The YADE Client Command Line Interface - Tutorial 4 - File Transfer

Introduction

This is the fourth in a series of articles describing how to get started with using the the YADE Client via its Command Line Interface and covers:

- renaming files and
- file cumulation.

YADE Tutorials List

YADE Client Command Line Interface

1. Getting Started and Downloading files
2. Simple File Selection
3. More Advanced File Selection
4. File Transfer
5. Checking files for completeness
6. Public / Private Key Authentication
7. Transfer via a Jump Host / DMZ

Prerequisites

See the Using the tutorials with the YADE Client Command Line Interface article for guidelines to setting up and running the tutorial examples.

Instructions for installing, configuring and using the XML Editor can be found in the XML Editor series of articles.

Download file

The configuration described in this tutorial can be downloaded and then directly opened in the XML Editor using the following link:

- local_2_local_select_adv.xml

Renaming files

The following example combines two renaming possibilities:

- Renaming files
- Adding a date-stamp to file names

See the YADE Parameter Reference - Rename article for a full list of the renaming possibilities.

Example

The example assumes that five *test_*.txt* files have previously been downloaded to the local `${USERPROFILE}\jade_demo\a` directory before the example is run. Such a download can be carried out using the simple YADE file transfer operation described in The YADE Client Command Line Interface - Tutorial 1 - Getting Started.

The file name part(s) to be replaced is/are defined in the *Replacing* parameter with round brackets () and the new part(s) in the *Replacement* parameter, separated by semi-colons.

The example shows a two-part substitution:

- *test* will be replaced with *file_* and
- '_' with the current date in the format specified.

Configuration structure

ProtocolFragment

ProtocolFragments are not specified in this configuration as both source and target are on the local file system. This has been done to allow *write* permissions at both source and target.

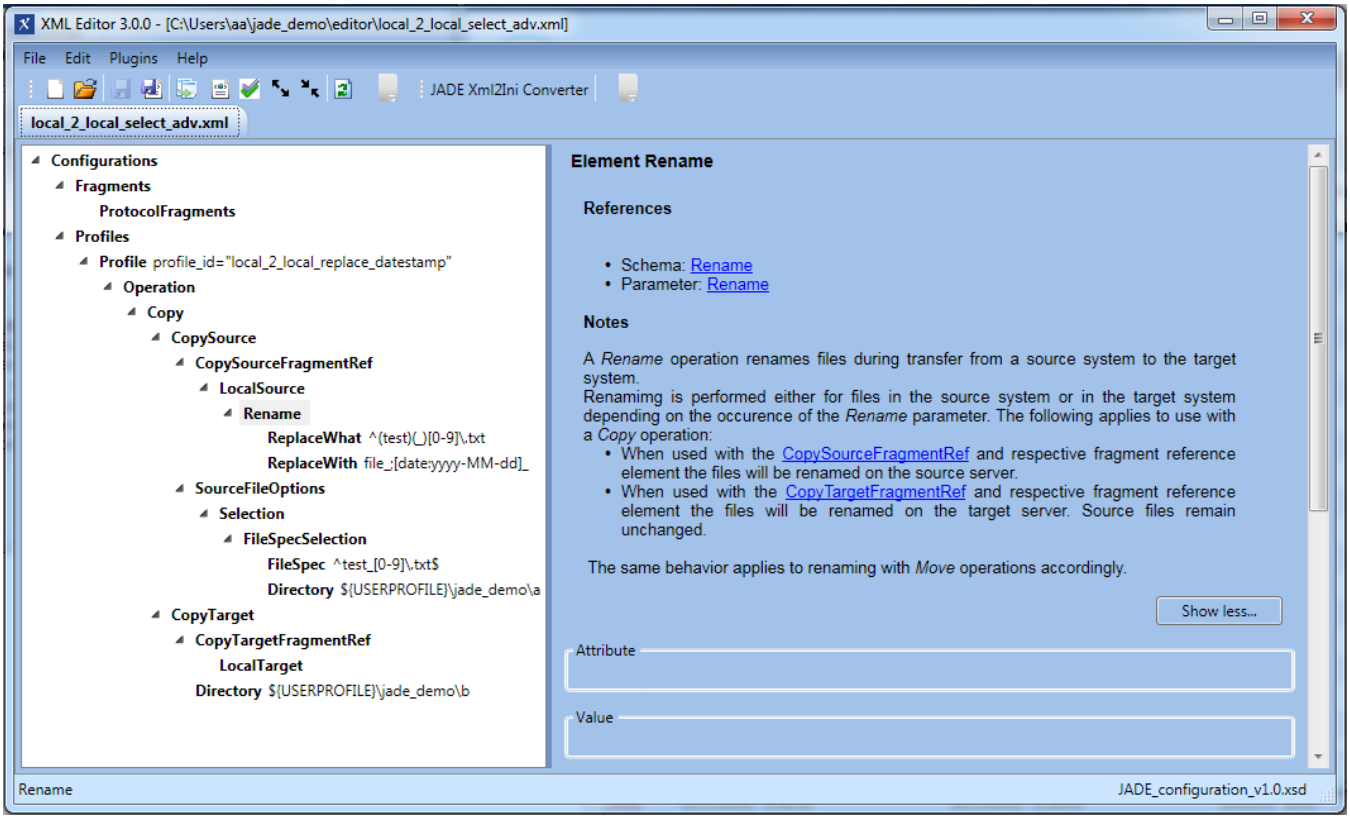
Profile

The meaning of the three elements required to specify renaming - *Rename*, *ReplaceWhat* and *ReplaceWith* elements should be self-explanatory.

Note that

- the *Rename* operation in this example described here is specified for the transfer source, meaning that the source files will be renamed and that the files copied to the target will not be. It is also possible to specify *Rename* for the transfer target, in which case the source files will remain unchanged for a *copy* operation.
- *write* permissions are required before names of files on a file system can be changed.

XML Editor Configuration



The Profile Code

The following code boxes can be opened to show the *Profile* used in this example in XML and in `settings.ini` formats.

'local_2_local_replace_datestamp' Profile in XML Format

```
<?xml version="1.0" encoding="utf-8"?>
<Configurations>
  <Fragments>
    <ProtocolFragments />
  </Fragments>
  <Profiles>
    <Profile profile_id="local_2_local_replace_datestamp">
      <Operation>
        <Copy>
          <CopySource>
            <CopySourceFragmentRef>
              <LocalSource>
                <Rename>
                  <ReplaceWhat><![CDATA[^(test)(_)[0-9]\.txt]]></ReplaceWhat>
                  <ReplaceWith><![CDATA[file_;[date:yyyy-MM-dd]_]]></ReplaceWith>
                </Rename>
              </LocalSource>
            </CopySourceFragmentRef>
            <SourceFileOptions>
              <Selection>
                <FileSpecSelection>
                  <FileSpec><![CDATA[^test_[0-9]\.txt$]]></FileSpec>
                  <Directory><![CDATA[${USERPROFILE}\jade_demo\a]]></Directory>
                </FileSpecSelection>
              </Selection>
            </SourceFileOptions>
          </CopySource>
          <CopyTarget>
            <CopyTargetFragmentRef>
              <LocalTarget />
            </CopyTargetFragmentRef>
            <Directory><![CDATA[${USERPROFILE}\jade_demo\b]]></Directory>
          </CopyTarget>
        </Copy>
      </Operation>
    </Profile>
  </Profiles>
</Configurations>
```

'local_2_local_replace_datestamp' Profile in settings.ini Format

```
[local_2_local_replace_datestamp]
operation                       = copy
source_protocol                 = local
source_replacing                = ^(test)(_)[0-9]\.txt
source_replacement              = file_;[date:yyyy-MM-dd]_
file_spec                       = ^test_[0-9]\.txt$
source_dir                      = ${USERPROFILE}\jade_demo\a
target_protocol                 = local
target_dir                      = ${USERPROFILE}\jade_demo\b
```

Running the Profile

This profile is called on Windows systems using the following command, depending on the YADE version being used:

Running the file transfer configuration in Windows format using an XML (JADE 1.11 and later) or settings.ini file (YADE 1.10 and earlier)

```
jade.cmd -settings="%USERPROFILE%\jade_demo\local_2_local_select_adv.xml" -profile="
local_2_local_replace_datestamp"
jade.cmd -settings="%USERPROFILE%\jade_demo\local_2_local_select_adv.ini" -profile="
local_2_local_replace_datestamp"
```

On Unix systems the profile is called using one of the following commands, again depending on the YADE version being used:

Running the file transfer configuration in Unix format using an XML (JADE 1.11 and later) or settings.ini file (YADE 1.10 and earlier)

```
./jade.sh -settings="${HOME}/jade_demo/local_2_local_select_adv.xml" -profile="local_2_local_replace_datestamp"
./jade.sh -settings="${HOME}/jade_demo/local_2_local_select_adv.ini" -profile="local_2_local_replace_datestamp"
```

Behavior

- The five *test_*.txt* files in the the *a* folder will be copied to the *b* folder under their original names. The source files will then be renamed as described above.
- If a file is found but not transferred - because for example it has zero bytes and *TransferZeroByteFiles* is set to *false* then the file name will remain unchanged.

File cumulation

YADE can cumulate the contents of individual files to a single target file:

- Files matching the *FileSpec* regular expression are cumulated together into a new file.
- The *CumulativeFileDelete* element is used to specify whether an already existing *CumulativeFile* is deleted and a new, empty, *CumulativeFile* created before the content of the individual files being transferred is added to this file. (The *CumulativeFile* is by default opened in append mode.)
- The individual, original files can be deleted if required in a *Move* operation or separate *Remove* operation.
- There is not a *de*-cumulate parameter available at the moment.

Example

The example assumes that five *test_*.txt* files have previously been downloaded to the local `${USERPROFILE}\jade_demo\a` directory before the example is run.

In the example, the contents of these five files written to a *CumulativeFile* in the local `${USERPROFILE}\jade_demo\b` directory. The content of each file being transferred is added successively to the *CumulativeFile* in the order in which the files are received. This order cannot be controlled during the file transfer operation.

The *CumulativeFile* is given the name specified in the *CumulativeFileName* parameter in the profile.

The text specified in the *CumulativeFileSeparator* parameter is incorporated in the *CumulativeFile* between the contents of the individual files.

Configuration structure

ProtocolFragment

ProtocolFragments are not specified in this configuration as both source and target are on the local file system. This has been done to allow *write* permissions at both source and target.

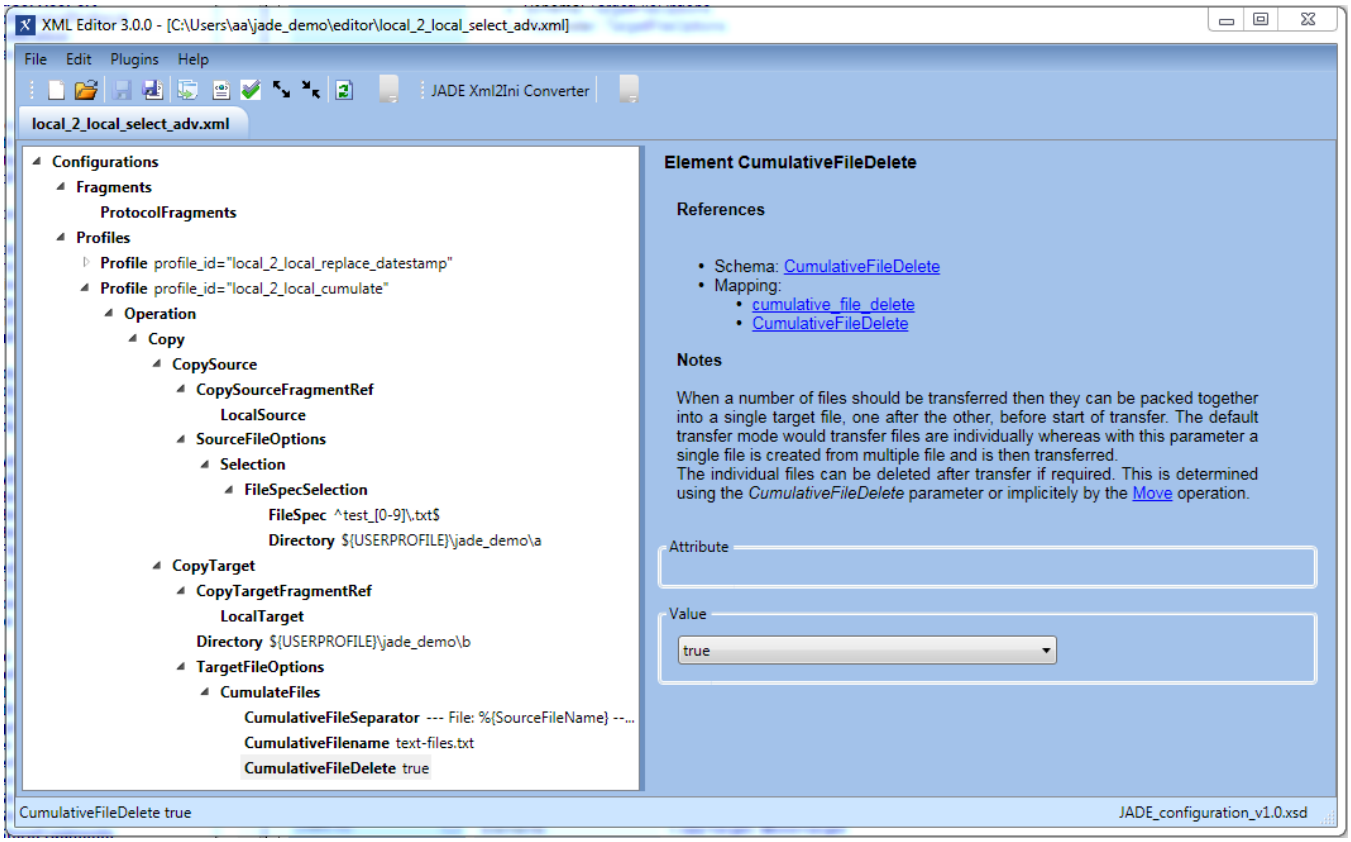
Profile

The three *CumulateFile* elements are child elements of the *TargetFileOptions* element as can be seen in the screenshot below.

Note that

- if the *Rename* operation is specified for the transfer source, then the source files will be renamed and the files copied to the target will not be. It is also possible to specify *Rename* for the transfer target, in which case the source files will remain unchanged for a *Copy* operation.
- *write* permissions are required before names of files on a file system can be changed.

XML Editor Configuration



The Profile Code

The following code boxes can be opened to show the *Profile* and *ProtocolFragments* used in this example in XML and in `settings.ini` formats.

The 'ftp_server_2_local_poll_minfiles' Profile in XML Format

```
<?xml version="1.0" encoding="utf-8"?>
<Configurations>
  <Fragments>
    <ProtocolFragments />
  </Fragments>
  <Profiles>
    <Profile profile_id="local_2_local_replace_datestamp">
     ...
    </Profile>
    <Profile profile_id="local_2_local_cumulate">
      <Operation>
        <Copy>
          <CopySource>
            <CopySourceFragmentRef>
              <LocalSource />
            </CopySourceFragmentRef>
            <SourceFileOptions>
              <Selection>
                <FileSpecSelection>
                  <FileSpec><![CDATA[^test_[0-9]\.txt$]]></FileSpec>
                  <Directory><![CDATA[${USERPROFILE}\jade_demo\a]]></Directory>
                </FileSpecSelection>
              </Selection>
            </SourceFileOptions>
          </CopySource>
          <CopyTarget>
            <CopyTargetFragmentRef>
              <LocalTarget />
            </CopyTargetFragmentRef>
            <Directory><![CDATA[${USERPROFILE}\jade_demo\b]]></Directory>
            <TargetFileOptions>
              <CumulateFiles>
                <CumulativeFileSeparator><![CDATA[--- File: %{SourceFileName} ---]]></CumulativeFileSeparator>
                <CumulativeFilename><![CDATA[text-files.txt]]></CumulativeFilename>
                <CumulativeFileDelete>true</CumulativeFileDelete>
              </CumulateFiles>
            </TargetFileOptions>
          </CopyTarget>
        </Copy>
      </Operation>
    </Profile>
  </Profiles>
</Configurations>
```

The 'ftp_server_2_local_poll_minfiles' Profile in settings.ini Format

```
[local_2_local_replace_datestamp]
...

[local_2_local_cumulate]
operation                       = copy
source_protocol                 = local
file_spec                       = ^test_[0-9]\.txt$
source_dir                      = ${USERPROFILE}\jade_demo\a
target_protocol                 = local
target_dir                      = ${USERPROFILE}\jade_demo\b
cumulate_files                  = true
cumulative_file_separator       = --- File: %{SourceFileName} ---
cumulative_file_name            = text-files.txt
cumulative_file_delete          = true
```

Running the Profile

This profile is called on Windows systems using the following command, depending on the YADE version being used:

Running the file transfer configuration in Windows format using an XML (JADE 1.11 and later) or settings.ini file (YADE 1.10 and earlier)

```
jade.cmd -settings="%USERPROFILE%\jade_demo\local_2_local_select_adv.xml" -profile="local_2_local_cumulate"
jade.cmd -settings="%USERPROFILE%\jade_demo\local_2_local_select_adv.ini" -profile="local_2_local_cumulate"
```

Running the file transfer configuration in Unix format using an XML (JADE 1.11 and later) or settings.ini file (YADE 1.10 and earlier)

```
./jade.sh -settings="${HOME}/jade_demo/local_2_local_select_adv.xml" -profile="local_2_local_cumulate"
./jade.sh -settings="${HOME}/jade_demo/local_2_local_select_adv.ini" -profile="local_2_local_cumulate"
```

Behavior

Opening the cumulative file *text-files.txt* in a text editor after transfer has been completed will show the content of all five individual files.

Note that the order of arrival of files in the target system determines the order in which they are added to the cumulative file.

Setting the *CumulativeFileDelete* parameter to *false* and running the transfer operation a number of times will show the that this setting allows the ongoing addition of content to the *CumulativeFile* - a procedure that is commonly used, for example, for log files.