

runChangeDependencies

Command Reference

Table of contents

1 Description.....	2
2 Usage.....	2

1. Description

Query model and switch Package dependencies to specified version or buildstamp.

The runChangeDependencies workflow command transitions the deployment model from one state to another, reconfiguring deployments to stop using one set of package dependencies and begin using a new set.

The figure below shows the model in two states, before and after. In the first state, the deployments depend on packages with buildstamp X, while there also exists a different set of packages with buildstamp Y. The runChangeDependencies command, re-maps the package dependencies such that the deployments no longer depend on the set X and instead depend on the set Y.

runChangeDependencies is useful for applying bulk model changes for package dependency re-mapping.

The basic logic of the command is as follows:

1. Queries the object traversing its hierarchy of child dependency Deployments and for each Deployment found, checks for package instances allowed as Package dependencies for that Deployment. The result of this query represents a set of candidate packages. This query takes options such as version which will filter the query to return packages that share the specified version string.
2. Reads the query result data and for each deployment adds or replaces the allowed package dependency. It is possible, the deployments only accept one dependency per package type. In this case, and if multiple exist, this command will prompt the user to choose the desired dependency.

The above operation occurs as a single transaction.

Warning:

If the Deployment has any singleton Package type constraints and multiple Package objects are found, the user will be prompted to choose one from among the set.

Currently, -version and -buildstamp options are interchangeable but this might change in future releases.

Note:

The value of -version argument is implicitly treated as a regular expression of the form: `/^opts.version $/`.

2. Usage

```
ctl -t Deployment -o <objectname> -c runChangeDependencies
[-allowmultiple <>] [-buildstamp <>] [-defaultPackageName <>]
[-deploymenttype <>] [-dispatch] [-failIfNotReplaced <>]
[-packagetype <>] [-resourcename <.*>] [-resourcetype
<[^\\.]*>] [-version <>]
```

2.1. Options

Option	Description	Type	Default
allowmultiple	<i>if true, allow multiple matches when default regex is included</i>	string	<code>\${entity.attribute.defaultAllow}</code>
buildstamp	<i>build identifier. required if no version option</i>	string	<code>\${entity.attribute.buildstamp}</code>
defaultPackageName	<i>default package name regex</i>	string	<code>\${entity.attribute.defaultPack}</code>
deploymenttype	<i>type of Deployment to modify. Defaults to type of this object.</i>	string	<code>\${entity.attribute.defaultDepl}</code>
dispatch	<i>dispatch command to child dependencies (unused)</i>	boolean	
failIfNotReplaced	<i>if true, fails if any packages don't have a matching replacement</i>	string	<code>\${entity.attribute.defaultFail}</code>
packagetype	<i>type of packages to change</i>	string	<code>\${entity.attribute.defaultPack}</code>
resourcename	<i>resource name pattern</i>	string	<code>.* / \${entity.attribute.dispatchRes}</code>
resourcetype	<i>resource type name</i>	string	<code>[^\\.]* / \${entity.attribute.dispatchRes}</code>
version	<i>version of packages to change to. required if no buildstamp option</i>	string	<code>\${entity.attribute.buildstamp}</code>