

# Griffon 0.9.3

## Error rendering macro 'toc' : null

### Overview

Griffon 0.9.3 – "Aquila morphnoides" - is a maintenance release of Griffon 0.9.

### New Features

#### Buildtime

#### Dependencies

Griffon 0.9.3 relies on Groovy 1.8.1 and Gant-groovy-1.8 19.6. This means code compiled with previous versions of Groovy might not work immediately out of the box. The reason being that Groovy 1.7 and 1.8 are not entirely binary compatible. Griffon 0.9.3 includes a temporary workaround. It's perfectly safe to recompile your code with this release.

#### Change dock icon in OSX

RunApp now assumes there is a `<applicationFile>.icns` file located at `griffon-app/conf/dist/shared`. This setting can be overridden by defining a configuration flag in `griffon-app/conf/BuildConfig.groovy`

```
application.icon =  
' /application-relative/path/to/file.icns '
```

If neither of these two settings is available then the default icon (`griffon.icns`) will be used

#### Jumpstart archetype improvements

The jumpstart archetype code has been streamlined with the addition of the [Actions Plugin](#). Now all actions are automatically configured via resources, which reduces the amount of code to set them up. Also, Java based templates have been added.

#### Templates

Almost every `create-*` script relies on a template in order to create the desired artifact. This template could only be overridden if a plugin (or the application) provided another template that matched the same name as the original one. Clearly that is not a scaling solution so the template system has been revamped to accept a different template depending on the options specified during the command invocation. For example, creating a service with a custom template named `MyService` you would now invoke the following command

```
griffon create-service -service=MyService
```

Basically you specify the type of the artifact as an option (using lowercase on the first char), which leads to the following cases to be valid too

```
griffon create-mvc -view=CustomView
```

Speaking of MVC groups, the `create-mvc` command has additional settings that can be configured, like this

```
griffon create-mvc foo -group=Custom
```

Creates a new group definition where MVC member templates are assumed to be `CustomModel`, `CustomView` and `CustomController`. Will use the default template when there's no match.

```
griffon create-mvc foo -skipController=true
```

Creates a new group definition without a Controller. The configuration will look like this

```
mvcGroups {
    // MVC Group for "foo"
    'foo' {
        model      = 'foo.FooModel'
        view       = 'foo.FooView'
    }
    ...
}
```

```
griffon create-mvc foo
-withController=foo.BarController
```

Creates a new group definition with another Controller class. The Controller is assumed to exist, a file will not be created for it. The configuration will look like this

```
mvcGroups {
    // MVC Group for "foo"
    'foo' {
        model      = 'foo.FooModel'
        view       = 'foo.FooView'
        controller = 'bar.BarController'
    }
    ...
}
```

## Application Archetypes

There's a new application archetype available. It bootstraps an application in a similar way as Ubuntu's Quickly does. Here's how to use it

```
griffon create-app sample
-archetype=jumpstart
```

The generated code is fully i18n aware and customizable.

## Default Package Name

When an application is created it will attempt to use the name of the application as the default package name for the initial MVC group (unless `-skipPackagePrompt` is specified). This package name is now stored in `griffon-app/conf/BuildConfig.groovy` for future references by additional scripts.

## Filter Plugins by Platform

Calling `griffon list-plugins` will now filter the list automatically, leaving out those plugins that do not match the current development platform you're working on.

## IDE Support

It's been a while since IDEA added support for [Groovy DSL descriptors](#). Recently Eclipse gained the same capabilities via DSLD ([explained by Andrew Eisenberg](#), [Vladimír Oraný](#) posted a thorough tutorial [here](#)). During Gr8conf Copenhagen 2010 a group of Griffon enthusiasts banded together under the Hackergarten space and created a couple of GDSDLs specifically tailored for Griffon. This year was no different, a similar group managed to build improved versions of these GDSDLs plus the first cur of DSLDs. These scripts should be automatically picked up by either IDE as soon as the Griffon jars are placed in the classpath. So what exactly do you gain now?

- autocompletion of Swing nodes when working in a Griffon View script.
- autocompletion of methods from standard Griffon artifacts.
- autocompletion of methods from the following AST transformations: @Bindable, @Vetoable, @EventPublisher, @MVCAware, @ThreadingAware

## Configure Application's Manifest

It's now possible to configure the manifest that's placed inside the application's jar. Griffon will automatically create the following entries in the application's manifest

```
Manifest-Version: 1.0
Ant-Version: Apache Ant 1.8.1
Created-By: ${jvm.version} (${jvm.vendor})
Main-Class: ${griffonApplicationClass} |
${griffonAppletClass}
Built-By: ${user.name}
Build-Date: dd-MM-yyyy HH:mm:ss
Griffon-Version: ${griffonVersion}
Implementation-Title:
capitalize(${griffonAppName})
Implementation-Version: ${appVersion}
Implementation-Vendor:
capitalize(${griffonAppName})
```

There might be times when you must specify additional attributes or override existing ones. You can do this by adding a new block of configuration to `BuildConfig.groovy`, for example

```

griffon {
  jars {
    manifest = [
      'Foo': 'Bar'
      'Built-By': 'Acme'
    ]
  }
}

```

### Merge duplicate files when packaging

There's a high chance of some files to have duplicates, e.g. griffon-artifacts.properties if you have installed a plugin that provides MVC groups. It's possible to instruct the build to merge duplicate files by specifying a regular expression and a merging strategy. The following table explains the different merging strategies available

| Strategy              | Description  |
|-----------------------|--|
| Skip                  | Do not perform any merge. Duplicate is discarded.  |
| Replace               | Duplicate is preferred and overwrites previous.  |
| Append                | Duplicate is appended at the end of previous.  |
| Merge                 | Common lines found in duplicate are discarded. New lines found in duplicate are appended at the end. |
| MergeManifest         | Duplicate keys override the previous ones. New keys are added to the merged result.                  |
| MergeProperties       | Duplicate keys override the previous ones. New keys are added to the merged result.                  |
| MergeGriffonArtifacts | Merges artifact definitions per type.  |

You can specify merging preferences in @BuildConfig.groovy@ like this

```

griffon {
    jars {
        merge = [
            '*.xml':
org.codehaus.griffon.ant.taskdefs.FileMergeT
ask.Replace
        ]
    }
}

```

This setting will overwrite any XML file found in the path with the last version encountered as jars are processed. The griffon build defines a set of default mappings, which are the ones found in the next table

| Regexp                                | MergeStrategy         |
|---------------------------------------|-----------------------|
| META-INF/griffon-artifacts.properties | MergeGriffonArtifacts |
| META-INF/MANIFEST.MF                  | MergeManifest         |
| META-INF/services/*                   | Merge                 |
| *.properties                          | MergeProperties       |

Merging preferences must be defined from the most specific to the least. Your preferences will override any default settings.

## Runtime

### Creating and MVC group with the same mvcName

The value of the mvcName parameter must be unique otherwise a collision will occur. When that happens the application will report an exception and terminate. This behavior can be configured to be more lenient, by defining a configuration flag `griffon.mvcid.collision` in `Config.groovy`.

Accepted values are

- `warning` - reports the error but allows the application to continue. Destroys the existing group before continuing.
- `exception` - reports the error and terminates the application. this is the default behavior.

### Set multiple System props on the command line

It's now possible to define multiple System properties on the command line when invoking `run-app` for example.

These properties will be directly available when you call `System.getProperty(somePropertyName)`. Here's a sample invocation with two parameters

```
griffon -Dfoo=foo -Dbar=bar run-app
```

### Event Aware Logging Appender

There's a new application event aware logging appender that can fire events whenever a logging statement is received. Here's how it can be configured in `griffon-app/conf/Config.groovy`

```
log4j = {
    appenders {
        console name: 'stdout', layout:
pattern(conversionPattern: '%d [%t] %-5p %c
- %m%n')
        event name: 'redirect', layout:
pattern(conversionPattern: '%d [%t] %-5p %c
- %m%n')
    }
    ...
}
```

This appender triggers an event named "LogEvent" which takes 3 arguments: the logging level (as a String), the logging message and an optional throwable.

### Short-lived MVC groups

There are times when an MVC group instance must be created and discarded shortly afterwards. Griffon already sports a simple MVC lifecycle however it's easy to miss it which can lead to memory leaks. This is why a new set of methods have been added to both application and artifact interfaces. Say for example you want to create a modal dialog that is backed by an MVC group named 'display'. The model of this dialog has a title and message properties that are used to customize its looks. Heres' how it can be used with the new construct

```
withMVCGroup('display') { m, v, c ->
    m.title = 'Information'
    m.message = 'Something cool is about to
happen'
    c.show()
}
```

This code assumes the controller of this group also has an action named `show`. The group will be automatically destroyed once the dialog is dismissed or closed.

### Automatic Registration of ArtifactHandlers

In the past plugin authors were required to initialize and register their custom `ArtifactHandlers` during the `addon init` step. This is no longer the case as long as the full qualified class name of the `ArtifactHandler` is placed in the following file `griffon-app/conf/metainf/services/griffon.core.ArtifactHandler`.

### Configurable Platform Customizations

In the past, platform customizations like the handling of the About and Preferences menu in OSX, were handled internally by the Griffon runtime, giving you no chance to override or alter the default behavior. That has been changed now. Starting with this release you should be able to instruct the runtime how you want those customizations to be applied. You only need to implement the `griffon.util.PlatformHandler` interface and register your implementation. The following configuration in `Config.groovy` specifies a different handler for `macosx`:

```
platform {
    handler = [
        macosx:
        'com.acme.MyMacOSXPlatformHandler'
    ]
}
```

Now you only need to create such handler, like this:

```

package com.acme

import griffon.core.GriffonApplication
import griffon.util.PlatformHandler

class MyMacOSXPlatformHandler implements
PlatformHandler {
    void handle(GriffonApplication app) {

System.setProperty('apple.laf.useScreenMenuB
ar', 'true')

        ...
    }
}

```

The following platform keys are recognized by the application in order to locate a particular handler: linux, macosx, solaris and windows.

## New AST Transformations

It's possible for non-artifact classes to participate in the MVC group mechanism (but not the life cycle itself) by implementing the `griffon.core.MVCHandler` interface. This task is easily achieved by annotating the class with `griffon.transform.MVCAware`. The same can be said for classes that would like to gain the capabilities of executing code using the threading facilities exposed by Griffon. The interface is `griffon.core.ThreadingHandler` and the transformation is `griffon.transform.ThreadingAware`.

## Breaking Changes

### Runtime Behavior

All of the `createMVCGroup`, `buildMVCGroup` and `withMVCGroup` methods have been moved from the `griffon.core.GriffonMvcArtifact` interface to the `griffon.core.GriffonArtifact` interface.

The signature of `mvcGroupInit(Map<String, ?>)` has been changed to `mvcGroupInit(Map<String, Object>)`.

All AST xforms have been relocated to package `griffon.transform` to align them with Groovy 1.8.0 where most are now found in `groovy.transform`; this results in the following changes

- `griffon.beans.Listener` -> `griffon.transform.PropertyListener`

- `griffon.util.EventPublisher` -> `griffon.transform.EventPublisher`
- `griffon.util.Threading` -> `griffon.transform.Threading`

The package `griffon.transform` is not auto imported by default. Usage of these AST transformations must be declared explicitly.

Both `griffon.core.GriffonApplication` and `griffon.core.GriffonArtifact` now extend `griffon.core.MVCHandler` and `griffon.core.ThreadingHandler`.

The interface `griffon.util.EventPublisher` is no longer an AST transformations (because it was relocated), it now identifies a class that can publish events using an `EventRouter`.

`griffon.util.EventRouter` moved to `org.codehaus.griffon.runtime.core.EventRouter`

`griffon.util.UIThreadHelper` moved to `griffon.core.UIThreadManager`

## Dependencies

New versions for the following dependencies

- Groovy 1.8.1
- Gant 1.9.6

## Sample Applications

Griffon 0.9.3 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.

## 0.9.3 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 0.9.3 Resolved Issues (13 issues)   |                             |   |
|---|-----------------------------|---|
| Type  | Key                         | Summary   |
|  | <a href="#">GRIFFON-409</a> | <a href="#">Compile classpath should be resolved as late as possible</a>                          |
|  | <a href="#">GRIFFON-378</a> | <a href="#">A path with a space will cause run-webstart to fail on Windows</a>                    |
|  | <a href="#">GRIFFON-379</a> | <a href="#">The template for a griffon app with java views has a bug</a>                          |
|  | <a href="#">GRIFFON-380</a> | <a href="#">Default controller template needs to import the Threading class</a>                   |
|  | <a href="#">GRIFFON-382</a> | <a href="#">Upgrade on plugins failed</a>   |
|  | <a href="#">GRIFFON-390</a> | <a href="#">Cannot close secondary windows</a>  |
|  | <a href="#">GRIFFON-391</a> | <a href="#">Smarter handling of candidate controller actions for threading injection</a>          |
|  | <a href="#">GRIFFON-393</a> | <a href="#">Use lowercase name for groups when adding the to Application.groovy</a>               |
|  | <a href="#">GRIFFON-394</a> | <a href="#">Creating a group with an existing mvcName should result in a warning or exception</a> |

|   |                             |  |
|---|-----------------------------|--|
|  | <a href="#">GRIFFON-381</a> | <a href="#">create-app archetype=jumpstart and fileType=java creates Groovy code</a> |
|  | <a href="#">GRIFFON-383</a> | <a href="#">Allow multiple java options to be set on command line</a>                |
|  | <a href="#">GRIFFON-388</a> | <a href="#">allow changing the dock icon on OS X</a>                                 |
|  | <a href="#">GRIFFON-389</a> | <a href="#">explain application icons</a>  |

## 0.9.3-beta-1 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`

| <b>Griffon 0.9.3-beta-1 Resolved Issues (18 issues)</b>                             |                             |   |
|---|-----------------------------|---|
| Type  | Key                         | Summary   |
|    | <a href="#">GRIFFON-329</a> | <a href="#">Filter plugin list by toolkit/platform</a>  |
|    | <a href="#">GRIFFON-330</a> | <a href="#">Application event aware logging appender</a>  |
|    | <a href="#">GRIFFON-336</a> | <a href="#">VerifyError with a sample method added to a controller</a>                          |
|  | <a href="#">GRIFFON-338</a> | <a href="#">Generate build.properties only when needed</a>                                      |
|  | <a href="#">GRIFFON-339</a> | <a href="#">Gradle task pdfGuide does not support caching</a>                                   |
|  | <a href="#">GRIFFON-342</a> | <a href="#">Events.groovy is not taken into account for LOC stats</a>                           |
|  | <a href="#">GRIFFON-344</a> | <a href="#">Allow all templates of an MVC group to be overridden with a single command flag</a> |
|  | <a href="#">GRIFFON-345</a> | <a href="#">Consider adding a MVC method for dealing with short lived groups</a>                |
|  | <a href="#">GRIFFON-346</a> | <a href="#">Order is not guaranteed for events posted asynchronously</a>                        |
|  | <a href="#">GRIFFON-349</a> | <a href="#">Can't run application if addons are installed in the same step</a>                  |
|  | <a href="#">GRIFFON-350</a> | <a href="#">Single jar package does not include all resources</a>                               |
|  | <a href="#">GRIFFON-352</a> | <a href="#">Automatically register ArtifactHandlers at boot time</a>                            |
|  | <a href="#">GRIFFON-353</a> | <a href="#">Add a method for handling temporal MVC groups</a>                                   |
|  | <a href="#">GRIFFON-354</a> | <a href="#">Allow closure-like event handlers in Java code</a>                                  |

|   |                             |  |
|---|-----------------------------|--|
|  | <a href="#">GRIFFON-355</a> | <a href="#">Support handling About and Preferences menu in OSX</a>                               |
|  | <a href="#">GRIFFON-357</a> | <a href="#">@EventPublisher applied to a Controller collides with automatic thread injection</a> |
|  | <a href="#">GRIFFON-358</a> | <a href="#">CreateApp uses the wrong app name when a package is specified</a>                    |
|  | <a href="#">GRIFFON-359</a> | <a href="#">Remember the name of the package used in create-app</a>                              |

## 0.9.3-beta-2 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`

| <b>Griffon 0.9.3-beta-2 Resolved Issues (14 issues)</b>                             |                             |  |
|---|-----------------------------|--|
| Type  | Key                         | Summary  |
|    | <a href="#">GRIFFON-360</a> | <a href="#">Single jar package does not merge special files</a>  |
|    | <a href="#">GRIFFON-285</a> | <a href="#">Add back missing GDSL files to griffon-cli</a>   |
|  | <a href="#">GRIFFON-361</a> | <a href="#">Rework DSL support for IDEA</a>  |
|  | <a href="#">GRIFFON-362</a> | <a href="#">Add support for Eclipse's DSLD</a>   |
|  | <a href="#">GRIFFON-363</a> | <a href="#">SwingPad has wrong dependency version on gfx-builder</a>   |
|  | <a href="#">GRIFFON-364</a> | <a href="#">Thread injection does not work for controller actions that make use of the default parameter</a>         |
|  | <a href="#">GRIFFON-365</a> | <a href="#">[SwingPad] Cannot run the application due to an unexpected error when fetching an image</a>              |
|  | <a href="#">GRIFFON-366</a> | <a href="#">Switch default imports feature to CompilerCustomizers provided by Groovy 1.8.0</a>                       |
|  | <a href="#">GRIFFON-368</a> | <a href="#">'prefix' for addons ignored</a>  |
|  | <a href="#">GRIFFON-369</a> | <a href="#">Variables written to binding in Event NewInstance will be thrown away if it is an instance of Script</a> |
|  | <a href="#">GRIFFON-370</a> | <a href="#">Allow the manifest of the app's jar to be configured with external properties</a>                        |
|  | <a href="#">GRIFFON-372</a> | <a href="#">Allow platform customizations to be extenarilly configured</a>   |
|  | <a href="#">GRIFFON-375</a> | <a href="#">Java based View template does not</a>  |

|   |                             |  |
|---|-----------------------------|--|
|   |                             | <a href="#">work in applet mode</a>  |
|  | <a href="#">GRIFFON-371</a> | <a href="#">Launching the application in applet mode displays the wrong name in the menu bar (OSX)</a> |