

Example for YADE cascaded profiles

- [Introduction](#)
- [Managing settings in one profile](#)
 - [YADE settings managed in one profile](#)
- [Managing settings in cascaded profiles with fragments](#)
 - [YADE settings managed with cascaded profiles](#)
- [YADE profile from a settings file](#)
 - [JobScheduler Job jade_copy_localhost2server_with_cs](#)
 - [YADE Command Line Interface \(CLI\)](#)

Introduction

The YADE profiles can be cascaded or included by other profiles. This feature enable to create generic profile fragments, i.e. server connection settings, background service etc.

The other YADE profiles then can include such fragments. The cascaded profiles enable users to create generic profile fragments. The fragments can be included as target settings or source settings, thus the same host can be configured as source or target without duplicating the settings.

Managing settings in one profile

YADE settings managed in one profile

```
[copy_localhost2server_with_cs]
file_spec                = .*
operation                = copy
verbose                 = 9
;;
source_protocol         = local
source_host             = localhost
source_dir              = ${TEMP}/source
source_make_dirs        = true
;;
target_use_credential_Store = true
target_CredentialStore_FileName = C:\Program Files\sos-berlin.com\jobscheduler.1.6.4043
\djsmp02_4445\examples\keepassX-test.kdb
target_CredentialStore_password = testing
target_CredentialStore_KeyPath  = sos/server/mp.sos
;;
target_dir              = /home/test/inbound
target_make_dirs        = true
target_protocol         = ftp
target_transfer_mode    = ascii
```

Managing settings in cascaded profiles with fragments

YADE settings managed with cascaded profiles

```

[Keepass_DataBase]
use_credential_Store      = true
CredentialStore_FileName  = .\examples\keepassX-test.kdb
CredentialStore_password  = testing

[keepass_mp_ftp]
CredentialStore_KeyPath   = sos/server/mp.sos
protocol                  = ftp

[localhost_default]
protocol = local
host=localhost

[copy_localhost2server_with_cs2]
file_spec                 = .*
operation                 = copy
verbose                   = 9
;;
source_include            = localhost_default
source_dir                = ${TEMP}/source
source_make_Dirs         = true
;;
target_include            = Keepass_DataBase, keepass_mp_ftp
target_dir               = /home/test/inbound
target_make_dirs         = true
target_transfer_mode     = ascii

```

YADE profile from a settings file

The *jade_settings.ini* file can be called from JITL jobs and from YADE CLI, thus providing a single source for file transfer settings to JobScheduler JITL Jobs and to command line applications.

JobScheduler Job `jade_copy_localhost2server_with_cs`

The JobScheduler *jade_copy_localhost2server_with_cs.job.xml* JITL job can be configured with two parameters - profile and settings:

```

<?xml version="1.0" encoding="ISO-8859-1"?>

<job title="Localhost to FTP server copy (API Job for JobScheduler Advanced Data Exchange)" order="no" name="
jade_copy_localhost2server_with_cs">
  <settings >
    <log_level ><![CDATA[debug9]]></log_level>
  </settings>

  <description >
    <include file="jobs/jadeJob.xml"/>
  </description>

  <params >
    <param name="profile" value="copy_localhost2server_with_cs"/>
    <param name="settings" value="${SCHEDULER_HOME}/examples/jade_settings.ini"/>
  </params>

  <script language="java" java_class="sos.scheduler.jade.JadeJob"/>

  <run_time />
</job>

```

YADE Command Line Interface (CLI)

The *jade_settings.ini* profiles can be used by the YADE CLI client. See the YADE CLI client example below.

```
C:\Program Files\sos-berlin.com\jade\jade_client_1.6.4043\bin>set CS_HOME=C:\Program Files\sos-berlin.com\jobscheduler.1.6.4043\djsmp02_4445\examples
```

```
C:\Program Files\sos-berlin.com\jade\jade_client_1.6.4043\bin>jade.cmd -settings="%CS_HOME%"/jade_settings.ini -profile=copy_server2localhost_with_cs
```