

Griffon 0.9.4

Error rendering macro 'toc' : null

Overview

Griffon 0.9.4 – "Aquila kienerii" - is a maintenance release of Griffon 0.9.

New Features

Buildtime

Specify source encoding

It's now possible to specify a source encoding for files during compilation. Specify `griffon.source.encoding` either as a System property or inside `BuildConfig.groovy` / `settings.groovy`

```
griffon
-Dgriffon.source.encoding='ISO-8859-1'
compile
```

Runtime

Eager instantiation of Services

Services will be lazily instantiated unless the following flag is set to true in `Config.groovy`.

```
griffon.services.eager.instantiation = true
```

Service handling

All services instances will become available through an instance of type `griffon.core.ServiceManager`. This helper class exposes available services via a Map. You can query all currently available services in the following manner

```
app.serviceManager.services.each { name,
instance ->
    // do something cool with services
}
```

You can also query for a particular service instance in the following way

```
def fooService =
app.serviceManager.findService('foo')
```

It's worth mentioning that the previous method will instantiate the service if it wasn't available up to that point.

Identify the top node in a secondary View script

Most of the times a View script defines a single node. This node is set as the return value of the script, which is useful for composing secondary Views by calling the `build()` method on the primary View.

However, there may be times where the "root" node is not the last expression evaluated in the script (due to additional configuration like bindings or setting up listeners for example). In this case the developer must set a reference to the top node and make an explicit return at the end of the script.

There's a new node called `root()` that can either identify or grant access to the top level node of a View script depending on the usage mode.

Secondary view script (assuming class name is Secondary)

```
root(
    l = label('foooooooooo')
)

noparent {
    l.text = 'FOOOOOO !!!'
}
```

Primary view script

```
build(Secondary)
...
widget root(Secondary)
```

Custom managers

Developers are now able to define the type of `griffon.core.ArtifactManager`, `griffon.core.AddonManager` and `griffon.core.MVCGroupManager` to be instantiated if the default one is not suitable for their needs. A set of matching factories can be found in the `griffon.core.factories` package. A developer must configure the chosen factory in `Config.groovy`. For example

```
app.artifactManager.factory =
'com.acme.CustomArtifactManagerFactory'
app.addonManager.factory =
'com.acme.CustomAddonManagerFactory'
app.mvcGroupManager.factory =
'com.acme.CustomMVCGroupManagerFactory'
```

Breaking Changes

Runtime Behavior

All variants of the method `griffon.core.MVCGroupHandler.buildMVCGroup` no longer return a `Map` with all MVC members keyed by type. They now return an instance of `griffon.core.MVCGroup`.

The following application events have changed their parameters:

Event	Params
InitializeMVCGroup	MVCGroupConfiguration config, MVCGroup group
CreateMVCGroup	MVCGroup group
DestroyMVCGroup	MVCGroup group

Applications no longer have a dynamic method

Dependencies

New versions for the following dependencies

- Groovy 1.8.2

Sample Applications

Griffon 0.9.4 ships with 5 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.

0.9.4 Release Notes

 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`

Griffon 0.9.4 Resolved Issues (17 issues)		
Type	Key	Summary
	GRIFFON-396	Views should be able to identify the top node and expose to other MVC members
	GRIFFON-397	All updates on UI components resulting from a declarative binding should be guaranteed to happen in the UI thread
	GRIFFON-398	allow other encodings than utf-8
	GRIFFON-399	Allow instantiation of custom ArtifactManager class
	GRIFFON-400	Allow instantiation of custom AddonManager class
	GRIFFON-401	Allow instantiation of custom MVCGroupManager class
	GRIFFON-402	Create a type that identifies an MVC group and its configuration
	GRIFFON-404	An app created with a capitalized name will not run by default
	GRIFFON-407	Exception within closure of an action is suppressed
	GRIFFON-416	Allow services to be injected into other services
	GRIFFON-417	SwingPad fails because of Threading issue - maybe within RSyntaxTextArea
	GRIFFON-421	app.services throws MPE
	GRIFFON-422	Allow services to be instantiated eagerly
	GRIFFON-423	Can't resolve addon from a plugin with a score in its name
	GRIFFON-384	slurper.parse(SomeConfGroovyClasses) fails after clean
	GRIFFON-405	Missing colon in Griffon DSLD
	GRIFFON-425	calling app.shutdown() from



[Initialize.groovy causes a NPE](#)