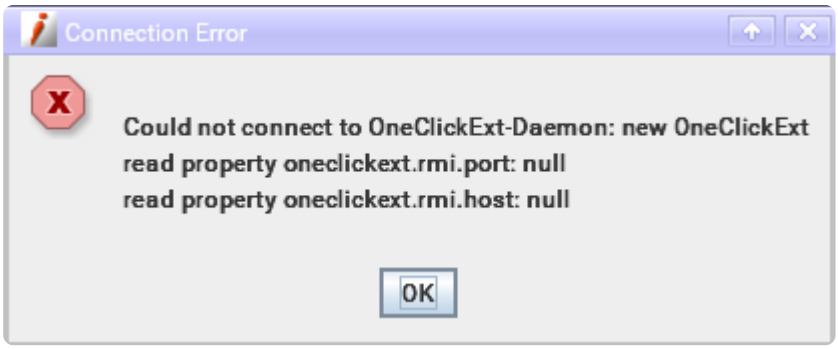
- `oneclickext.mls.primary`, `oneclickext.mls.backup`, and `oneclickext.ss.host` are set correctly.
- The Spectrum user (`oneclickext.ss.user`) exists and has the required access rights in Spectrum.
- SpectroSERVER is running.

The daemon retries the CORBA connection automatically. The retry interval and maximum count are configured with `oneclickext.ss.retryIntervall` and `oneclickext.ss.retryCount`.

Could not connect to OneClickExt-Daemon via OneClick WebApp

If you get the following error popup window when trying to execute a script via the OneClick WebApp console:

Copyright © 1998 - 2025 DICOS GmbH Kommunikationssysteme



Make sure that you have added the RMI parameters to the Webswing JVM Arguments. Refer to the OneClick WebApp Integration for details.

3.1.3 Script execution issues

Script permission denied

For web-invoked scripts, a permission denial is logged as the user not being found in the `scripts` or `groups` configuration. Verify the user name matches what is configured under `oneclickext.scripts.users.*`.

Scriptlauncher not licensed

The daemon log shows:

```
Scriptlauncher not licensed!
```

or

```
Script license count exceeded!
```

The ScriptLauncher module is either not included in the license or the licensed number of scripts has been reached. The license file is located at `$SPECROOT/DICOS/OneClickExt/bin/.license`. Contact DICOS support if the license file is missing or does not include the `sl` module.

Script command not found

```
oneclickext.scripts.<scriptitem>.command
```

If the property is missing or the path does not resolve, the script will fail at execution time. Verify the path using the variable expansion pattern:

```
oneclickext.scripts.path: ../scripts/
oneclickext.scripts.ext:  .sh
oneclickext.scripts.RebootDevice.command: ${oneclickext.scripts.path}RebootDevice${oneclickext.scripts.ext}
```

Check that the resolved path points to an executable file that the daemon's OS user has permission to run.

Script output is empty

- Confirm `oneclickext.scripts.<scriptitem>.showOutput: true`.
- Confirm the script actually writes to stdout. Output sent only to stderr may not be captured depending on the script and OS.
- Check the encoding setting. A mismatch between `oneclickext.scripts.<scriptitem>.encoding` and the script's actual output encoding will produce garbled or missing output. Use `CP437` for Windows batch files and `UTF-8` for Linux shell scripts.

New script does not appear after editing the properties file

The additional properties file (`oneclickext.additional.props`) is reloaded before each script call, but only for scripts that are already known. A completely new script item becomes visible only after the user closes and reopens the script dialog, or after restarting the daemon.

Copyright © 1998 - 2025 DICOS GmbH Kommunikationssysteme

If the script still does not appear, confirm the properties file is listed in `oneclickext.additional.props` and that the file path is correct relative to the daemon's working directory.

3.1.4 Configuration issues

Properties file not loaded

The daemon searches for configuration files in a fixed order (see Configuration Load Order). If the expected file is not loaded, check:

- The file exists at the expected path.
- The daemon's working directory is correct (the `bin/` directory is typical).
- For additional properties files, the paths in `oneclickext.additional.props` are relative to the daemon's working directory.

The daemon logs each additional properties file it loads:

```
new properties <path> loaded.
```

If a file is listed but not found:

```
new properties <path> not found!
```

Changes to core settings take no effect

Core settings such as RMI ports, SpectroSERVER host, and thread pool size require a daemon restart to take effect. They are read once at startup and not reloaded.

Only the additional properties files (those listed under `oneclickext.additional.props`) are hot-reloaded.

Copyright © 1998 - 2025 DICOS GmbH Kommunikationssysteme

3.2 Configuration

3.2.1 Configuration Load Order

This page describes the order in which the OneClickExt daemon searches for and loads its configuration files. Understanding this order is important when multiple configuration files are present, as later files in the sequence take precedence over earlier ones.

The configuration files are searched for and loaded in the following order. All paths are relative to the daemon installation directory (`$SPECROOT/DICOS/` for Spectrum installations, `$DCSROOT/DICOS/` for standalone):

```
OneClickExt/conf/oneclickext.props
OneClickExt/conf/log4j.xml
OneClickExt/conf/version
OneClickExt/oneclickext.vers
OneClickExt/bin/oneclickext.props
OneClickExt/version
```

The parameter `oneclickext.additional.props` is then read and the additional properties are loaded. This list is checked before each script call.

Copyright © 1998 - 2025 DICOS GmbH Kommunikationssysteme

3.2.2 Changing the RMI Port

The RMI port is configured in two places: on the server (daemon) and on the client (the Spectrum console or Script Server). Both must be updated together whenever the port changes, and the daemon must be restarted afterwards.

Spectrum installation

Server side Client side

Edit `$SPECROOT/DICOS/OneClickExt/bin/oneclickext.props` and update:

```
oneclickext.rmi.port:     <new port>
oneclickext.rmi.connport: <new connection port>
```

Restart the OneClickExt daemon for the change to take effect. In the Spectrum OneClick console, select File > Exit — this triggers a restart of the embedded OneClickExt process.

The Spectrum console reads the RMI connection details from the JNLP launch file. These are managed through the OneClick Web Application configuration UI — editing the file directly is not recommended as changes may be overwritten.

1. Open the OneClick Web Application in a browser. By default it is available at:

   ```
   http://<host>:9443/spectrum/
   ```

2. Log in.
3. Select Manage in the upper right corner.
4. Go to the Applications tab and select the OneClick WebApp.
5. Open the App Config tab and scroll down to Java/JVM Arguments.
6. Update the following arguments:

Argument	Description
`-Doneclickext.rmi.primary.port`	RMI registry port (must match `oneclickext.rmi.port` on the server)
`-Doneclickext.rmi.primary.host`	Hostname of the daemon host (only if the host changed)

7. Click Apply. The change takes effect when users next launch the console.

Copyright © 1998 - 2025 DICOS GmbH Kommunikationssysteme

Standalone installation

Both the daemon and any clients (Script Server) read their configuration from properties files.

Daemon Script Server

Edit `$DCSROOT/DICOS/OneClickExt/bin/oneclickext.props` and update:

```
oneclickext.rmi.port:     <new port>
oneclickext.rmi.connport: <new connection port>
```

Restart the daemon for the change to take effect.

Edit the Script Server properties file and update:

```
sl.rmi.port: <new port>
sl.rmi.host: <daemon hostname>
```

Restart Tomcat for the change to take effect.

Copyright © 1998 - 2025 DICOS GmbH Kommunikationssysteme

3.3 Installation

3.3.1 Installation with Spectrum

This page describes how to install OneClickExt as a Spectrum management module on a Linux server. The installation uses the `setup.sh` script included in the distribution archive, which configures the daemon and registers it with Spectrum's process management system.

Prerequisites

- The distribution archive `VCD-DcsOcExt_<version>_<spectrum_version>.tar`
- Root access on the server
- SpectroSERVER and OneClick must be stopped before installation (see below)
- X11 forwarding enabled in your SSH session (`ssh -X` or `ssh -Y`) — the installer launches a GUI window

Step 1 — Stop Spectrum services

Run as the Spectrum install owner (typically `spectrum`):

The installer modifies Spectrum configuration files that cannot be changed while the services are running. Stop them in this order:

```
$SPECROOT/tomcat/bin/stopTomcat.sh
$SPECROOT/webtomcat/bin/shutdown.sh
$SPECROOT/bin/stopSS.pl
```

Wait for both processes to terminate before continuing.

Step 2 — Run the installer

Run as root:

Copy the distribution archive to the server and unpack it:

```
tar xf VCD-DcsOcExt_<version>_<spectrum_version>.tar
```

Run `setup.sh` from the extracted directory:

```
$ ./setup.sh
Installing to /opt/SPECTRUM ...

    OneClickExt Server is myserver (y/n)? y
    Enter the RMI Port (1025-65535)? 16016
    Enter the RMI Connection Port (1025-65535)? 16017
    OneClickExt Server = myserver, RMI Port = 16016, RMI Conn Port = 16017 (y/n)? y

    Hostname of the Main Location Server? myserver
    Hostname of the Backup Location Server? myserver
    SpectroSERVER is myserver (y/n)? y
    SpectroSERVER is myserver, Location Servers are myserver / myserver (y/n)? y

OneClickExtension setup:

  MainLocation Server = myserver
  BackupLocation Server = myserver
  SpectroSERVER = myserver
  Is primary = Y
  Autostart Ticket = Y
  OneClickExt Host = myserver
  RMI Port = 16016
  RMI Connection Port = 16017

saving /tmp/dcssetup.config ...
starting installer ...
```

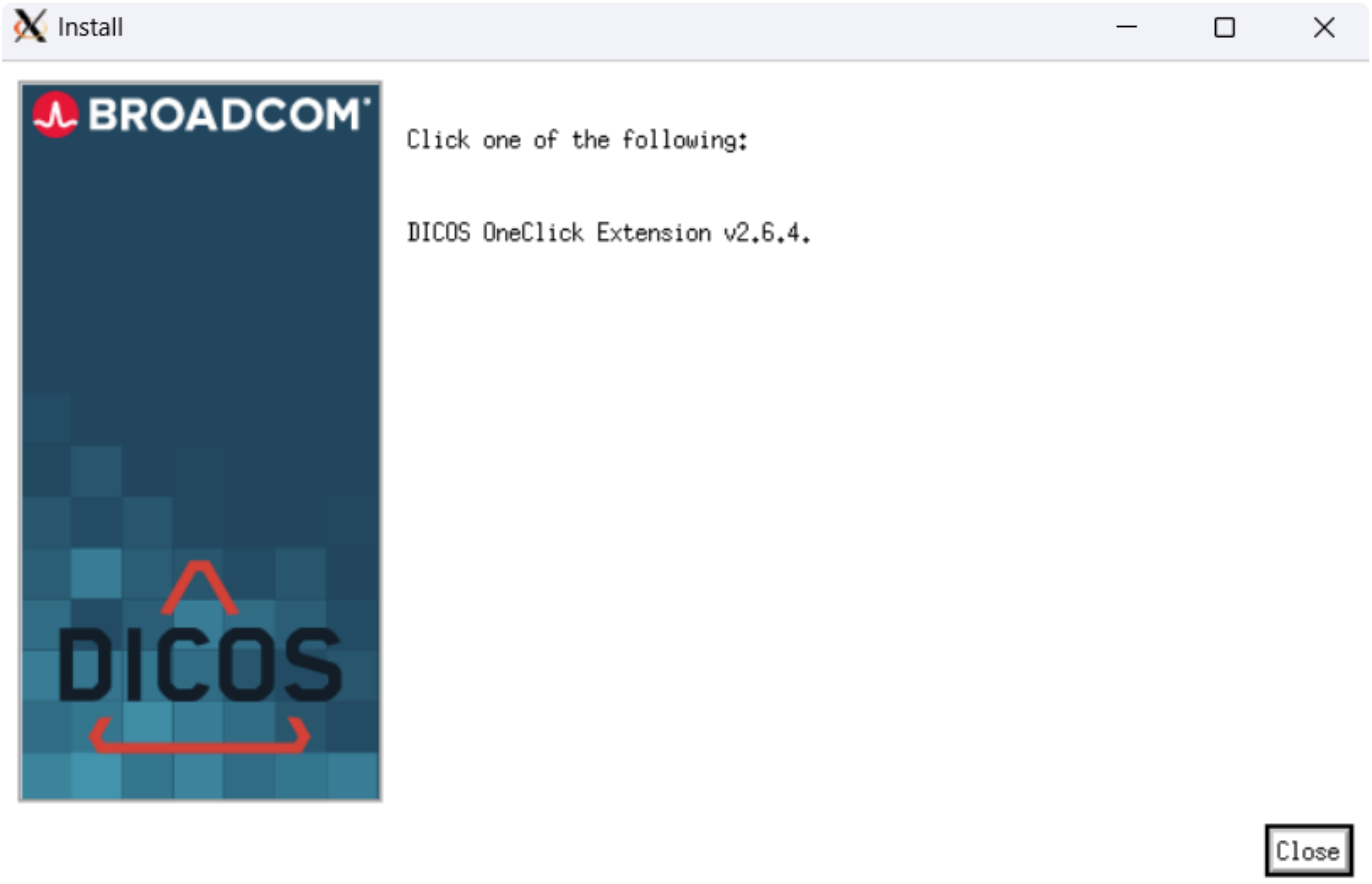

Copyright © 1998 - 2025 DICOS GmbH Kommunikationssysteme

The setup script asks for:

Prompt	What to enter
OneClickExt Server	Hostname of the server running the OneClickExt daemon (usually the same host)
RMI Port	Port the daemon listens on for client connections (default: 13689)
RMI Connection Port	Second port used by Java RMI for object communication (default: 13690)
Main Location Server	Hostname of the Spectrum Main Location Server
Backup Location Server	Hostname of the Spectrum Backup Location Server
SpectroSERVER	Hostname of the Spectrum Management Server (MOM)

Both RMI ports must be open in any firewall between the Spectrum console host and the daemon host.

After the text prompts, `setup.sh` launches a graphical installation wizard. Complete the wizard to finish the installation:



Step 3 — Restart Spectrum services

Run as the Spectrum install owner (typically `spectrum`):

After the installer completes, start the services again:

```
$SPECROOT/bin/startSS.pl
$SPECROOT/tomcat/bin/startTomcat.sh
$SPECROOT/webtomcat/bin/startup.sh
```


Copyright © 1998 - 2025 DICOS GmbH Kommunikationssysteme

Step 4 — Integrate OneClickExt into OneClick

Run as the Spectrum install owner (typically `spectrum`):

The installer registers the `CHECKONECLICKEXT` idb ticket, which automatically patches `oneclick.jnlp` to load the extension. After SpectroSERVER has started, verify the OneClickExt daemon is running:

```
ps -fu spectrum | grep OneClickExtServer
```

The process should appear in the output. If it does not, check the daemon log at `$SPECROOT/DICOS/OneClickExt/logs/oneclickext.log`.

If the menu entries do not appear after starting the console, see Troubleshooting.

To verify the installed version:

```
cat $SPECROOT/DICOS/OneClickExt/oneclickext.vers
```

Step 5 — Integrate OneClickExt into OneClick WebApp

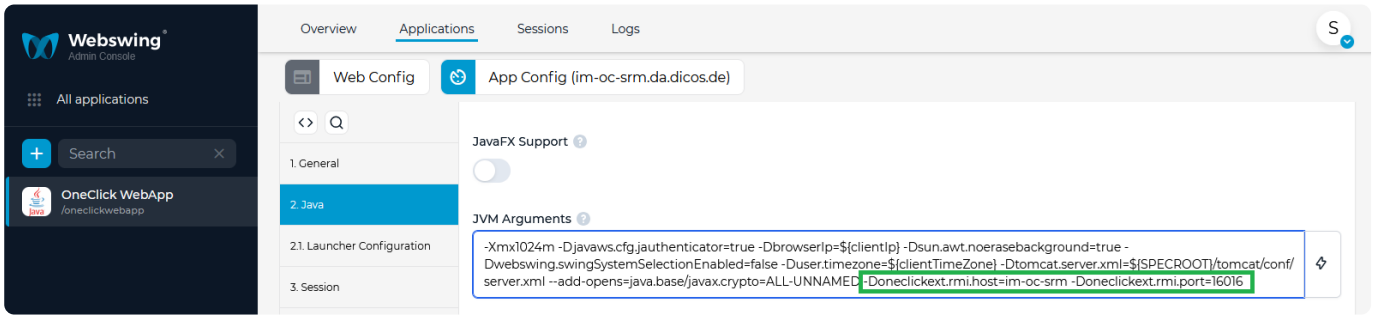
For each OneClick Server where the OneClick Extension has been installed, open the WebSwing administration at `http[s]://<oneclick_url>:9443/admin/apps/oneclickwebapp`.

In the section Applications -> App Config (), add the following to the JVM Arguments property:

```
-Doneclickext.rmi.host=<server> -Doneclickext.rmi.port=16016
```

If running in a fault-tolerant environment, specify the secondary OneClickExt Daemon as well by adding

```
-Doneclickext.rmi.secondary.host=<secondary_server> -Doneclickext.rmi.secondary.port=16016
```



Click `Apply` in the lower right corner to save the settings. For the changes to take effect, the OneClick WebApp console needs to be restarted.

Step 6 — Upgrade notes

When upgrading from a previous version, the new properties file is installed as:

```
$SPECROOT/tomcat/webapps/spectrum/oneclickext.props_<version>
```

The version suffix prevents the installer from overwriting your existing configuration. Either rename this file to `oneclickext.props` (fresh install), or merge its new properties into your existing `oneclickext.props` file (upgrade).

Copyright © 1998 - 2025 DICOS GmbH Kommunikationssysteme

3.3.2 Standalone Installation

This page describes how to install the OneClickExt daemon and the Script Server web application in standalone mode, independently of a Spectrum instance. This setup is used when scripts need to be invoked from a web browser or an external system without the Spectrum console, or when the daemon runs on a separate host.

What you will install

The base installation folder is referred to as `$DCSROOT` (Linux) / `%DCSROOT%` (Windows) throughout this guide. The defaults are `/opt/dcssl` on Linux and `C:\DcsSL` on Windows, but the location can be freely chosen — all internal path references are relative.

Component	Purpose	Windows path	Linux path
Java 17 JDK	Runtime for the daemon and Tomcat	`%DCSROOT%\Java`	`$DCSROOT/Java`
Apache Tomcat	Servlet container for the Script Server	`%DCSROOT%\webtomcat`	`$DCSROOT/webtomcat`
OneClickExt Daemon	The RMI server process	`%DCSROOT%\DICOS`	`$DCSROOT/DICOS`
Script Server web app	HTTP interface for script invocation	`%DCSROOT%\webtomcat\webapps`	`$DCSROOT/webtomcat/webapps`

Prerequisites

Windows Linux

Download the following before starting:

- Java 17 JDK installer (`.exe` or `.msi`) — installed in Step 1
- Apache Tomcat installer for Windows — installed in Step 2
- Distribution archives:
  - `OneClickExtSa_d_<version>.tar.gz` — the daemon
  - `OneClickExtSa_sl_<version>.tar.gz` — the web application
- Distribution archives:
  - `OneClickExtSa_d_<version>.tar.gz` — the daemon
  - `OneClickExtSa_sl_<version>.tar.gz` — the web application
- A dedicated service user (e.g. `spectrum`) who will own the installation directory. If the user does not exist yet: `sudo useradd -r spectrum`

Java and Apache Tomcat are installed as part of Steps 1 and 2 below.

Most installation steps below are run as the `spectrum` user. Only the initial directory setup and the final systemd registration require root (`sudo`). Log in as `spectrum` or prefix each command with `sudo -u spectrum` as appropriate.

Copyright © 1998 - 2025 DICOS GmbH Kommunikationssysteme

Installation steps

0. SET UP THE BASE DIRECTORY

Windows Linux

Create `%DCSROOT%` (e.g. `C:\DcsSL`) manually before continuing.

Create `$DCSROOT` as root and assign ownership to the service user. Then switch to the service user for all remaining steps:

```
sudo mkdir -p /opt/dcssl
sudo chown spectrum:spectrum /opt/dcssl
```

Log in as the service user and set the variable for the current session:

```
su - spectrum
export DCSROOT=/opt/dcssl
```

Add `export DCSROOT=/opt/dcssl` to `~/.bash_profile` so the variable is available in future sessions.

1. INSTALL JAVA 17

Windows Linux

Install the JDK to `%DCSROOT%\Java`. If the JRE is installed instead of the full JDK, the directory must still be named `Java` so that the startup scripts and `jsl64.ini` can locate the runtime.

After installation, verify that `%DCSROOT%\Java\bin\java.exe` exists.

Install via your distribution's package manager:

```
# Debian / Ubuntu
sudo apt install openjdk-17-jdk

# RHEL / Rocky Linux
sudo dnf install java-17-openjdk-devel
```

Alternatively, download and extract the JDK manually to `$DCSROOT/Java` and set `JAVA_HOME` accordingly.

Verify the installation:

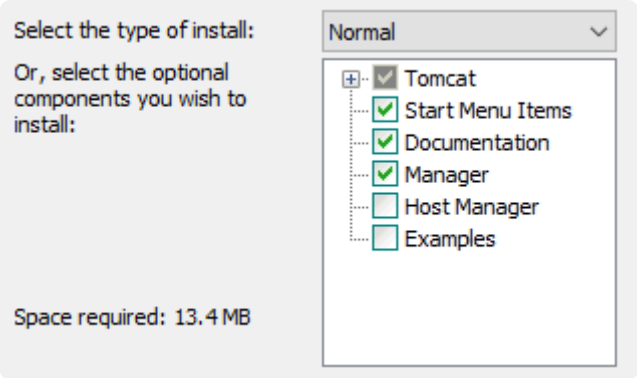
```
java -version
```

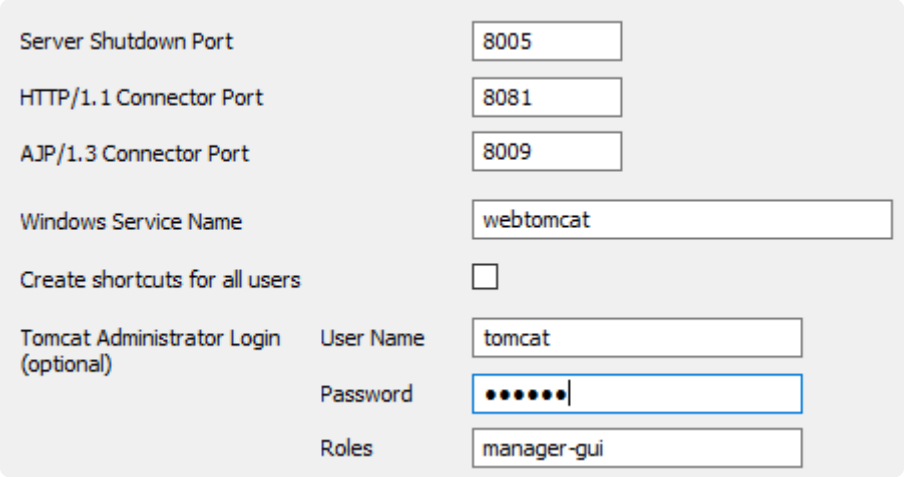

Copyright © 1998 - 2025 DICOS GmbH Kommunikationssysteme

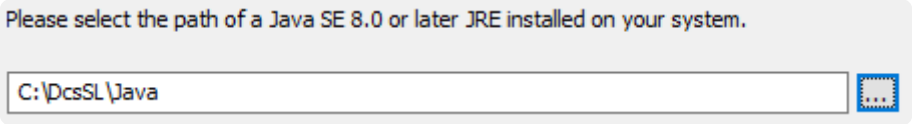
2. INSTALL TOMCAT

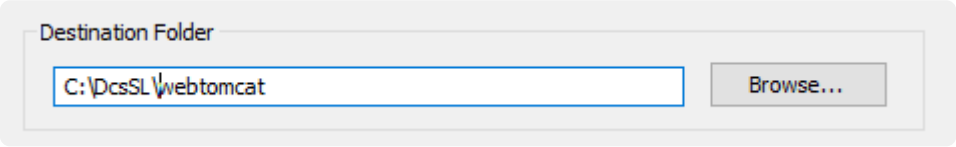
Windows Linux

Install Tomcat to `%DCSROOT%\webtomcat` with the Windows service name `webtomcat`.









Download Apache Tomcat 9 or later from tomcat.apache.org and extract it to the target directory:

```
wget https://downloads.apache.org/tomcat/tomcat-<major>/v<version>/bin/apache-tomcat-<version>.tar.gz

mkdir -p $DCSROOT/webtomcat
tar xzf apache-tomcat-<version>.tar.gz \
    -C $DCSROOT/webtomcat \
    --strip-components=1
chmod +x $DCSROOT/webtomcat/bin/*.sh
```

Replace `<major>` with the major version number (e.g. `10`) and `<version>` with the full version (e.g. `10.1.40`).

Verify the installation by checking that `startup.sh` and `shutdown.sh` are present in `$DCSROOT/webtomcat/bin/`.

Copyright © 1998 - 2025 DICOS GmbH Kommunikationssysteme

3. INSTALL THE ONECLICKEXT DAEMON AND WEB APPLICATION

Both components are delivered as tar archives.

Windows Linux

- Unpack `OneClickExtSa_d_<version>.tar.gz` to `%DCSROOT%\DICOS`.
- Unpack `OneClickExtSa_sl_<version>.tar.gz` to `%DCSROOT%\webtomcat\webapps`.

```
mkdir -p $DCSROOT/DICOS
tar xzf OneClickExtSa_d_<version>.tar.gz -C $DCSROOT/DICOS
chmod +x $DCSROOT/DICOS/OneClickExt/bin/*.sh
mkdir -p $DCSROOT/DICOS/OneClickExt/logs

tar xzf OneClickExtSa_sl_<version>.tar.gz -C $DCSROOT/webtomcat/webapps
```

The systemd service runs as the `spectrum` user by default (configurable via `installSystemdFilesUser.sh -u <username>`). Because `$DCSROOT` is already owned by `spectrum` (from Step 0), no ownership change is needed.

If Java was installed via the package manager rather than manually to `$DCSROOT/Java`, create a symlink so the service file can locate it:

```
mkdir -p $DCSROOT/Java
ln -s /usr/lib/jvm/jre-17-openjdk $DCSROOT/Java/jre
```

Note

When upgrading from a previous version, back up the existing files before unpacking.

Next steps

After installation, the properties file is placed as `$DCSROOT/DICOS/OneClickExt/bin/oneclickext.props_<version>`. Rename it to `oneclickext.props` before editing:

```
mv $DCSROOT/DICOS/OneClickExt/bin/oneclickext.props_<version> \
   $DCSROOT/DICOS/OneClickExt/bin/oneclickext.props
```

Then configure the daemon connection settings in `$DCSROOT/DICOS/OneClickExt/bin/oneclickext.props`:

```
oneclickext.mls.primary:  <Main Location Server>
oneclickext.mls.backup:   <Backup Location Server>
oneclickext.ss.host:      <SpectroSERVER hostname>
oneclickext.ss.user:      <Spectrum username>
oneclickext.rmi.port:     13689
oneclickext.rmi.connport: 13690
```

`rmi.port` is the RMI registry port where the daemon listens for client connections. `rmi.connport` is the second port Java RMI uses for object data transfer. Both ports must be open in any firewall between the Spectrum console host and the daemon host. The defaults (13689/13690) are fine unless they conflict with another service on your server.

Copyright © 1998 - 2025 DICOS GmbH Kommunikationssysteme

Then register the daemon as a system service:

Windows Linux

Copy the generated `jsl64.ini` file to the `bin` directory and install the service as administrator:

```
jsl64 -install
```

Check `jsl64.ini` for these settings beforehand (adjust paths if `%DCSROOT%` differs from the default `C:\DcsSL`):

```
wrkdir=C:/DcsSL/DICOS/OneClickExt/bin
jrepath=C:/DcsSL/Java/jre
jvmtype=server
```

Note

For `jvmtype`, check whether the `jvm.dll` inside your Java installation is under a `server` or `client` subdirectory and set the value accordingly.

Start the service:

```
net start oneclickextd
```

To uninstall before an upgrade:

```
jsl64 -remove
```

Run the systemd installer script as root from the `bin` directory:

```
sudo ./installSystemdFilesUser.sh
sudo systemctl enable oneclickext
sudo systemctl start oneclickext
```

Verify the service is running:

```
systemctl status oneclickext
```

The output should show `Active: active (running)`. If it shows `active (auto-restart)` or `failed`, check the log at `$DCSROOT/DICOS/OneClickExt/logs/oneclickext.log` or run `journalctl -u oneclickext` for the full output.

To uninstall:

```
sudo ./uninstallSystemdFilesUser.sh
```


Copyright © 1998 - 2025 DICOS GmbH Kommunikationssysteme

3.3.3 Uninstallation

Spectrum installation

Since Spectrum does not provide an automated uninstall mechanism, all installed files must be removed by hand. The steps differ slightly depending on the installed version.

STEPS

- Stop the OneClickExt daemon:
- If running as a systemd service: `sudo systemctl stop oneclickext`
- If running via idb tickets: `cmdC localhost 1 ONECLICKEXT` and `cmdC localhost 1 CHECKONECLICKEXT`
- Remove the OneClickExt section from `$SPECROOT/tomcat/webapps/spectrum/oneclick.jnlp` .
- Delete the menu entries for the script launcher in the file `$SPECROOT/custom/console/config/custom-menu-config.xml`
- Delete the following files/directories from `$SPECROOT` :

```
lib/SDPM/partslist/CHECKONECLICKEXT.idb
lib/SDPM/partslist/ONECLICKEXT.idb
Install-Tools/IDX/OC/DCS-OneClickExt.i
Install-Tools/MMD/DCS-OneClickExt.mmd
Install-Tools/CUS/oneclickext.cus
Install-Tools/WEB-XML/oneclickext-web.xml
Install-Tools/LOGS/VCD-DCS-OneCklickExt_*
tomcat/webapps/spectrum/lib/oneclickext.jar
tomcat/webapps/spectrum/oneclickext.jnlp
tomcat/webapps/spectrum/oneclickext.props
tomcat/webapps/spectrum/oneclicknetx.jnlp
tomcat/webapps/sl
DICOS/OneClickExt
```

Additional steps are necessary depending on the installed version:

< v2.5.8 >= v2.5.8

- Delete the servlet mappings for `/oneclickext.jnlp` , `/oneclicknetx.jnlp` , and `/scriptlauncher.jnlp` from `$SPECROOT/tomcat/webapps/spectrum/WEB-INF/web.xml`
- Delete the following additional files/directories from `$SPECROOT` :

```
tomcat/webapps/spectrum/dcslib
tomcat/webapps/spectrum/scriptlauncher.jnlp
tomcat/webapps/spectrum/WEB-INF/lib/oneclickext.jar
```

- Delete the servlet mappings for `/oneclickext.jnlp` and `/oneclicknetx.jnlp` from `$SPECROOT/tomcat/webapps/spectrum/WEB-INF/web.xml`
- If the Script Server was deployed to webtomcat, also delete from `$SPECROOT` :

```
webtomcat/webapps/sl
```


Copyright © 1998 - 2025 DICOS GmbH Kommunikationssysteme

Standalone installation

Windows Linux

Stop and remove the daemon service (run as administrator):

```
net stop oneclickextd
cd %DCSROOT%\DICOS\OneClickExt\bin
jsl64 -remove
```

Remove the installation files:

Delete the following directories:

```
%DCSROOT%\DICOS\OneClickExt
```

If the Script Server was deployed, also delete:

```
%DCSROOT%\webtomcat\webapps\sl
```

If Tomcat was installed solely for the Script Server and is no longer needed, uninstall the `webtomcat` Windows service via the Tomcat uninstaller or the Services control panel.

Stop and disable the systemd service (run as root):

```
sudo systemctl stop oneclickext
sudo systemctl disable oneclickext
```

Remove the systemd service files (run as root from the bin directory):

```
sudo $DCSROOT/DICOS/OneClickExt/bin/uninstallSystemdFilesUser.sh
```

Remove the installation files:

Delete the daemon directory:

```
rm -rf $DCSROOT/DICOS/OneClickExt
```

If the Script Server was deployed, also delete:

```
rm -rf $DCSROOT/webtomcat/webapps/sl
```

If Tomcat was installed solely for the Script Server and is no longer needed, stop it and delete `$DCSROOT/webtomcat`.

Copyright © 1998 - 2025 DICOS GmbH Kommunikationssysteme

3.4 Script launcher

3.4.1 Defining a Script

This page explains how to add a new script to OneClickExt so that it appears in the Spectrum console menu and can be launched by users. It covers choosing the correct launcher type, writing the properties definition, and adding the corresponding menu entry.

Adding a script involves two steps:

1. Defining the script in a properties file.
2. Adding a menu entry in the Spectrum console menu configuration XML.

Step 1 — Choose the launcher type

Three launcher classes exist, each for a different use case. You select one by specifying the class in the menu XML.

Class	Use when
`ScriptLauncher`	The script runs against exactly one device selected in the console
`MultiScriptLauncher`	The script runs against one or more devices, or against all members of a Global Collection
`GlobalScriptLauncher`	The script needs no device context at all (e.g. a report generator)

Step 2 — Define the script in properties

Scripts are defined in `oneclickext.props` or in any additional properties file listed under `oneclickext.additional.props`. Each script has a unique name (the `<scriptitem>` identifier) used throughout its configuration.

MINIMUM REQUIRED PROPERTIES

```
oneclickext.scripts.<scriptitem>.command: <path/to/script>
```

For a `MultiScriptLauncher` script, also add:

```
oneclickext.scripts.<scriptitem>.multidevice.active: true
```

USING PATH VARIABLES

To avoid hardcoding paths and extensions, define shared variables and reference them with `${...}`:

```
# Shared path settings (Windows example)
oneclickext.scripts.path: ..\scripts\
oneclickext.scripts.ext:  .bat
oneclickext.scripts.encoding: CP437

# Per-script references
oneclickext.scripts.RebootDevice.command:  ${oneclickext.scripts.path}RebootDevice${oneclickext.scripts.ext}
oneclickext.scripts.RebootDevice.encoding: ${oneclickext.scripts.encoding}
```

For Linux/Unix:

```
oneclickext.scripts.path: ../scripts/
oneclickext.scripts.ext:  .sh
oneclickext.scripts.encoding: UTF-8
```

EXECUTION SETTINGS

```
# Show script stdout in the result dialog (default: true)
oneclickext.scripts.<scriptitem>.showOutput: true

# Show exit code in the result dialog (default: true)
oneclickext.scripts.<scriptitem>.showExitCode: true
```


Copyright © 1998 - 2025 DICOS GmbH Kommunikationssysteme

```
# Run one instance at a time per device (AUTO), or allow parallel runs (NONE)
oneclickext.scripts.<scriptitem>.exec.class: AUTO

# Run all executions of this script sequentially, regardless of device (default: false)
oneclickext.scripts.<scriptitem>.exec.sequential: false

# Priority relative to other scripts (default: stdprio)
oneclickext.scripts.<scriptitem>.prio: stdprio
```

MULTISCRIPTLAUNCHER-SPECIFIC SETTINGS

```
oneclickext.scripts.<scriptitem>.multidevice.active: true

# Maximum number of devices (default: 10)
oneclickext.scripts.<scriptitem>.multidevice.maxdevices: 10

# Additional Spectrum attributes to fetch for each device (beyond Name and Network_Address)
# Attribute IDs in hex; these can then be used as parameter values via .attr
oneclickext.scripts.<scriptitem>.multidevice.attributes: 0x129e7
```

UI TEXTS

```
# Text shown above the parameter input fields
oneclickext.scripts.<scriptitem>.paramtext: Enter credentials for all devices:

# Confirmation question shown before execution (supports \n and %Pn% substitution)
oneclickext.scripts.<scriptitem>.secquestion: Are you sure you want to reboot?

# Label above the device list (MultiScriptLauncher only)
oneclickext.scripts.<scriptitem>.devicestext: Devices for RebootDevice:
```

Step 3 — Add a menu entry

Add an `<item>` element to the Spectrum console menu configuration file (`$SPECROOT/custom/console/config/custom-menu-config.xml`). The `name` attribute must match the `<scriptitem>` identifier used in the properties.

```
<menu name="Scripts">

    <!-- One or more devices -->
    <item name="RebootDevice">
        <popup-visibility>allways</popup-visibility>
        <action>
            <context>com.aprisma.spectrum.app.topo.client.render.ModelContext</context>
            <class>de.dicos.spectrum.oneclick.client.MultiScriptLauncher</class>
        </action>
    </item>

    <!-- Exactly one device -->
    <item name="SetModelName">
        <popup-visibility>allways</popup-visibility>
        <action>
            <context>com.aprisma.spectrum.app.topo.client.render.ModelContext</context>
            <class>de.dicos.spectrum.oneclick.client.ScriptLauncher</class>
        </action>
    </item>

    <!-- No device -->
    <item name="testscript">
        <popup-visibility>allways</popup-visibility>
        <action>
            <class>de.dicos.spectrum.oneclick.client.GlobalScriptLauncher</class>
        </action>
    </item>

</menu>
```

`ScriptLauncher` and `MultiScriptLauncher` require a `<context>` element pointing to `ModelContext` . `GlobalScriptLauncher` does not.

ROUTING TO AN ALTERNATIVE DAEMON

To route a specific script to a different OneClickExt daemon, append that daemon's hostname and RMI port to the item name:

```
<item name="SetModelName_myserver_13689">
```

The client parses the suffix and connects directly to the RMI daemon at `myserver:13689` , bypassing the primary (and secondary) daemon configured in the JNLP file. The script must still be defined in `oneclickext.props` using the base name (`SetModelName`), without the suffix.

Copyright © 1998 - 2025 DICOS GmbH Kommunikationssysteme

Complete example — RebootDevice

The following is the complete properties definition for the `RebootDevice` script from `conf/testscript.props`.

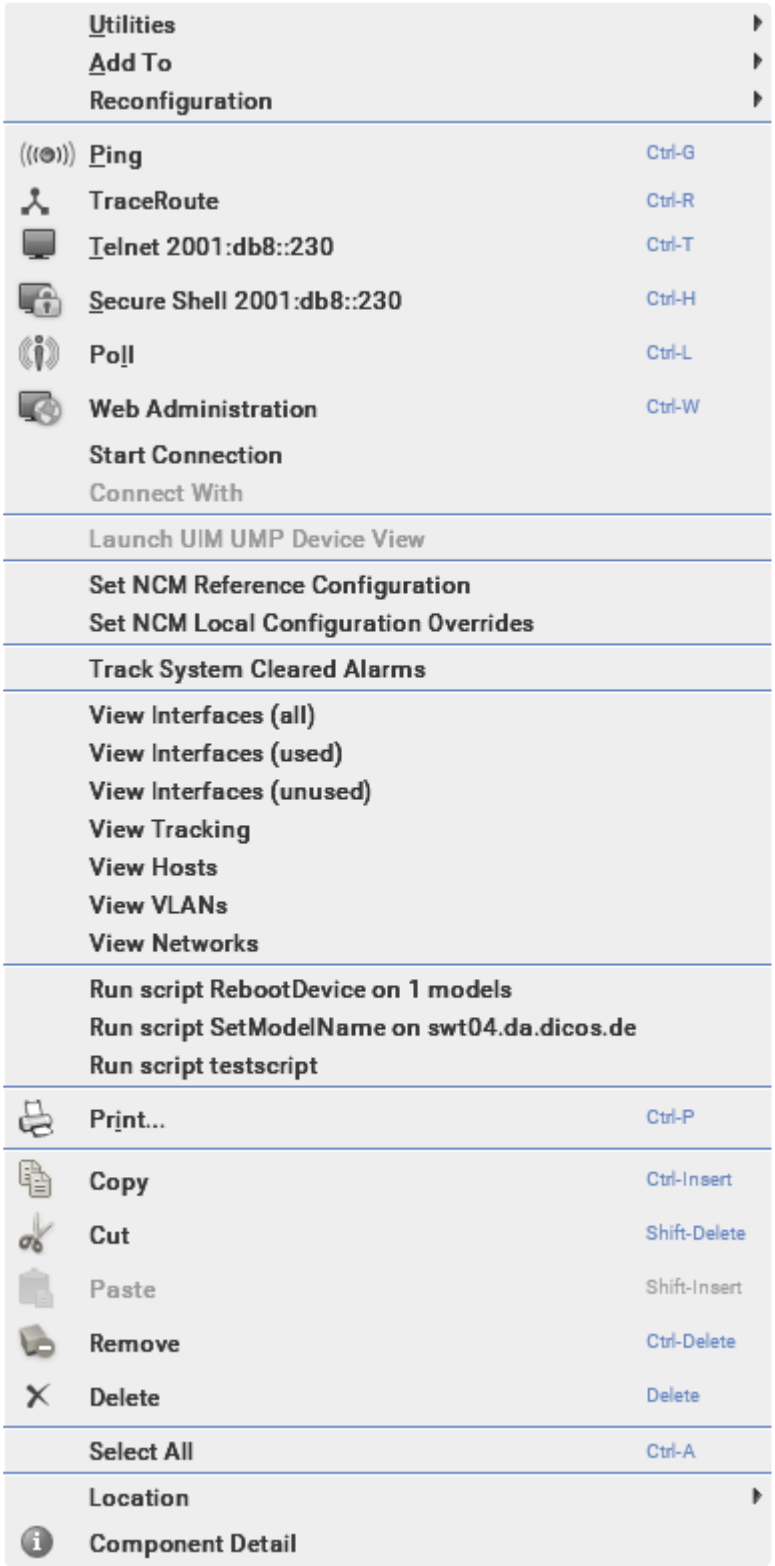
```
# Script command
oneclickext.scripts.RebootDevice.command:  ${oneclickext.scripts.path}RebootDevice${oneclickext.scripts.ext}
oneclickext.scripts.RebootDevice.encoding: ${oneclickext.scripts.encoding}

# Multi-device settings
oneclickext.scripts.RebootDevice.multidevice.active:      true
oneclickext.scripts.RebootDevice.multidevice.maxdevices:  10
oneclickext.scripts.RebootDevice.multidevice.attributes:  0x129e7

# Execution
oneclickext.scripts.RebootDevice.exec.sequential: false
oneclickext.scripts.RebootDevice.exec.class:      AUTO
oneclickext.scripts.RebootDevice.showOutput:      true
oneclickext.scripts.RebootDevice.showExitCode:    true

# UI texts
oneclickext.scripts.RebootDevice.devicestext:  Devices for RebootDevice:
oneclickext.scripts.RebootDevice.paramtext:    Enter credentials for all devices:
oneclickext.scripts.RebootDevice.secquestion: Are you sure you want to reboot?
oneclickext.scripts.RebootDevice.width:       230

# Parameters
oneclickext.scripts.RebootDevice.params: p1 p2 p3 p4 p5

oneclickext.scripts.RebootDevice.p1.name:   Network Address
oneclickext.scripts.RebootDevice.p1.attr:   0x12d7f
oneclickext.scripts.RebootDevice.p1.hidden: true

oneclickext.scripts.RebootDevice.p2.name:     User
oneclickext.scripts.RebootDevice.p2.value:    SpecAdmin
oneclickext.scripts.RebootDevice.p2.editable: false

oneclickext.scripts.RebootDevice.p3.name:    Password
oneclickext.scripts.RebootDevice.p3.visible: false

oneclickext.scripts.RebootDevice.p4.name:   Topology
oneclickext.scripts.RebootDevice.p4.attr:   0x129e7
oneclickext.scripts.RebootDevice.p4.hidden: true

oneclickext.scripts.RebootDevice.p5.name:   Name
oneclickext.scripts.RebootDevice.p5.attr:   0x1006e
oneclickext.scripts.RebootDevice.p5.hidden: true
```

For a detailed explanation of parameter options, see Parameters, Validators, and Actions.

What the user sees

Once the menu entry is in place and the daemon is running, the script appears in the right-click context menu of any device in the Spectrum topology view:

Copyright © 1998 - 2025 DICOS GmbH Kommunikationssysteme



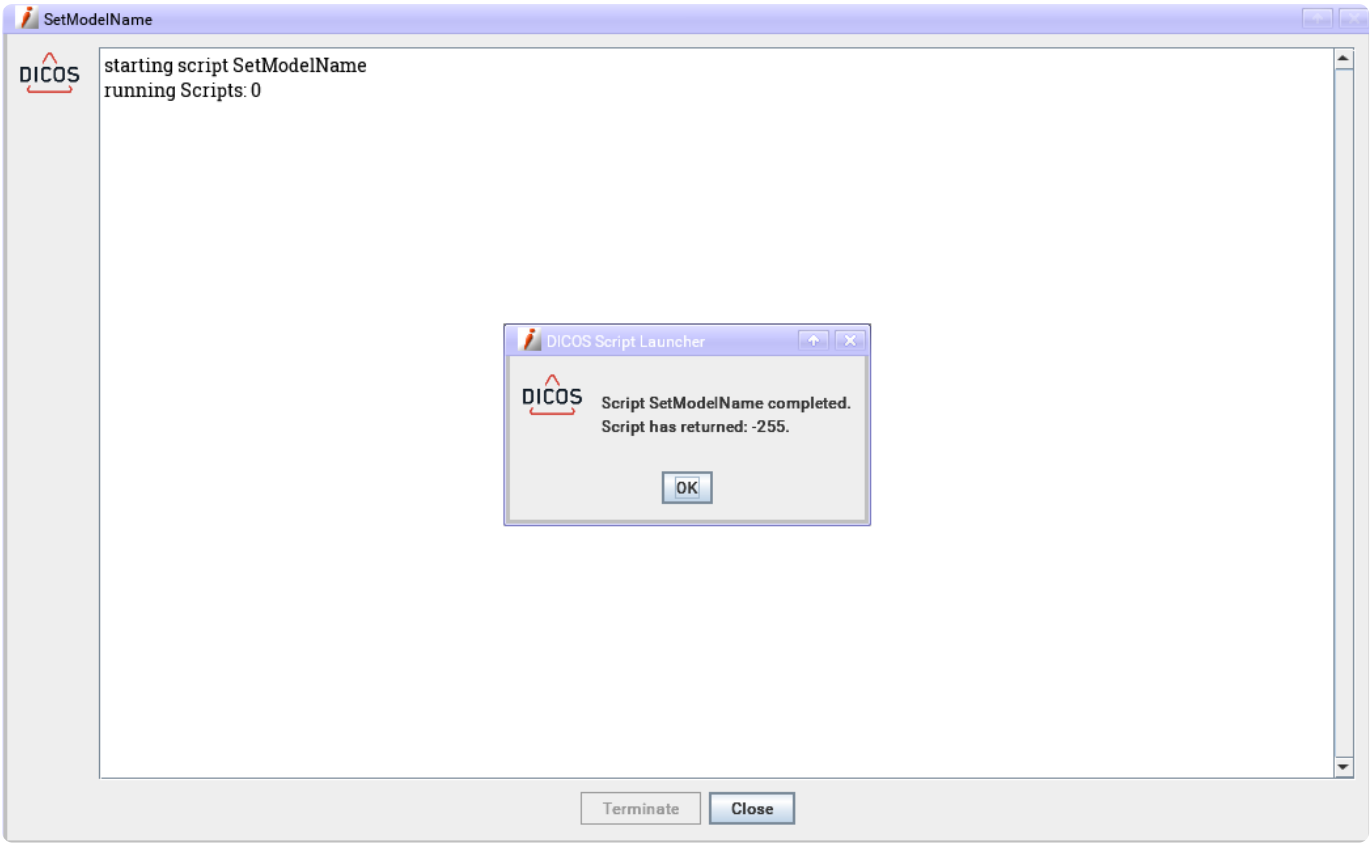
LAUNCH FLOW

Clicking a script entry triggers the following sequence:

1. Parameter dialog — the user fills in any input fields and clicks Run Script.
2. Confirmation (optional) — if `secquestion` is configured, a Yes/No dialog asks the user to confirm before execution proceeds. The dialog text is taken from the `secquestion` property.
3. Output window — the script runs and its output is streamed into a result window. A completion dialog reports the exit code when the script finishes.

Copyright © 1998 - 2025 DICOS GmbH Kommunikationssysteme

After clicking Run Script, the output window opens and the script runs:



See Exit code formatting for how to highlight the exit code in the result.

Copyright © 1998 - 2025 DICOS GmbH Kommunikationssysteme

3.4.2 Parameters, Validators, and Actions

This page explains how to configure the input parameters shown in the script launch dialog. It covers the available value sources for each parameter, display and visibility options, input validation via regular expressions, and combo box actions that dynamically control the dialog at runtime.

Parameters

Parameters define what the user sees and enters in the dialog before running a script. They are passed to the script in the order they are listed.

DECLARING PARAMETERS

First, list all parameter identifiers for the script. The order here determines the order in the dialog and the order in which values are passed to the script.

```
oneclickext.scripts.<scriptitem>.params: p1 p2 p3
```

The identifiers (`p1` , `p2` , etc.) are arbitrary; they are only used internally to look up the other properties.

Each parameter needs at least a display name:

```
oneclickext.scripts.<scriptitem>.p1.name: IP Address
```

VALUE SOURCES

A parameter's value can come from one of the following sources. Only one source should be set per parameter.

Static default value:

```
oneclickext.scripts.<scriptitem>.p1.value: SpecAdmin
```

Supports the `%DATE%` variable, which is replaced with the current date and time.

Spectrum attribute of the selected model:

```
oneclickext.scripts.<scriptitem>.p1.attr: 0x12d7f
```

The value is read from the specified attribute ID on the model selected in the console. If reading fails, the static `.value` is used as a fallback (unless `.emptyvalueonerror` is set).

Note

`devattr` (reading from the device of the selected model) is not supported for multi-device scripts.

Spectrum attribute of the device linked to the selected model:

```
oneclickext.scripts.<scriptitem>.p1.devattr: 0x12d7f
```

Reads the attribute from the device reached via the model's `Significant_Model_ID` (attribute `0x12a56`). Only for single-device scripts.

System parameter:

```
oneclickext.scripts.<scriptitem>.p1.sysparam: osuser
```


Copyright © 1998 - 2025 DICOS GmbH Kommunikationssysteme

Available values:

Value	Description
`osuser`	OS user name of the session
`specuser`	Spectrum user name
`starttime`	Formatted start time (format set by `oneclickext.scripts.starttime`)
`hostname`	Hostname of the client machine
`scriptip`	IP address of the current device (multi-device scripts only)
`scriptdns`	DNS name of the current device (multi-device scripts only)
`devlist`	Full device list file (multi-device scripts only)
`param1`, `param2`, …	Command-line parameters (standalone mode only)

URL parameter (for web-invoked scripts):

```
oneclickext.scripts.<scriptitem>.p1.urlparam: myParam
```

The value is taken from the URL query string: `&myParam=<value>`.

Combo box (dropdown):

```
oneclickext.scripts.<scriptitem>.p1.combo: Option A|Option B|Option C
```

Pipe-separated list of choices. See Combo boxes and actions below.

DISPLAY OPTIONS

```
# Hide the parameter row entirely (value is still passed to the script)
oneclickext.scripts.<scriptitem>.p1.hidden: false

# Show asterisks instead of the actual value (for passwords). Not applicable to combo boxes.
oneclickext.scripts.<scriptitem>.p1.visible: true

# Make the field read-only. Not applicable to combo boxes.
oneclickext.scripts.<scriptitem>.p1.editable: true

# Show empty value when an attribute read fails, instead of an error string
oneclickext.scripts.<scriptitem>.p1.emptyvalueonerror: false

# Tooltip text shown on hover
oneclickext.scripts.<scriptitem>.p1.tooltip: Enter the management IP address
```

INPUT FIELD WIDTH

```
# Width in pixels of all input fields in this script's dialog (default: 180)
oneclickext.scripts.<scriptitem>.width: 230
```

Validators

A validator checks user input against a regular expression before the script is allowed to run. If the input does not match, an error message is shown.

Validators are defined globally and then referenced per parameter:

```
# Define the validator
oneclickext.scripts.validator.<name>.regexp:  <regular expression>
oneclickext.scripts.validator.<name>.message: <error message shown to user>

# Reference from a parameter
oneclickext.scripts.<scriptitem>.p1.validator: <name>
```

EXAMPLE

Copyright © 1998 - 2025 DICOS GmbH Kommunikationssysteme

```
oneclickext.scripts.validator.valid_ip.regexp:  (25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])\.(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])\.(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])\.(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])
oneclickext.scripts.validator.valid_ip.message: Please enter a valid IP address.

oneclickext.scripts.validator.non_empty.regexp:  .*\S+.*
oneclickext.scripts.validator.non_empty.message: This field must not be empty.

oneclickext.scripts.MyScript.p1.validator: valid_ip
```

Combo boxes and actions

A parameter can be displayed as a dropdown (combo box). When the user selects an entry, optional combo actions can automatically update other parameters or enable/disable the Run button.

DEFINING A COMBO BOX

```
# Pipe-separated list of options
oneclickext.scripts.<scriptitem>.p1.combo: Option A|Option B|Option C

# Return the index (1, 2, 3, ...) instead of the text value (default: false)
oneclickext.scripts.<scriptitem>.p1.comboidx: false

# Pre-select the nth entry (1-based; default: first entry)
oneclickext.scripts.<scriptitem>.p1.combosel: 1
```

COMBO ACTIONS

Actions are triggered when a specific combo entry is selected. They are indexed by selection number (1 = first entry, 2 = second, etc.).

```
# List of action names to execute when entry n is selected
oneclickext.scripts.<scriptitem>.p1.comboaction.<n>.list: action1 action2

# Inline action definition (scoped to this parameter)
oneclickext.scripts.<scriptitem>.p1.comboaction.<actionname>.action: <action>
oneclickext.scripts.<scriptitem>.p1.comboaction.<actionname>.value:  <value>
oneclickext.scripts.<scriptitem>.p1.comboaction.<actionname>.param:  <Pn>
```

Actions can also be defined globally (shared across scripts) and referenced by name:

```
# Global action definition
oneclickext.scripts.comboaction.<name>.action: <action>
oneclickext.scripts.comboaction.<name>.value:  <value>
oneclickext.scripts.comboaction.<name>.param:  <Pn>
```

AVAILABLE ACTIONS

Action	Value	Parameter	Description
`enable`	—	`Pn`	Enable the input field of parameter n
`disable`	—	`Pn`	Disable the input field of parameter n
`clear`	—	`Pn`	Clear the value of parameter n
`text`	text to insert	`Pn`	Set the value of parameter n to the given text
`date`	date specifier	`Pn`	Set parameter n to a computed date range value
`rbenable`	—	—	Enable the Run button
`rbdisable`	—	—	Disable the Run button

Date specifiers for the `date` action: `lastDay.from`, `lastDay.to`, `lastWeek.from`, `lastWeek.to`, `lastMonth.from`, `lastMonth.to`.

DISABLING THE RUN BUTTON BY DEFAULT

To force a user to make a combo selection before the script can run, disable the Run button in the initial state and use `rbenable` to re-enable it:

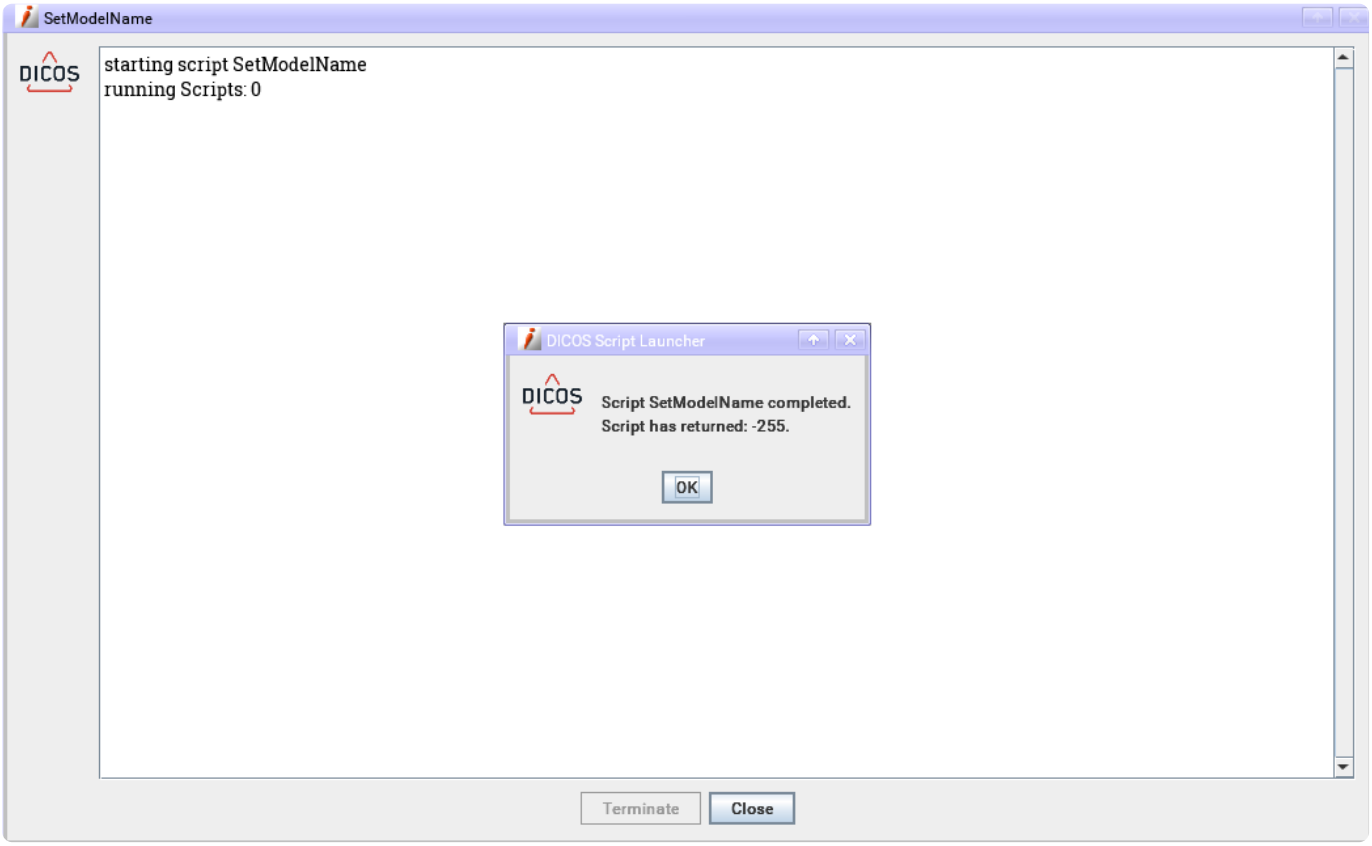
```
oneclickext.scripts.<scriptitem>.runbutton.enabled: false
```


Copyright © 1998 - 2025 DICOS GmbH Kommunikationssysteme

```
oneclickext.scripts.comboaction.en_rb.action: rbenable
oneclickext.scripts.comboaction.di_rb.action: rbdisable
```

Exit code formatting

After a script finishes, a completion dialog reports the exit code. The output window remains open so the user can review the full output:



Exit codes can be highlighted with HTML formatting in the output dialog. Define named format rules and list them in the order they should be applied:

```
oneclickext.scripts.exitcode.formats: ltZero zero gtZero

oneclickext.scripts.exitcode.format.zero.op:    eq
oneclickext.scripts.exitcode.format.zero.val:   0
oneclickext.scripts.exitcode.format.zero.start: <font size='5' color='green'>
oneclickext.scripts.exitcode.format.zero.end:   </font>

oneclickext.scripts.exitcode.format.gtZero.op:    gt
oneclickext.scripts.exitcode.format.gtZero.val:   0
oneclickext.scripts.exitcode.format.gtZero.start: <font size='5' color='red'>
oneclickext.scripts.exitcode.format.gtZero.end:   </font>

oneclickext.scripts.exitcode.format.ltZero.op:    lt
oneclickext.scripts.exitcode.format.ltZero.val:   0
oneclickext.scripts.exitcode.format.ltZero.start: <font size='5' color='blue'>
oneclickext.scripts.exitcode.format.ltZero.end:   </font>
```

Supported operators: `eq` (equal), `gt` (greater than), `lt` (less than).

The text block surrounding the output can also be formatted:

```
oneclickext.scripts.exitcode.format.text.start: <font size='5' face='verdana' color='black'>
oneclickext.scripts.exitcode.format.text.end:   </font>
```


Copyright © 1998 - 2025 DICOS GmbH Kommunikationssysteme

3.4.3 Permissions

This page describes the permission model used by OneClickExt to control which users may launch which scripts. Permissions apply to scripts invoked from the web interface (Script Server) and are evaluated server-side before execution. Scripts launched directly from the Spectrum console are not subject to this check.

How permissions work

Permissions are evaluated in two passes:

1. Direct script permission — the user is explicitly listed as allowed to run the script.
2. Group permission — the user belongs to a group, and the script is assigned to that group.

If either check passes, the script is permitted.

Defining groups

A group is a named collection of scripts:

```
oneclickext.scripts.groups.<groupname>: <script1> <script2> ...
```

Example:

```
oneclickext.scripts.groups.Reboot: RebootDevice RebootDevice2
```

Assigning permissions to users

VIA GROUP MEMBERSHIP

```
oneclickext.scripts.users.<username>.groups: <group1> <group2> ...
```

Example:

```
oneclickext.scripts.users.tomcat.groups:   Reboot
oneclickext.scripts.users.spectrum.groups: Reboot
```

VIA DIRECT SCRIPT ASSIGNMENT

```
oneclickext.scripts.users.<username>.scripts: <script1> <script2> ...
```

Use the special value `all` to grant access to every script:

```
oneclickext.scripts.users.sladmin.scripts: all
```

Both types can be combined for the same user:

```
oneclickext.scripts.users.spectrum.groups:  Reboot
oneclickext.scripts.users.spectrum.scripts: SetModelName testscript
```

Complete example

```
# Groups
oneclickext.scripts.groups.Reboot: RebootDevice RebootDevice2

# Users via group
oneclickext.scripts.users.tomcat.groups:   Reboot
oneclickext.scripts.users.sladmin.groups:  Reboot
oneclickext.scripts.users.spectrum.groups: Reboot

# Users via direct script assignment
oneclickext.scripts.users.sladmin.scripts:  SetModelName testscript
oneclickext.scripts.users.spectrum.scripts: SetModelName testscript
```


Copyright © 1998 - 2025 DICOS GmbH Kommunikationssysteme

In this example: - `tomcat` , `sladmin` , and `spectrum` may all run `RebootDevice` and `RebootDevice2` (via the `Reboot` group). - `sladmin` and `spectrum` may additionally run `SetModelName` and `testscript` directly.

Copyright © 1998 - 2025 DICOS GmbH Kommunikationssysteme

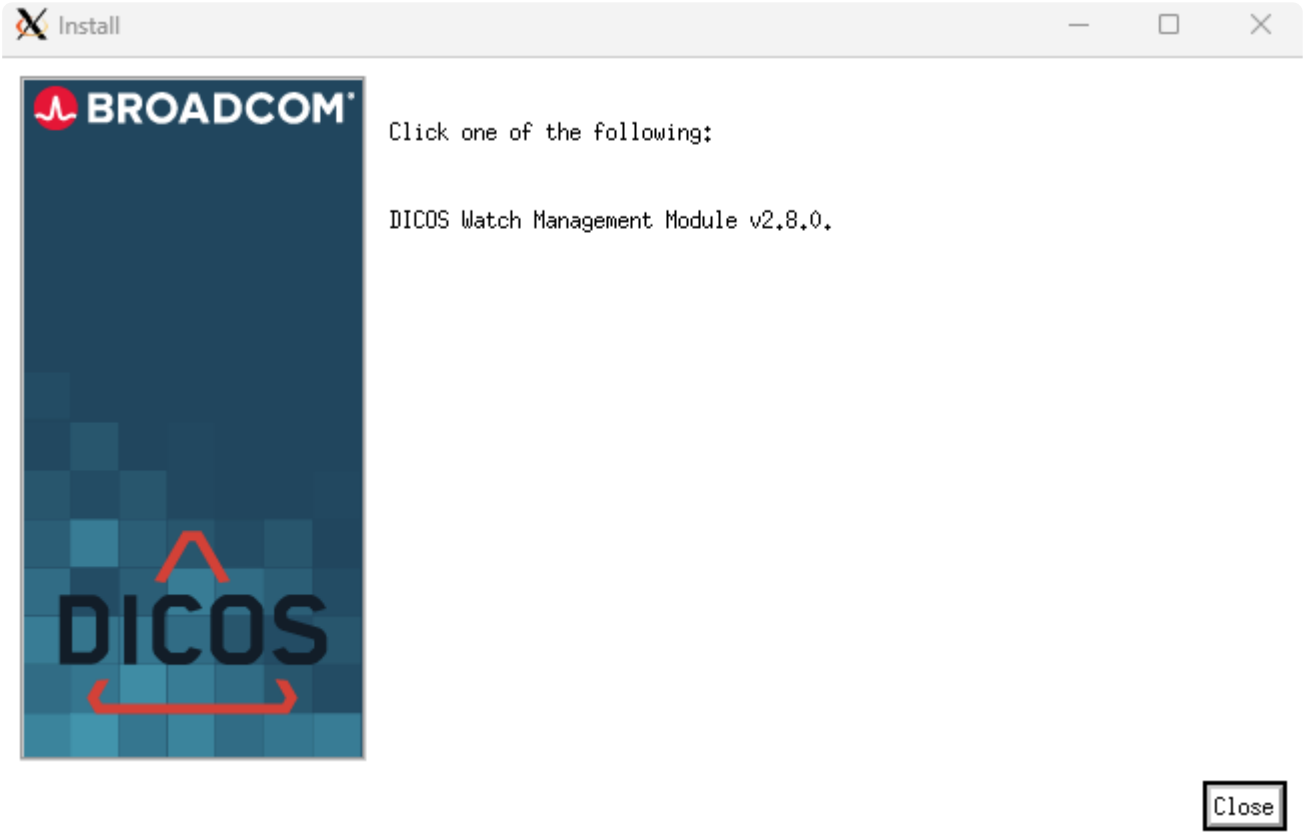
3.5 Watch management

3.5.1 DCSWatch Integration

DCSWatch is a separate DICOS product for managing watches on Spectrum models. It uses OneClickExt as its platform. The OneClickExt daemon and client must be running for DCSWatch to function, but DCSWatch is installed and licensed independently.

Installation

DCSWatch has its own installer, separate from the OneClickExt installation:

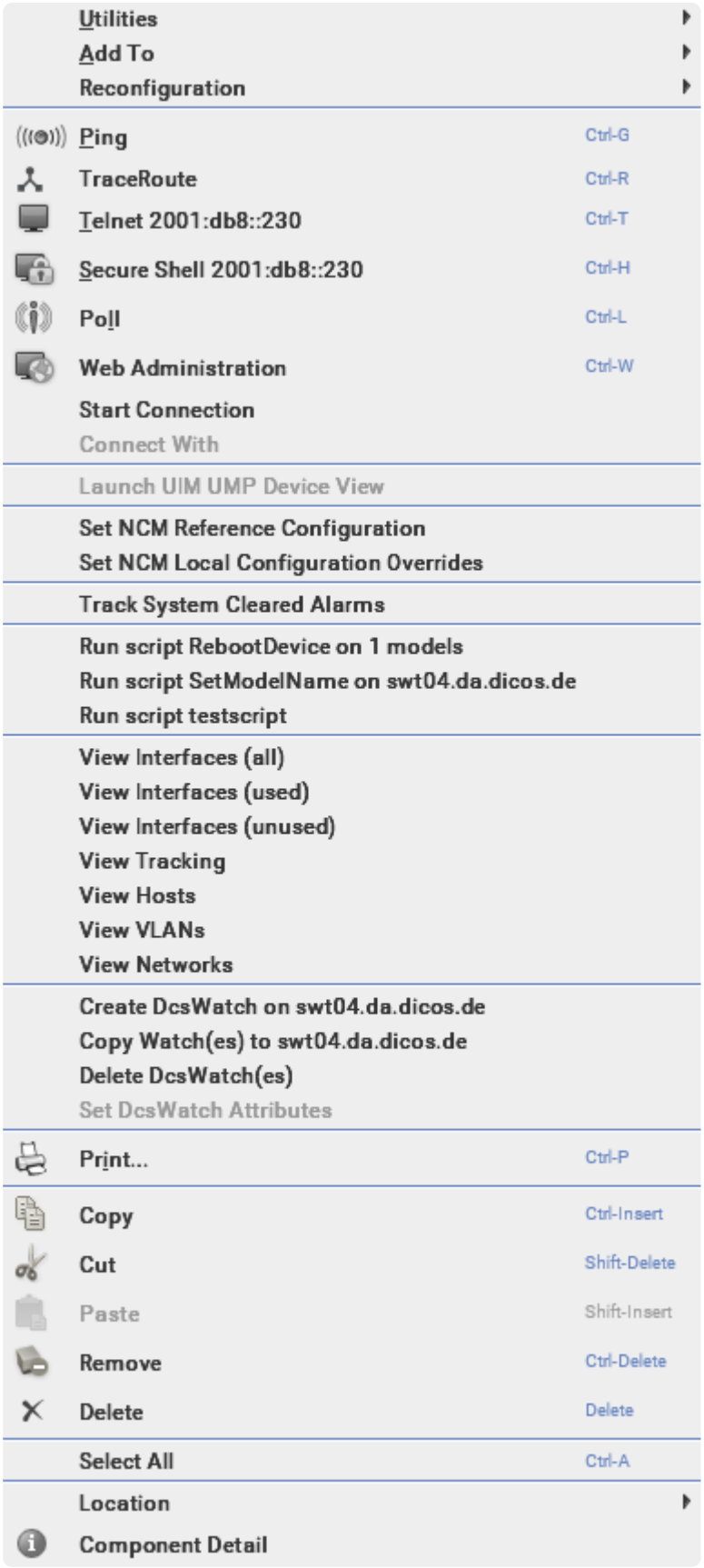


Run the DCSWatch installer on the Spectrum server after OneClickExt is already installed and running. The installer registers the DCSWatch model types with Spectrum and configures the integration.

Menu integration

Once DCSWatch is installed, its menu entries appear automatically in the Spectrum console. No manual changes to `custom-menu-config.xml` are required:

Copyright © 1998 - 2025 DICOS GmbH Kommunikationssysteme



Copyright © 1998 - 2025 DICOS GmbH Kommunikationssysteme

The four DCSWatch actions are:

Entry	Description
Create DcsWatch	Create a new watch on the selected device
Copy Watch(es)	Copy existing watches to the selected device
Delete DcsWatch(es)	Delete watches from the selected device
Set DcsWatch Attributes	Configure the attributes monitored by an existing watch (only active when a watch exists on the device)

Further documentation

For full usage documentation on configuring watches, attribute monitoring, and reporting, refer to the DCSWatch product documentation.

Copyright © 1998 - 2025 DICOS GmbH Kommunikationssysteme

4. Reference

4.1 OneClickExt properties reference

This page is the complete properties reference for OneClickExt. For task-oriented guidance, see the Script Launcher guides.

4.1.1 SpectroSERVER configuration

The daemon connects to Spectrum via CORBA. These properties identify which Spectrum servers to connect to and control reconnect behavior.

Main Location Server and Backup Main Location Server

`oneclickext.mls.primary: <MLS host>`

`oneclickext.mls.backup: <BLS host>`

Hostname of the MOM (Main SpectroSERVER)

`oneclickext.ss.host: <MLS host>`

Spectrum user account used by the daemon for all CORBA operations (attribute reads, model queries).

`oneclickext.ss.user: <spectrum user>`

Set to `true` when domain names in Spectrum are registered as FQDNs (e.g., via `MapUpdate -v`).

`dcsspcexp.ss.domain.fqdn: true | false`

If the CORBA connection to the MOM fails at startup, the daemon retries at this interval. `retryCount: 0` means unlimited retries.

Retry interval [ms]:

`oneclickext.ss.retryIntervall: 1000`

Maximum retry count (0 = unlimited):

`oneclickext.ss.retryCount: 300`

When `true`, the daemon waits until all configured Spectrum domains are reachable before becoming operational. When `false` (default), it starts with whichever domains are available.

`oneclickext.ss.connectAll: false`

How often the CORBA monitor polls the MOM to detect connection loss [ms].

`oneclickext.ss.pollingIntervall: 1000`

How often the daemon queries Spectrum for the current domain list [ms].

`oneclickext.ss.abfrageIntervall: 10000`

When `true`, each CORBA monitor uses its own timer thread rather than sharing one.

`oneclickext.ss.owntimer: true`

Whitelist/blacklist of Spectrum domains to query. `include.domains.list` takes precedence: if set, only those domains are queried regardless of `exclude.domains.list`.

`oneclickext.ss.include.domains.list: <domainlist>`

`oneclickext.ss.exclude.domains.list: <domainlist>`

4.1.2 RMI Parameters

The daemon exposes its interface over Java RMI. Two ports are required: one for the RMI registry and one for the actual object communication.

Copyright © 1998 - 2025 DICOS GmbH Kommunikationssysteme

Daemon listener port (RMI registry) and object communication port:

`oneclickext.rmi.port: <RMI port>`

`oneclickext.rmi.connport: <RMI connection port>`

The client reads these from system properties set in the JNLP launch file. The installer sets them automatically in `oneclickext.jnlp`:

`oneclickext.rmi.host: <host>`

`oneclickext.rmi.port: <port>`

Both RMI ports must be open in any firewall between the console host and the daemon host.

In a fault-tolerant environment, a secondary host can also be configured. The client tries the primary first and falls back to the secondary if the primary is unreachable:

`oneclickext.rmi.secondary.host: <host>`

`oneclickext.rmi.secondary.port: <port>`

4.1.3 Settings for script execution

Maximum number of parallel execution threads. Scripts run on the parallel pool unless `exec.sequential` is set, in which case they use a single sequential thread.

`oneclickext.exec.threads: 32`

Diagnostic: log queue and thread status at this interval [ms]. `0` disables the log (default).

`oneclickext.exec.info.interval: 0`

4.1.4 Additional Properties

The daemon hot-reloads these files before each script call. Paths are relative to the daemon's working directory (typically `bin/`). Multiple files are separated by spaces.

`oneclickext.additional.props: ../conf/testscript.props`

4.1.5 ScriptLauncher classes

There are three ScriptLauncher classes:

- `ScriptLauncher`: can only be launched on one model
- `MultiScriptLauncher`: launches on one or more models — on a Global Collection or on selected models in Spectrum
- `GlobalScriptLauncher`: launches without models

When starting scripts from OneClick, an alternative script server can be specified. To do this, the script item is extended to include the host and port:

`<item name="SetModelName">`

becomes

`<item name="SetModelName_<hostname>_<port>">`

4.1.6 Permissions for launching the ScriptLauncher via a website

Permissions control which users can invoke scripts through the HTTP interface. They are evaluated on the server side; the username passed in the URL request is checked against this configuration.

A permission can be granted via a group or directly to a user.

Script groups — assignment of scripts to a group:

`oneclickext.scripts.groups.<group name>: <script 1> <script 2> ...`

Group permissions — assignment of a group to a user:

`oneclickext.scripts.users.<user name>.groups: <group name>`

Copyright © 1998 - 2025 DICOS GmbH Kommunikationssysteme

Script permissions — assignment of scripts directly to a user:
`oneclickext.scripts.users.<user name>.scripts: <script 1> <script 2> ...`

4.1.7 Configuration for OneClick script runner

Script command

The `command` property specifies the executable (or full path) that the daemon runs. The `encoding` property controls how the process output stream is decoded; a mismatch causes garbled output.

`oneclickext.scripts.<scriptitem>.command: <Windows or UNIX Command>`
`oneclickext.scripts.<scriptitem>.encoding: <CP437 on Windows | UTF-8 on UNIX>`

Script execution

By default, all scripts run in parallel, subject to the thread pool limit. Setting `exec.sequential: true` routes the script through a single dedicated thread, so it never runs concurrently with other sequential scripts.

`oneclickext.scripts.<scriptitem>.exec.sequential: <true|false>`
Default: `false`.

`exec.class` controls per-device concurrency — whether a second invocation for the same device is held in a wait queue until the first finishes.

Possible values:

- `AUTO` — uses the Spectrum model handle as the device key when launched from OneClick; uses the IP address as the key when launched from the command line. This is the default and prevents two scripts from running simultaneously on the same device.
- `NONE` — no per-device restriction; multiple scripts may run on the same device at the same time.

`oneclickext.scripts.<scriptitem>.exec.class: <NONE|AUTO>`
Default: `AUTO`.

Note

Older configuration examples may show `exec.type` instead of `exec.class`. The server reads `exec.class`; `exec.type` has no effect.

Scripts are dispatched in priority order. Within the same priority level, scripts are processed in the order they were submitted (FIFO). Internal numeric values: `highprio` = 1, `stdprio` = 2, `lowprio` = 3.

`oneclickext.scripts.<scriptitem>.prio: <highprio|stdprio|lowprio>`
Default: `stdprio`.

Diagnostic: when `true`, the script output window includes a line showing the queue status and thread count at dispatch time.
`oneclickext.scripts.<scriptitem>.runinfo: false`

The Run button initial state. When set to `false`, the button starts disabled and must be re-enabled by a combo action before the user can run the script. Read each time the dialog is opened.
`oneclickext.scripts.<scriptitem>.runbutton.enabled: false`

Control the output and exit code of the script

`oneclickext.scripts.<scriptitem>.showOutput: <true|false>`
Default: `true`.

Copyright © 1998 - 2025 DICOS GmbH Kommunikationssysteme

`oneclickext.scripts.<scriptitem>.showExitCode: <true|false>`
Default: `true`.

Configuration for multiscripts

Scripts launched via `MultiScriptLauncher` must have `multidevice.active: true`. The dialog shows the device list and runs the script once per device.

`oneclickext.scripts.<scriptitem>.multidevice.active: <true|false>`
Default: `false`.

Maximum number of devices that can be selected for a single run.
`oneclickext.scripts.<scriptitem>.multidevice.maxdevices: <n>`
Default: `10`

The attributes `Model_Name (0x1006e)` and `Network_Address (0x12d7f)` are always fetched for each device. List any additional attributes here; they are fetched and made available as parameter values. Only attributes listed here (or the two mandatory ones) can be referenced by script parameters using `attr`.

`oneclickext.scripts.<scriptitem>.multidevice.attributes: <Attr ID> ...`

Definition of texts

Both texts support `\n` for line breaks. `%Pn%` is replaced at display time with the value the user has entered in parameter field n.

`paramtext` is displayed above the input fields and should describe the script's purpose.
`oneclickext.scripts.<scriptitem>.paramtext: <parameter text including \n and %Pn%>`

`secquestion` is shown in a confirmation dialog after the user clicks Run, giving them a last chance to review or cancel. The `%Pn%` substitution lets you echo back the values the user entered.
`oneclickext.scripts.<scriptitem>.secquestion: <Question text including \n and %Pn%>`

Web invocation

These properties apply when the script is invoked through the Script Server HTTP interface.

Width (in characters) of the HTML input fields in the web dialog:
`oneclickext.scripts.<scriptitem>.input.size: 35`

Help text displayed in the web dialog:
`oneclickext.scripts.<scriptitem>.helptext: <text>`

4.1.8 Parameter definition

Parameter list — all parameter identifiers, in order. The order determines both the dialog layout and the order arguments are passed to the script.
`oneclickext.scripts.<scriptitem>.params: <p1 p2 p3 ... >`

Width in pixels of all input fields in this script's dialog.
`oneclickext.scripts.<scriptitem>.width: <pixels>`
Default: 180

Label shown to the left of the input field.
`oneclickext.scripts.<scriptitem>.p<n>.name: <Parameter Name>`

Copyright © 1998 - 2025 DICOS GmbH Kommunikationssysteme

Value definition

- Static default value. `%DATE%` is expanded to the current date and time.
  `oneclickext.scripts.<scriptitem>.p<n>.value: <any text>`
- Validate against a named validator (defined globally, see below).
  `oneclickext.scripts.<scriptitem>.p<n>.validator: <validator name>`
- Read from a Spectrum attribute on the selected model. On read error, falls back to `.value` (or empty if `emptyvalueonerror: true`).
  `oneclickext.scripts.<scriptitem>.p<n>.attr: <attrId to read on selected model>`
- Read from the device model associated with the selected model, reached via `Significant_Model_ID (0x12a56)`. Not supported for multi-device scripts.
  `oneclickext.scripts.<scriptitem>.p<n>.devattr: <attrId to read on device of selected model>`
- Use a system-provided value:
  `oneclickext.scripts.<scriptitem>.p<n>.sysparam: <name>`

Value	Scope	Description
`osuser`	all	OS user name of the current session
`specuser`	all	Spectrum user name
`hostname`	all	Hostname of the client machine
`starttime`	all	Formatted timestamp captured when the dialog opens (format set by `oneclickext.scripts.starttime`)
`scriptip`	multi-device only	IP address of the device currently being processed
`scriptdns`	multi-device only	DNS name of the device currently being processed
`devlist`	multi-device only	Path to the full device list file
`param1`, `param2`, …	command-line / Script Server only	Command-line arguments passed to the script

- Use a URL query parameter (Script Server only), passed as `&<name>=<value>` in the invocation URL:
  `oneclickext.scripts.<scriptitem>.p<n>.urlparam: <name>`
- Display a dropdown (combo box). Values are pipe-separated.
  `oneclickext.scripts.<scriptitem>.p<n>.combo: <text 1|text 2|text 3| ...>`

Return the 1-based index of the selected entry instead of the text:
`oneclickext.scripts.<scriptitem>.p<n>.comboidx: <true|false>`
Default: `false`

Pre-select the nth entry when the dialog opens (1-based):
`oneclickext.scripts.<scriptitem>.p<n>.combosel: <n>`

Combo actions are triggered when the user selects an entry. They can set values, enable/disable fields, or enable/disable the Run button.
`oneclickext.scripts.<scriptitem>.p<n>.comboaction.<n (selection)>.list: <action 1> [<action 2> ...]`
`oneclickext.scripts.<scriptitem>.p<n>.comboaction.<action name>.action: <action>`
`oneclickext.scripts.<scriptitem>.p<n>.comboaction.<action name>.value: <value>`
`oneclickext.scripts.<scriptitem>.p<n>.comboaction.<action name>.param: <param>`
Action definitions and available values are described under Global Actions below.

Copyright © 1998 - 2025 DICOS GmbH Kommunikationssysteme

Parameter options

- Show asterisks instead of the actual value (for passwords). Does not apply to combo boxes.
  `oneclickext.scripts.<scriptitem>.p<n>.visible: <true|false>`
  Default: `true`
- Make the field read-only. Does not apply to combo boxes.
  `oneclickext.scripts.<scriptitem>.p<n>.editable: <true|false>`
  Default: `true`
- Hide the parameter row entirely. The value is still passed to the script.
  `oneclickext.scripts.<scriptitem>.p<n>.hidden: <true|false>`
  Default: `false`
- Show an empty field on attribute read error instead of an error string.
  `oneclickext.scripts.<scriptitem>.p<n>.emptyvalueonerror: <true|false>`
  Default: `false`
- Tooltip text shown when the user hovers over the input field. Also rendered as an HTML `title` attribute in the web dialog.
  `oneclickext.scripts.<scriptitem>.p<n>.tooltip: <text>`

4.1.9 Syslog settings

The Script Server can send RFC 3164 syslog messages when a script starts and when it completes. The message format template is configured on the Script Server (see the Script Server reference).

These per-script properties control whether syslog is sent and what priority value to include in the `%PRI%` template token:

`oneclickext.scripts.<scriptitem>.startsyslog.enabled: <true|false>`
Default: `false`
`oneclickext.scripts.<scriptitem>.endsyslog.enabled: <true|false>`
Default: `false`

`oneclickext.scripts.<scriptitem>.startsyslog.pri: <pri>`
Default: `190` (facility 23 = local7, severity 6 = informational)
`oneclickext.scripts.<scriptitem>.endsyslog.pri: <pri>`
Default: `191` (facility 23 = local7, severity 7 = debug)

The `pri` value is encoded as `(facility × 8) + severity`, matching RFC 3164. Common choices: `190` for informational, `187` for warning (severity 3), `184` for emergency (severity 0).

4.1.10 Global Script settings

Validators

Validators are defined once and referenced by name from any parameter. If the input does not match the regular expression, the dialog shows the error message and prevents the script from running.

`oneclickext.scripts.validator.<validator name>.regexp: <Regular Expression>`
`oneclickext.scripts.validator.<validator name>.message: <Text for error message>`

Example:
`oneclickext.scripts.validator.valid_ip.regexp: \\d+\\.\\d+\\.\\d+\\.\\d+`

Copyright © 1998 - 2025 DICOS GmbH Kommunikationssysteme

Global Actions

Global combo actions are defined once and can be referenced by name from any script parameter. They work identically to inline actions but are shared across scripts.

`oneclickext.scripts.comboaction.<gaction name>.action: <action>`

`oneclickext.scripts.comboaction.<gaction name>.value: <value>`

`oneclickext.scripts.comboaction.<gaction name>.param: <param>`

Reference from a parameter:

`oneclickext.scripts.<scriptitem>.p<n>.comboaction.<n (selection)>.list: <gaction 1> [<gaction 2> ...]`

Action values and params:

Action	Value	Parameter	Description
enable	n.a.	Pn	enable a parameter
disable	n.a.	Pn	disable a parameter
text	'text'	Pn	insert text
clear	n.a.	Pn	clear value
date	date specifier	Pn	set input to Date
rbenable	n.a.	n.a.	enable Run-Button
rbdisable	n.a.	n.a.	disable Run-Button

date specifier

- `lastDay.from` and `lastDay.to`
- `lastWeek.from` and `lastWeek.to`
- `lastMonth.from` and `lastMonth.to`

4.1.11 Date formatting

The `starttime` sysparam value is formatted using this pattern, captured at the moment the script dialog is opened.

Copyright © 1998 - 2025 DICOS GmbH Kommunikationssysteme

`oneclickext.scripts.starttime: <format>`

Letter	Date or Time Component	Presentation	Examples
G	Era designator	Text	AD
y	Year	Year	1996; 96
M	Month in year	Month	July; Jul; 07
w	Week in year	Number	27
W	Week in month	Number	2
D	Day in year	Number	189
d	Day in month	Number	10
F	Day of week in month	Number	2
E	Day in week	Text	Tuesday; Tue
a	Am/pm marker	Text	PM
H	Hour in day (0-23)	Number	0
k	Hour in day (1-24)	Number	24
K	Hour in am/pm (0-11)	Number	0
h	Hour in am/pm (1-12)	Number	12
m	Minute in hour	Number	30
s	Second in minute	Number	55
S	Millisecond	Number	978
z	Time zone	General time zone	Pacific Standard Time; PST; GMT-08:00
Z	Time zone	RFC 822 time zone	-0800

Examples

Date and Time Pattern	Result
`yyyy.MM.dd G 'at' HH:mm:ss z`	`2001.07.04 AD at 12:08:56 PDT`
`EEE, MMM d, ''yy`	`Wed, Jul 4, '01`
`hh 'o''clock' a, zzzz`	`12 o'clock PM, Pacific Daylight Time`
`K:mm a, z`	`0:08 PM, PDT`
`yyyyy.MMMMM.dd GGG hh:mm aaa`	`02001.July.04 AD 12:08 PM`
`EEE, d MMM yyyy HH:mm:ss Z`	`Wed, 4 Jul 2001 12:08:56 -0700`
`yyMMddHHmmssZ`	`010704120856-0700`
`yyyy-MM-dd'T'HH:mm:ss.SSSZ`	`2001-07-04T12:08:56.235-0700`

Copyright © 1998 - 2025 DICOS GmbH Kommunikationssysteme

4.2 Script Server Reference

This page describes the Script Server servlet: its HTTP interface for invoking OneClickExt scripts from a browser or external system, and its configuration properties.

4.2.1 Invoking scripts via HTTP

Scripts are invoked by sending an HTTP request to the servlet:

```
http://<host>:<port>/sl/server?scriptname=<name>[&<parameter>=<value>]...
```

URL parameters

Parameter	Description	Required
`scriptname`	Script identifier, as defined in the properties file	Yes
`username`	User account for permission checks (defaults to the OS user)	No
`iplist`	Path to a device list file on the server — one IP or DNS name per line, `#` for comments	No
`eiplist`	Same as `iplist`, but opens the file in an editor window first so the user can modify it	No
`iplist=input`	Opens an editor window for the user to enter the device list manually	No
`<name>=<value>`	Any custom parameter, referenced in the script config via `.urlparam: <name>`	No

Output history

If output history is enabled, previous script runs can be reviewed at:

```
http://<host>:<port>/sl/runner/history
```

4.3 Configuration properties

4.3.1 Connection to scriptserver daemon

```
sl.rmi.port: <port>
sl.rmi.host: <localhost or hostname>
```

4.3.2 Syslog settings

The script server can send a syslog when starting and after finishing a script. A template for the message can be specified. Multiple destinations are supported.

Syslog can be enabled per syslog, configured with the script.

Syslog destination and source

```
sl.syslog.destinations: <host:port> ...
sl.syslog.source: <hostname>
```


Copyright © 1998 - 2025 DICOS GmbH Kommunikationssysteme

Syslog Format Template

Syslogs are sent according to RFC 3164. The template supports the following variables:

Name	Description
`%PRI%`	Script specific, defined with the script
`%TS%`	start/stop time
`%SOURCE%`	syslog source hostname (`sl.syslog.source`)
`%SCRIPT%`	script name
`%USER%`	authenticated user
`%DEVCOUNT%`	number of devices

Format Examples:

```
sl.syslog.format.start: <%PRI%>%TS% %SOURCE% started script %SCRIPT% by user %USER% with %DEVCOUNT% devices
sl.syslog.format.end: <%PRI%>%TS% %SOURCE% completed script %SCRIPT% started by user %USER% with %DEVCOUNT% devices
```

Output history

The output of the scripts can be viewed later in the output history.

- enable history

```
sl.output.history.enable: true
```

- keep one day or until tomcat restart

```
sl.output.history.clean: 86400
```


Copyright © 1998 - 2025 DICOS GmbH Kommunikationssysteme

4.4 URLs and Ports

This page lists the URLs and default ports used by OneClickExt components.

4.4.1 Default ports

Component	Default port
Spectrum OneClick (HTTP, Linux)	8080
Spectrum OneClick (HTTP, Windows)	8080
Script Server — standalone	8080 (default, configurable in `$DCSROOT/webtomcat/conf/server.xml`)
OneClickExt daemon — RMI	13689
OneClickExt daemon — RMI connection	13690

Both RMI ports must be open in any firewall between the Spectrum console host and the daemon host.

4.4.2 Script Server

Description	URL
Start page	`http://<host>:<port>/sl/`
Script invocation	`http://<host>:<port>/sl/server?scriptname=<name>[&<param>=<value>]...`
Output history	`http://<host>:<port>/sl/runner/history`

For a full description of the invocation URL and its parameters, see the Script Server Reference.

4.4.3 Spectrum OneClick

Description	URL
OneClick website	`http://<host>:8080/spectrum`
OneClick web application	`http://<host>:8080/spectrum/webapp`
Script Server (Spectrum-integrated)	`http://<host>:<port>/sl/` — port as configured in `$SPECROOT/webtomcat/conf/server.xml`

Copyright © 1998 - 2025 DICOS GmbH Kommunikationssysteme

5. Tutorials

5.1 Running Your First Script

This tutorial walks you through running a OneClickExt script from the Spectrum console for the first time. By the end you will have launched a script against a device, confirmed the action, and reviewed the output, covering the complete end-to-end flow a user experiences every day.

5.1.1 Prerequisites

Before starting, make sure:

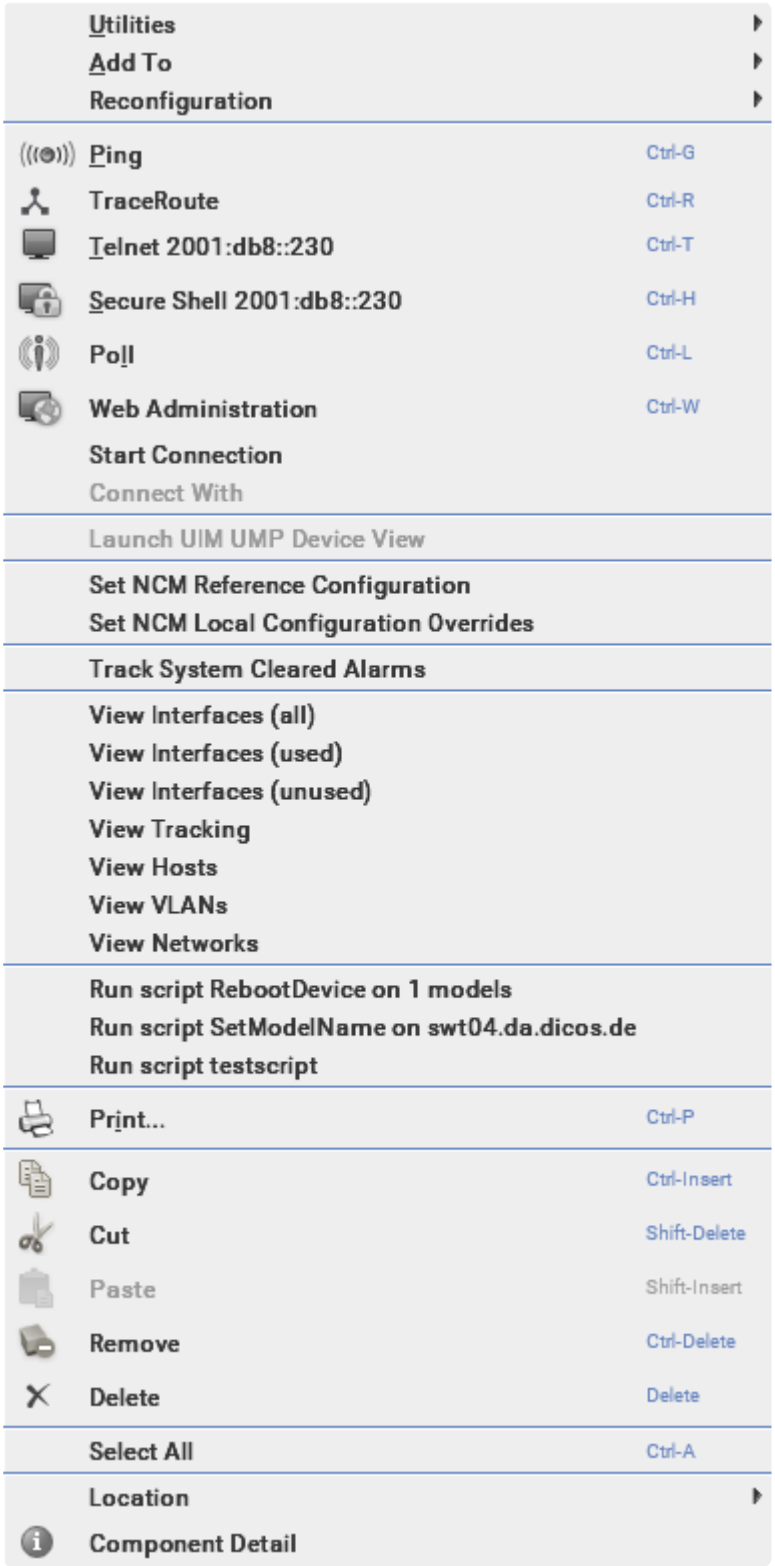
- OneClickExt is installed and the daemon is running (see Installation with Spectrum)
- At least one script is configured in `oneclickext.props` or an additional properties file
- The script's menu entry is present in `$SPECROOT/custom/console/config/custom-menu-config.xml`
- You have a device visible in the Spectrum topology view to test against

5.1.2 Step 1 — Open the topology view

Log in to the Spectrum console and navigate to the topology view containing the device you want to test against. Right-click the device to open the context menu.

OneClickExt script entries appear directly in the context menu, labelled Run script `<name>`:

Copyright © 1998 - 2025 DICOS GmbH Kommunikationssysteme



If no script entries appear, the client could not connect to the OneClickExt daemon at startup. See The client menu items are missing for diagnosis steps.

5.1.3 Step 2 — Open the script dialog

Click the script you want to run. A parameter dialog opens, showing the input fields defined for that script.

Copyright © 1998 - 2025 DICOS GmbH Kommunikationssysteme

Fill in any editable fields. Read-only fields are pre-populated from Spectrum attributes of the selected device and cannot be changed.

5.1.4 Step 3 — Run the script

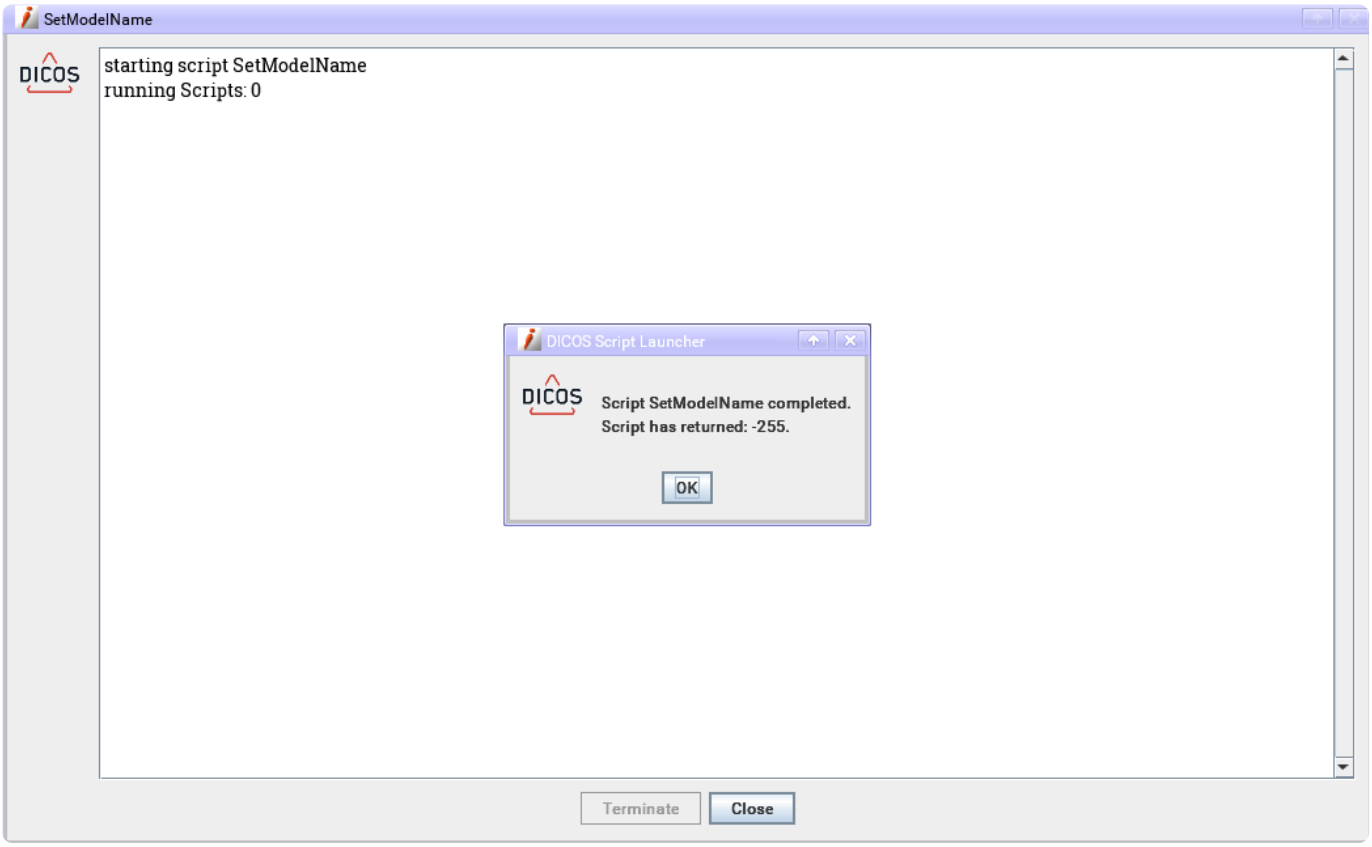
Click Run Script. If the script has a confirmation question configured (`secquestion`), a Yes/No dialog appears first, for example:

> Are you sure you want to change the model name from swt04.da.dicos.de to swt04-new?

Click Yes to proceed, or No to cancel.

5.1.5 Step 4 — Review the output

The output window opens and streams the script's stdout in real time. When the script finishes, a completion dialog reports the exit code:



Click OK to dismiss the completion dialog, then Close to close the output window.

Copyright © 1998 - 2025 DICOS GmbH Kommunikationssysteme

A negative exit code (such as -255) indicates the script returned an error. Exit codes can be formatted with colours to make results easier to read at a glance. See Exit code formatting.

5.1.6 What's next

- To add your own script, follow Defining a Script.
- To configure input parameters, validators, and combo boxes, see Parameters, Validators, and Actions.
- To restrict which users can run which scripts, see Permissions.

Copyright © 1998 - 2025 DICOS GmbH Kommunikationssysteme