



Service Framework

Administrator Manual

Software Version 1.0.0



EducateIT GmbH

Steinenbergstrasse 7, 4532 Feldbrunnen, Switzerland

Phone: +41 32 355 39 00

E-Mail: info@educateit.ch

Web: <http://www.educateit.ch/>

Service Framework – Administrator Manual

Copyright: ©2013 by EducateIT GmbH

EducateIT GmbH
Steinenbergstrasse 7
CH-4532 Feldbrunnen
Switzerland

Phone: +41 32 355 39 00
E-Mail: info@educateit.ch
Web: <http://www.educateit.ch/>

Illustrations and Text: TE
Layout and Design: TE
Last Modification: 2013-03-28 17:24

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by and means, electronic, mechanical, photocopying, recording, or otherwise, without the prior consent of the EducateIT GmbH. The EducateIT GmbH assumes no responsibility for errors or omissions, or for damages that may result from the use of information (especially including example configurations and listings) contained herein.

Content

Preface	5
Features	5
See Also	5
Command Line Arguments	6
The --serviceAction=... Argument	6
The --serviceName=... Argument	7
The --serviceTrace Flag	8

Preface

The “*EducateIT Service Framework*” is embedded in the software you got from our company. This framework communicates with the service API from Windows and keeps the software up and running. It also provides embedded functionality to install and remove the service from a computer.

This is the administrator manual for this framework. You got it along with the manual for the software. It describes the actions the framework is providing to you as user of the software.

Features

- Stable and failure safe control of the running software.
- Actions to start and stop the service from command line.
- Actions to install and remove the service on a system.
- Embedded trace mode to run a service from command line, bypassing the service manager of the system.
- Allows installation of the same service multiple times on the same computer with different identifiers.

See Also

The service framework uses the *Configuration B* system to parse and verify the command line arguments. It is *not* necessary to read the documentation of the *Configuration B* system to understand the *service framework*. But if you like to fully understand all aspects of the configuration, read the document “*Configuration B – Administrator Manual*”. You should get this document along with the manual for the software, otherwise please request it from us.

Command Line Arguments

Each software which uses the *service framework* provides a number of special command line arguments. The purpose of this command line arguments is just to control the service or execute service related actions. They are not passed to the application itself and are used only by the framework.

The following table provides a quick overview of the available command line arguments and their meanings. Each argument is described in detail in one of the following sections.

Argument	Description
--serviceAction=...	The service action argument takes the name of an action to execute. This name is case sensitive.
--serviceName=...	If you specify this argument together with an action, you can change the identifier of the service which is used in the service manager.
--serviceTrace	To start the service directly from the command line, without using the service manager, you have to use this flag. It starts the trace mode.

The --serviceAction=... Argument

If you specify the service action argument, only the specified action is executed, but the service itself is not started. There are a number of actions which are explained in the following sections.

```
ExampleService.exe --serviceAction=install
ExampleService.exe --serviceAction=stop
```

start

The **start** action starts the service using the service manager of the operating system. This is only working if the service is already installed properly on the system.

You can use this action to start a service from the command line or out of a script.

stop

The **stop** action stops the service using the service manager of the operating system. This is only working if the service is installed properly on the system and if the service manager started the service.

Note

Use the install action if the service executable is at its *final location*.

Do not move the service executable after installing it into the service database.

Use this action to stop the service from the command line or out of a script.

install

The `install` action installs the service in the service database of the operating system.

This action is using the absolute path to the service executable from which you are calling this action. This path is stored in the service database. For this reason, it is important that you call this action only from the final destination of the executable. After it is installed in the correct directory.

It is also important to know, that after installing a service, you must not move the executable to another location. Otherwise the service will not start, because the absolute path stored in the service database points at the wrong place.

uninstall

The `uninstall` action will remove the service from the service database. Please stop the service before it is removed.

The --serviceName=... Argument

This argument sets a different identifier for all the actions above. You only need to set a different service identifier if you plan to run the same service multiple times on the same system.

This service name can only contain letters (a-z), numbers, the point (.) and underline (_) character. The maximum length of the name is 64 characters.

If you install a service with a custom service identifier, this identifier is shown behind the name of the service in round brackets.

If you installed a service with a custom identifier, you also have to use the same identifier for the start, stop and uninstall action.

```
exmpl.exe --serviceAction=install --serviceName=second
exmpl.exe --serviceAction=uninstall --serviceName=second
exmpl.exe --serviceAction=start --serviceName=second
exmpl.exe --serviceAction=stop --serviceName=second
```

If you have a closer look to the created entries in the service database, you will discover that your service name is automatically added to the start command as command line parameter. This is necessary to address the right entry in the communication with the service manager.

Note

If you installed a service with a custom identifier, you also have to use the same identifier for the start, stop and uninstall action..

The --serviceTrace Flag

A service executable cannot be started from the command line, because a such executable has to connect with the service manager during startup.

If you start a service executable anyway from the command line, the service manager will reject the access and the start from the command line will fail as shown in the following example:

```
E:\Test>ExampleServer.exe
serviceRun() started.
Could not start service control dispatcher. You can not run this service from command line.
For testing, use the --serviceTrace command line option.
Could not start service dispatcher.
E:\Test>_
```

Start the Service from the Command Line

For quick tests you can use the `--serviceTrace` flag, which will start the service in the *trace mode*. In trace mode the software is started without using the service manager from the operating system. Therefore the software is also started in the security context of the user who is starting the service.

Be aware, that if the service you like to test needs additional privileges, you either have to start the service with a user which owns this rights or start the service as a different user.

Starting a service with the trace flag is shown in the following example. The colours and highlights are added for readability.

```
E:\Test>ExampleServer.exe --serviceTrace
serviceRun() started.
... Output from the service ...
Service changed state into: Starting
Service changed state into: Running
```

If everything is fine, you can see how the service state changes from “Starting” to “Running”. The service is now running in the console where you started the service.

Make sure you configured the log system of your software to write a log file containing debugging information. Watch this log file to see additional details about the service startup.

Stop the Running Service

The proper way to stop a service running in trace mode is by pressing `Ctrl+C` or `Ctrl+Break` on the keyboard while the console window has focus.

This will start the shutdown process for the service. When everything is stopped, the process will exit and you get back to the prompt.

In the following example you can see what happens after you press `Ctrl+C` while service is running in trace mode on the console. I added the console output from the start of the service and coloured it gray to provide a complete example. The highlights and colours are added manually to enhance the readability of the output.

```
E:\Test>ExampleServer.exe --serviceTrace
serviceRun() started.
... Output from the service ...
Service changed state into: Starting
Service changed state into: Running
Received Ctrl+C/Ctrl+Break. User requests stop of service.
Service changed state into: Stopping
Service changed state into: Stopped
Exit service with exit code: 0x00000000
E:\Test>_
```

At the end, you can see the exit code of the process. Usually an exit code of zero indicates a successful end of the process.



EducateIT GmbH
Steinenbergstrasse 7, 4532 Feldbrunnen, Switzerland
Phone: +41 32 355 39 00
E-Mail: info@educateit.ch
Web: <http://www.educateit.ch/>