

Griffon 1.4.0

- 1 Overview
 - 1.1 Dependencies
- 2 Features
 - 2.1 Buildtime
 - 2.1.1 License report
 - 2.1.2 Support Bintray's JCenter
 - 2.2 Runtime
 - 2.2.1 Custom Event Classes
 - 2.2.2 @InjectedResource on Methods
 - 2.3 Compatibility
- 3 Sample Applications
 - 3.1 File Viewer
 - 3.2 GroovyEdit
 - 3.3 Font Picker
 - 3.4 Greet
 - 3.5 SwingPad
 - 3.6 GroovyFXPad
 - 3.7 FxBrowser
 - 3.8 WeatherWidget
- 4 Release Notes
 - 4.1 1.4.0

Overview

Griffon 1.4.0 - is the latest release in the 1.x series

Dependencies

The following dependencies have been upgraded

- groovy-all 2.1.7
- ant 1.9.2
- ant-launcher 1.9.2
- ant-junit 1.9.2
- slf4j-api 1.7.5
- slf4j-log4j12 1.7.5
- jcl-over-slf4j 1.7.5
- jul-to-slf4j 1.7.5
- gant_groovy2.1 1.9.9
- grails-docs 2.2.3
- junit 4.11
- hamcrest-core 1.3
- org.springframework.beans 3.2.4.RELEASE
- org.springframework.context 3.2.4.RELEASE
- org.springframework.context.support 3.2.4.RELEASE
- org.springframework.core 3.2.4.RELEASE

Features

Buildtime

License report

This command collects and displays all available licenses of runtime dependencies required by the application. The command creates both simple text and XML output reports, simply invoke it as follows

```
griffon license-report
```

A label ****UNKNOWN**** will be used if a license cannot be resolved.

Support Bintray's JCenter

This release enables a quick route for gaining access to dependencies hosted by [Bintray](#) repositories. You may specify Bintray's JCenter or a custom repository using any of the following formats

```
jcenter()
bintray(username: 'foo', repository: 'bar')
bintray('foo/bar')
```

Runtime

Custom Event Classes

Since the early days of the framework you could send custom event messages by identifying each event using an unique name. Now you can do it by sending custom event instances, thus keeping events and arguments in a common place. Here's a very simple example of a controller triggering such custom event and handling it as well.

```
class MyController {
  void mvcGroupInit(Map args) {
    app.addApplicationListener(Foo) { e -> assert e instanceof Foo }
  }
  def fooAction = { evt = null ->
    // do something
    app.event(new MyController.Foo(this))
  }
  static class Foo extends griffon.core.Event {
    Foo(Object source) { super(source) }
  }
}
```

@InjectedResource on Methods

The *@InjectedResource* annotation may now be applied to a method provided it represent a property setter.

Compatibility

The following plugins need to be updated to their latest versions

- lombok

Sample Applications

Griffon 1.4.0 ships with 8 sample applications of varying levels of complexity demonstrating various parts of the framework. In order of complexity they are:

File Viewer

File Viewer is a simple demonstration of creating new MVCGroups on the fly.

Source: `samples/FileViewer`

To run the sample from source, change into the source directory and run `griffon run-app` from the command prompt.

GroovyEdit

GroovyEdit is an improved version of FileViewer that uses custom observable models.

Source: `samples/GroovyEdit`

To run the sample from source, change into the source directory and run `griffon run-app` from the command prompt.

Font Picker

Font Picker demonstrates form based data binding to adjust the sample rendering of system fonts.

Source: `samples/FontPicker`

To run the sample from source, change into the source directory and run `griffon run-app` from the command prompt.

Greet

Greet, a full featured Griffon Application, is a Twitter client. It shows Joint Java/Groovy compilation, richer MVCGroup interactions, and network service based data delivery.

Source: `samples/Greet`

To run the sample from source, change into the source directory and run `griffon run-webstart` from the command prompt. Because Greet uses JNLP APIs for browser integration using `run-app` will prevent web links from working.

SwingPad

SwingPad, a full featured Griffon Application, is a scripting console for rendering Groovy SwingBuilder views.

Source: `samples/SwingPad`

To run the sample from source, change into the source directory and run `griffon run-app` from the command prompt.

GroovyFXPad

GroovyFXPad, a full featured Griffon Application, is a scripting console for rendering [GroovyFX](#) views.

Source: `samples/GroovyFXPad`

To run the sample from source, change into the source directory and run `griffon run-app` from the command prompt.

FxBrowser

FxBrowser is a trivial JavaFX powered browser that demonstrates Griffon's integration with JavaFX.

Source: `samples/FxBrowser`

To run the sample from source, change into the source directory and run `griffon run-app` from the command prompt.

WeatherWidget

WeatherWidget demonstrates binding, threading and plugin usage.

Source: `samples/WeatherWidget`

To run the sample from source, change into the source directory and run `griffon run-app` from the command prompt.

Release Notes

1.4.0

T	Key	Summary	Assignee	Reporter	P	Status	Resolution	Created	Updated	Due
	GRIFFON-341	Automatic packaging of licenses	Andres Almiray	ssgriffonuser		Resolved	Fixed	Mar 18, 2011	Jul 16, 2013	
	GRIFFON-374	Multiple calls of CreateMvc generate duplicate group entries	Andres Almiray	Andres Almiray		Resolved	Fixed	Jun 17, 2011	Jul 26, 2013	
	GRIFFON-514	"griffon" modifies timestamp in	Andres	Thomas		Resolved	Fixed	May 30,	Jul 26,	

	application.properties on every run	Almiray	Ferris Nicolaisen	Resolved		2012	2013
GRIFFON-634	getArtifactType() is missing in AbstractGriffonArtifactScript.java	Andres Almiray	Edo	Resolved	Fixed	May 16, 2013	May 16, 2013
GRIFFON-637	Metacomponents aren't destroyed until app shutdown	Andres Almiray	Edo	Resolved	Fixed	May 22, 2013	Jul 26, 2013
GRIFFON-638	Enable custom Event classes to be used with the application's event bus	Andres Almiray	Andres Almiray	Resolved	Fixed	Jul 02, 2013	Jul 02, 2013
GRIFFON-639	Upgrade dependencies	Andres Almiray	Andres Almiray	Resolved	Fixed	Jul 02, 2013	Jul 02, 2013
GRIFFON-640	Apply Javadoc patch required by jdk7u25	Andres Almiray	Andres Almiray	Resolved	Fixed	Jul 02, 2013	Jul 02, 2013
GRIFFON-641	[Swing] MissingPropertyExceptionNoStack: No such property: resourceResolver for class: griffon.swing.SwingApplet	Andres Almiray	Andres Almiray	Resolved	Fixed	Jul 12, 2013	Jul 12, 2013
GRIFFON-642	Plugin resolution fails if using JDK8	Andres Almiray	Andres Almiray	Resolved	Fixed	Jul 24, 2013	Jul 24, 2013
GRIFFON-645	Support property injection with @InjectedResource	Andres Almiray	Andres Almiray	Resolved	Fixed	Sep 04, 2013	Sep 04, 2013
GRIFFON-602	plugin install/uninstall not working correctly	Andres Almiray	Boris Matthys	Resolved	Fixed	Jan 18, 2013	Jul 26, 2013
GRIFFON-611	https link creation in project documentation	Andres Almiray	Marek Hudik	Resolved	Fixed	Mar 04, 2013	Jul 26, 2013
GRIFFON-636	run-webstart fails on Windows when the application name has a space	Andres Almiray	Brett Borchardt	Resolved	Fixed	May 20, 2013	Jul 26, 2013
GRIFFON-648	Handling files copied from griffon-app/conf/dist/bin	Andres Almiray	Ivan Arrizabalaga Getino	Resolved	Fixed	Sep 17, 2013	Oct 04, 2013
GRIFFON-644	Support Bintray's JCenter repository for resolving dependencies	Andres Almiray	Andres Almiray	Resolved	Fixed	Aug 06, 2013	Aug 06, 2013

16 issues