Files Mover
User Manual

Files Mover

Contents

Description

This utility is designed to create partial package from a local folder (local git folder for example, that may contain all metadata, maybe partially broken) based on package.xml file.

Quick Start

```
java -jar files_mover.jar
-s "d:\Master_Repository"
-t "d:\ForceDotComMigrationTool\configValidateRelease_1.20"
-p "D:\tasks\Deploy\1.20.3\LOCAL_FILES\package.xml"
```

Workflow overview

0. Pull changes to your local git repository

Command line example:

```
cd [gitfolder]

git checkout Release/Cog_1.2X

git pull
```

Note: you can use any GUI Git utility for this.

1. Modify package.xml file, if needed.

You need to copy original package.xml file from local git folder and modify it in next cases:

- It contains "*" in <members/> tag (solution: explicitly list each required member).
- It contains not top-level members, e.g. `CustomField` instead of `CustomObject` (solution: replace necessary entries with top-levels ones).
- It contains Process builder Flow with the version, already activated (solution: remove entry from package.xml).

2. Run files_mover utility itself to create a resulting package:

```
cd [file_mover_folder]

java -jar files_mover.jar -s [local_git_folder] -t
[target_package_folder] -p [package.xml_full_path]
```

Note: -p paramenter is optional.

3. Validate created package

You can use for example Ant Migration tool:

```
cd [ant_folder]

ant validate
```

Prerequisites

Install latest Java 8 JRE (1.8.0_171 or higher)

Installation

Copy files_mover.jar to any folder you have access to, e.g. "`d:\ForceDotComMigrationTool\`"

Execution (with command line)

1) Go to the folder you copied `files_mover.jar` i.e.

```
cd /d d:\ForceDotComMigrationTool
```

2) Execute `java -jar files_mover.jar` with 2 required options:

`-s` –source folder (usually git folder) with metadata sources

`-t` – target folder you want to copy files to.

`-p` [optional argument] – full path to package.xml you want to use (it may be different from package.xml, located inside git, or you can point it to package.xml, located in git).
If this argument is omitted, than package.xml, located in source folder

For example:

```
java -jar files_mover.jar
-s "d:\Master_Repository"
-t "d:\ForceDotComMigrationTool\configValidateRelease_1.20"
-p "D:\tasks\Deploy\1.20.3\LOCAL_FILES\package.xml"
```

Notes:

1) Do not use line breaks, run files_mover as a usual command line utility with arguments, separated with spaces.
2) As in every command line utility, if argument contains spaces, enclose it in double quotes.

Expected result of execution:

Package, ready for validation or deployment is created.

Known limitations:

It is not possible to validate package if it contains Process builder flow that is already deployed to sandbox or production and activated.

Solution:

1) If you are deploying first time, do not forget to activate flow after deployment.

2) If you already have active flow, please exclude this flow entry from copy of package.xml.