

find

Command Reference

Table of contents

1 Description.....	2
2 Usage.....	2
3 Examples.....	3

1. Description

Find objects matching name and type

Queries the model for objects matching the specified name and type. Query results can be presented in several output formats.

Inspired by ProjectBuilder#find-objects

static: This command can be run outside of an object context.

2. Usage

```
ctl -m modelutil -c find -depot <> [-format <properties>]
[-name <.*>] [-output <>] [-type <.*>] [-useregex <>true>]
```

2.1. Options

Option	Description	Type	Default
depot	<i>project depot name</i>	string	<code>#{context.depot}</code>
format	<p><i>output format.</i> <i>properties,xml,maprefuri</i></p> <ul style="list-style-type: none"> properties: entity.properties format xml: viewdoc xml format. TODO: create an xsl template to generate project.xml maprefuri: comma separated list of maprefuris 	string	properties
name	<i>object name (or regex)</i>	string	.*
output	<i>Output file</i>	string	
type	<p><i>type name (or regex)</i></p> <p>If the regex parameter is not specified, then all objects of this type are matched. This means any object that is also a</p>	string	.*

find

	subtype.		
useregex	<i>Use regexes to match object name and type</i>	string	true

3. Examples

Find all Setting objects and save the output as Java properties (the default format). This will return all objects that are Setting subtypes and save it in the file "/tmp/find.out".

```
ctl -p demo -m modelutil -c find -- -type Setting -output /tmp/find.out
```

Find also lets you specify the name and type as a regular expression.

This example will find all objects that have a name that begins with the pattern "my.*" and any type matching the pattern ".*Service. The results are written to the console.

```
ctl -p demo -m modelutil -c find -- -name myservice -regex true
```