Griffon 1.0.0

- 1 Overview
- 2 New Features
  - 2.1 Buildtime
    - 2.1.1 Dependencies
    - 2.1.2 New RunScript command
    - 2.1.3 Additional JVM Flags
    - 2.1.4 Unparsed Command Arguments
    - 2.1.5 Default Repository for Releases
    - 2.1.6 Framework Plugins
    - 2.1.7 Sublime Text2 Integration
    - 2.1.8 Start Debug Process in Suspend Mode
  - 2.2 Runtime
    - 2.2.1 Config Application Locale
    - 2.2.2 MetaComponent View Node
- 3 Breaking Changes
  - 3.1 Runtime
- 4 Sample Applications
  - 4.1 File Viewer
  - 4.2 GroovyEdit
  - 4.3 Font Picker
  - 4.4 Greet
  - 4.5 SwingPad
  - 4.6 GroovyFXPad
  - 4.7 FxBrowser
  - 4.8 WeatherWidget
- 5 Release Notes
  - 5.1 1.0.0

Overview

Griffon 1.0.0 – "Gryphys magnus" - is the latest and greatest Griffon release.

New Features

Buildtime

Dependencies

Griffon 1.0.0 upgrades the following dependencies

- Spring 3.1.1.RELEASE

New RunScript command

It's now possible to run arbitrary Groovy scripts with either the application classpath fully built or a bootstrapped application, as it happens with the `console` and `shell` commands. Simply call `run-script` with at least one

argument: the path to the script. Remaining arguments will be sent directly to the running script.

Additional JVM Flags

There are some JVM flags such as `-javaagent` that cannot be set via System properties. For such cases Griffon allows setting values for these flags. You can specify flag values either in configuration files (`BuildConfig.groovy` or `settings.groovy`) or at the command line prompt.

Here's an example using JRebel as a java agent, set in `BuildConfig.groovy`

```
griffon {
    app {
        jvmOpts = [

'-javaagent:/Applications/ZeroTurnaround/JRebel/jrebel.jar'
        ]
    }
}
```

The following command has the same effect however the value is specified at the command prompt

```
griffon run-app
--jvm-opts='-javaagent='/Applications/ZeroTurnaround/JRebel/jrebel.jar'
```

Unparsed Command Arguments

Griffon will automatically parse and format command line arguments. For those cases where you'd want to perform your' own parsing Griffon now exposes the original arguments as a variable in the running script's biding. The variable name is `unparsedArgs` and it's of type `String[]`.

Default Repository for Releases

Griffon 0.9.5 introduced artifact repositories and gave you the ability to specify default repositories for searching and local installation. Missing was the option to specify the default repository to where releases would be posted. Now you can define a value for `griffon.artifact.repository.default.release` either as a System property at the command prompt or in any of the build time configuration files. Here's an example of a plugin being released to a local Griffon repository.

```
griffon
-Dgriffon.artifact.repository.default.releas
e='my-local' release-plugin
```

When specified at the command prompt it has the same effect as defining a value for `--repository`, so this flag works better if placed in the build time configuration files, specifically `settings.groovy`.

Framework Plugins

Perhaps the biggest change for buildtime in this release is the availability of framework plugins. Plugins can now be installed at the distribution level, not just project level. this means for example installing the Git plugin at the framework level enables the usage of Git to all projects. Not all plugins can be installed as framework plugins, particularly those that deliver runtime capabilities.

If a plugin is installed both at the framework and project level then the one listed at the project level wins, even if it's an older version. The Griffon Artifact Portal identifies which plugins can be installed as framework plugins. All plugin related commands (`install-plugin`, `uninstall-plugin`, `plugin-info`, `list-plugins`, `list-plugin-updates`) can now be used outside of a Griffon project.

If `install-plugin` and `uninstall-plugin` are executed within a Griffon project then the plugin will affect the project only, otherwise it will affect the distribution. You can force a framework plugin install/uninstall if the `--framework` flag is specified at the command prompt.

Sublime Text2 Integration

Sublime Text2 support is now in place. You can generate a project file suitable for this editor just by typing

```
griffon integrate-with --sublimetext2
```

Start Debug Process in Suspend Mode

Prior to this release suspended debug mode was not engaged at all. Now you can configure it at the command prompt.

```
griffon run-app --debug --debug-suspend=y
```

Runtime

Config Application Locale

The application's Locale can now be set via configuration. Simply specify a value of type `java.util.Locale` or `java.lang.String` for the `application.locale` key in `Application.groovy`, like this

```
application {
    title = 'Sample'
    startupGroups = ['sample']

    locale = 'es'

    // Should Griffon exit when no Griffon
created frames are showing?
    autoShutdown = true

    // If you want some non-standard
application class, apply it here
    //frameClass = 'javax.swing.JFrame'
}
```

The string follows the format `language[_country[_variant]]`

MetaComponent View Node

You can now instantiate a metacomponent via View nodes. MetaComponents were introduced in Griffon 0.9.5. They are MVC groups that can be treated as "components". The `metaComponent` node automatically embeds the group's View into the current view. For example, a group with name custom defined as

```
mvcGroups {
    'custom' {
        model      = 'sample.CustomModel'
        view       = 'sample.CustomView'
        controller =
'sample.CustomController'
        config {
            component = true
            title = 'My Default Title'
        }
    }
}
```

This component can be used in a View as follows

```
panel {
    migLayout()
    button('Click me', constraints: 'left')
    widget(metaComponent('custom', title:
'New Title'), constraints: 'center')
}
```

Breaking Changes

Runtime

All methods marked as deprecated in previous release has been removed. Pay special attention to the threading methods renamed in 0.9.5.

Sample Applications

Griffon 1.0.0 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

1.0.0

> The JIRA server does not support trust requests. Issues have been retrieved anonymously.You can set the macro to always use an anonymous request by setting the `anonymous` parameter to `true`

JIRA Issues (18 issues)

Type	Key	Summary	Assignee	Reporter	Priority	Status	Resolution	Created	Updated	Due
	GRIFFON-414	Support framework plugins	Andres Almiray	Andres Almiray		Closed	Fixed	Sep 18, 2011	Jul 09, 2012	
	GRIFFON-490	Can't install a plugin with a SNAPSHOT version	Andres Almiray	Andres Almiray		Closed	Fixed	Apr 05, 2012	Jul 09, 2012	
	GRIFFON-491	Can't run test-app from griffonshin 0.9.5	Andres Almiray	Dean Iverson		Closed	Fixed	Apr 10, 2012	Jul 09, 2012	
	GRIFFO	create-u	Andres	Dean		Clos	Fixed	Apr 10,	Jul 09,	

	N-492	nit-spec does not work from griffonsh	Almiray	Iverson		ed		2012	2012	
	GRIFFON-494	Move RootFactory from swing plugin into core	Andres Almiray	Andres Almiray		Closed	Fixed	Apr 18, 2012	Jul 09, 2012	
	GRIFFON-495	Support metacomponent creation via View node	Andres Almiray	Andres Almiray		Closed	Fixed	Apr 18, 2012	Jul 09, 2012	
	GRIFFON-501	Can no longer set system properties for a Griffon app	Andres Almiray	Dean Iverson		Closed	Fixed	Apr 29, 2012	Jul 09, 2012	
	GRIFFON-502	Original args array is no longer passed to custom scripts in 0.9.5	Andres Almiray	Marc Paquette		Closed	Fixed	Apr 30, 2012	Jul 09, 2012	
	GRIFFON-504	Exception thrown from _GriffonCreateArtifacts.groovy	Andres Almiray	Dean Iverson		Closed	Fixed	May 01, 2012	Jul 09, 2012	
	GRIFFON-505	CreateScript --with-command-support creates wrong	Andres Almiray	Andres Almiray		Closed	Fixed	May 02, 2012	Jul 09, 2012	

		filename for help files								
	GRIFFON-506	package command does not deal with native libraries correctly	Andres Almiray	Ken Krebs		Closed	Fixed	May 08, 2012	Jul 10, 2012	
	GRIFFON-507	Allow additional JVM flags to be set before running an application	Andres Almiray	Andres Almiray		Closed	Fixed	May 09, 2012	Jul 09, 2012	
	GRIFFON-508	Add a RunScript command similar to the one found in Grails	Andres Almiray	Andres Almiray		Closed	Fixed	May 09, 2012	Jul 09, 2012	
	GRIFFON-510	Can't release to griffon-local from griffonsh	Andres Almiray	Dean Iverson		Closed	Fixed	May 14, 2012	Jul 09, 2012	
	GRIFFON-511	run-script eats args beginning with dash (-) or double dash (--)	Andres Almiray	Marc Paquette		Closed	Fixed	May 15, 2012	Jul 09, 2012	
	GRIFFON-512	Allow the application locale to be set via	Andres Almiray	Andres Almiray		Closed	Fixed	May 25, 2012	Jul 09, 2012	

		configuration								
	GRIFFON-499	Presence of some folders (.DS_Store) under a project's plugins directory leads to "IllegalArgument Exception: Cannot create Release based on file ...."	Andres Almiray	Marc Paquette		Closed	Fixed	Apr 23, 2012	Jul 09, 2012	
	GRIFFON-503	It would be nice if there was a way to mark a plugin as private	Andres Almiray	Dean Iverson		Closed	Fixed	May 01, 2012	Jul 09, 2012	