
REVISION HISTORY			
-------------------------	--	--	--

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	swiftwrap.wrapperstaging	1
2	Swift properties related to wrapper staging	1
3	Protocols	1
3.1	direct://	1
3.2	file://	1
3.3	gsiftp://	2
3.4	http://	2
3.5	stage://	2
3.6	Adding new protocols	2
4	Examples of how swiftwrap is called	2
4.1	Using absolute pathnames	2
4.2	Using relative pathnames	2

1 swiftwrap.wrapperstaging

Table 1: Command line arguments

Argument	Description
-a	Command line arguments passed to app
-d	List of directories to create
-e	App to execute
-err	Where to write stderr
-i	Set stdin
-if	Input file(s)
-jobdir	Swift job directory
-of	Output file(s)
-out	Where to write stdout
-scratchdir	Swift scratch directory
-sk	Sitedir keep, true/false
-status	Provider or files
-urlprefix	URL Prefix
-wt	Wrapper log always transfer, true/false

2 Swift properties related to wrapper staging

Setting	Description
use.wrapper.staging	If set to true, enable wrapper staging. Defaults to false.
wrapper.staging.local.server	Swift passes a URL prefix to swiftwrap.wrapperstaging that determines how to access local files. This setting determines what the prefix is (for example, file://).

3 Protocols

Below is a list of protocols supported by `_swiftwrap.wrapperstaging` and a brief explanation of how they are implemented.

3.1 direct://

The direct protocol allows you to bypass file staging to the swift created work directory. Instead of making a copy of the file, a symlink will be created which points to the original location of the file.

```
file data <"direct:///home/davidkelly999/staging/absolute/direct/data.txt">;
```

3.2 file://

By default, files that use the `file://` URI, or files that specify no URI at all, will act in a way that is similar to the behavior of `direct://`. A symlink will be created in the Swift work directory that points to the original location of the file.

```
file data <"file:///home/davidkelly999/staging/absolute/file/data.txt">;
file data2 <"/home/davidkelly999/staging/absolute/normal/data.txt">;
```

3.3 gsiftp://

GSIFTP files will be transferred using the `globus-url-copy` command.

3.4 http://

HTTP is implemented in `_swiftwrap.wrapperstaging` via the `wget` command.

3.5 stage://

If a file begins with `stage://`, a copy of the file will be made in the work directory. Files get copied using `dd` with a 10MB block size.

```
file data <"stage:///home/davidkelly999/staging/absolute/stage/data.txt">;
```

3.6 Adding new protocols

The file `cog/modules/provider-local/resources/cog-provider.properties` allows you to set aliases. Define your new file protocol as an alias to "local". Suppose you would like to add a new protocol called "megaftp", you could add a line that looks like this:

```
alias=megaftp:local
```

In your swift script, you would reference the file as `megaftp://path/to/file.txt`.

Note

You will need to recompile after making this change.

4 Examples of how swiftwrap is called

4.1 Using absolute pathnames

In this example, Swift tries to map the file below using an absolute path

```
file data <"/home/davidkelly999/staging/dd-absolute/data.txt">;
```

Table 2: `_swiftwrap.wrapperstaging` arguments

<code>-if</code>	<code>file://localhost/home/davidkelly999/staging/dd-absolute/-data.txt</code>
<code>-of</code>	<code>catsn.0001.out</code>
<code>-urlprefix</code>	<code>file:///home/davidkelly999/staging/dd-absolute</code>
<code>-d</code>	<code>_root_/home/davidkelly999/staging/dd-absolute/.</code>
<code>-a</code>	<code>_root_/home/davidkelly999/staging/dd-absolute/data.txt</code>

4.2 Using relative pathnames

Here is an example using relative pathnames in Swift.

```
file data <"data.txt">;
```

Table 3: `_swiftwrap.wrapperstaging` arguments

<code>-if</code>	file://localhost/data.txt
<code>-of</code>	catsn.0001.out
<code>-urlprefix</code>	file:///home/davidkelly999/staging/dd-relative
<code>-d</code>	l.
<code>-a</code>	data.txt