

Home

What is Griffon

Griffon is an application framework for developing desktop applications in the JVM, with Groovy being the primary language of choice. Inspired by Grails, Griffon follows the Convention over Configuration paradigm, paired with an intuitive MVC architecture and a command line interface. Griffon also follows the spirit of the Swing Application Framework (JSR 296), it defines a simple yet powerful application life cycle and event publishing mechanism. Another interesting feature comes from the Groovy language itself: automatic property support and property binding (inspired by BeansBinding (JSR 295)), which makes creating observable beans and binding to their properties a snap! As if property binding was not enough Groovy's SwingBuilder also simplifies building multi-threaded applications, say goodbye to the ugly gray rectangle (the bane of Swing apps)!

Grails developers should feel right at home when trying out Griffon. Many of Grails' conventions and commands are shared with Griffon. Granted, Swing is not the same as HTML/GSP but Builders simplify the task of creating the UI.

Seasoned Java developers will also be able to pick up the pace quickly, as the framework relieves you of the burden of maintaining an application structure, allowing you to concentrate on getting the code right.

The Griffon framework is extensible via [Plugins](#). There are many to choose from. For example you'll find plugins for 3rd party Swing components like [Swingx](#), [Jide](#) and [Macwidgets](#); [persistence](#) related plugins like [DataSource](#), [GSQL](#), [Hibernate](#) among others; 3D graphics and animation support is possible via [JOGL](#), [LWJGL](#) and [Processing](#). And many more!

Griffon also supports additional UI toolkits like [JavaFX](#) and [SWT](#).

Where do I start?

- [Download](#) - links to the latest stable release, you can also find previous releases
- [Documentation](#) - links to other projects related to Griffon ([Builders](#) and [Plugins](#) for example) and other documentation sources

Griffon in Action

***Griffon in Action* is available in eBook and printed formats since June 4, 2012. You can grab a copy from your preferred book retailer however note that you can get a copy at 37% discount when buying directly from <http://mannning.com>.**

Griffon in Action is a comprehensive tutorial written for Java developers who want a more productive approach to UI development. In this book, you'll immediately dive into Griffon. After a Griffon orientation and a quick Groovy tutorial, you'll start building examples that explore Griffon's

high productivity approach to Swing development. One of the troublesome parts of Swing development is the amount of Java code that is required to get a simple application off the ground.

Latest News

Andres Almiray posted on May 12, 2013 **Griffon 1.3.0 released!**

The Griffon team is happy to announce Griffon 1.3.0, the latest release of the 1.x series.

Griffon is an application framework for developing desktop applications in the JVM, with Groovy as the primary language of choice.

To begin with, some of the buildtime dependencies have been upgraded (including the latest Groovy 2.1.3 release). There are some enhancements to the plugin uninstall sequence. you can also obtain a fine grained dependencies/plugins graph reports (a la Gradle). On the runtime side of things you'll find more PropertyEditors available by default, better support for Groovy based i18n bundles and a totally new feature: action interceptors.

Full listing of bugs fixed in this release can be found at [<http://jira.codehaus.org/secure/ReleaseNote.jspa?projectId=11833&version=19049>]

Full release notes are located [here](#).

You can download the Griffon distribution from the [download page](#).

Thanks to all who contributed to this release!

Keep on Groovying!

Andres Almiray posted on Jan 11, 2013 **Griffon 1.2.0 released!**

The Griffon team is happy to announce Griffon 1.2.0, the latest release of the 1.x series.

Griffon is an application framework for developing desktop applications in the JVM, with Groovy as the primary language of choice.

To begin with, some of the buildtime dependencies have been upgraded, most notable Groovy 2.0! That's right, you can now take advantage of static type checking and static compilation in Griffon applications; however don't apply static type checking nor static compilation AST transformations to View scripts, as they are highly dynamic.

There are some new commands now: list-templates, upload-release, wrapper and usage-stats. The last one receives special attention as it enables the Griffon team to tweak future releases and make informed decisions based on your help. This command sends a small set of data based on how Griffon is being used in projects. We

value your feedback and privacy, that's why this feature is disabled by default. Rest assured that all information is anonymized and will be used with care. More information about the data collected by this feature can be found in the [Griffon Guide](#).

DSL support for both IntelliJ IDEA and Eclipse has been greatly improved on the core and swing packages.

Turning our attention to runtime aspects, services have received a huge lift with the addition of a pair of life-cycle hooks. Services are treated like singletons by the Griffon runtime, which means you needn't instantiate them, however that limits the configuration that can be applied to them during their instantiation. No more, with the addition of `serviceInit()` and `serviceDestroy()` you're in full control of what happens when a Service gets instantiated or destroyed.

Services may also have some of their properties be configured in external configuration files (like `Config.groovy`) by means of a DSL, for example the host and port properties of the following service



may be configured in `Config.groovy` like so



Additionally, a new event will be triggered when managed instances (such as MVC members and services) get destroyed. This event is named `DestroyInstance`, it's the counterpart of `NewInstance`.

Developers are highly encouraged to update to latest versions of the Spock, Spring and Eclipse-support plugins as there were some changes in core that affect them directly.

However the news don't end here, this release finally enables the usage of the [Griffon plugin](#) for Gradle. With this plugin you'll be able to adapt a Griffon project as a Gradle project, making it trivial for it to participate in a multi-project build for example.

Full listing of bugs fixed in this release can be found at [\[http://jira.codehaus.org/secure/ReleaseNote.jspx?projectId=11833&version=18431\]](http://jira.codehaus.org/secure/ReleaseNote.jspx?projectId=11833&version=18431)

Full release notes are located [here](#).

You can download the Griffon distribution from the [download page](#).

Thanks to all who contributed to this release!

Keep on Groovying!

[Andres Almiray](#) posted on Sep 20, 2012
[Griffon 1.1.0 released!](#)

The Griffon team is happy to announce Griffon 1.1.0, the latest release of the 1.x series.

Griffon is an application framework for developing desktop applications in the JVM, with Groovy as the primary language of choice.

This release brings new functionality that should make working with i18n content and resources a much pleasant experience. To begin with, the i18n plugin has been reworked and incorporated into core inside the griffon.core.i18n package. Every Griffon application is now i18n aware, messages by a (configurable) MessageSource. The Griffon Guide includes a new chapter on the subject.

Paired with i18n we get another set of APIs: resource management and injection. It's now possible to define resources such as images, icons, colors that should become available on a Model class, but instead of putting them on code you define them on external external resource files. These resource files happen to be Locale aware, exactly like their messages (i18n) counterparts. Resources may be injected programmatically or automatically by annotating properties/fields with @InjectedResource. The Griffon Guide also includes a brand new chapter explaining all options and formats.

Then there's the ActionManager incubated by the actions plugin. This component is capable of harvesting controller actions (whether they are closures or methods, doesn't matter) and create toolkit specific Actions that are added to builders using a naming convention. All aspects of an action (name, description, icon, etc) are handled by externalized resources (leveraging the new i18n features). Both Swing and JavaFX plugins provided custom support for their respective ActionManagers.

Be aware that some plugins should not be used anymore (i18n, i18n-support, actions) as they are no longer compatible with Griffon. The spring and the dialogs have changed to reflect the new i18n APIs, it's important that you upgrade to their latest releases when moving to Griffon 1.1.0.

Finally the [Griffon Guide](#) has been migrated to grails-doc 2.x, resulting in a more interactive experience.

Full listing of bugs fixed in this release can be found at [\[http://jira.codehaus.org/secure/ReleaseNote.jspa?projectId=11833&version=18430\]](http://jira.codehaus.org/secure/ReleaseNote.jspa?projectId=11833&version=18430)

Full release notes are located [here](#).

You can download the Griffon distribution from the [download page](#).

Thanks to all who contributed to this release!

Keep on Groovyng!

[Andres Almiray](#) posted on Jul 09, 2012

Griffon 1.0.1 released!

The Griffon team is happy to announce Griffon 1.0.1, the first maintenance release of the 1.0.x series.

Griffon is an application framework for developing desktop applications in the JVM, with Groovy as the primary language of choice.

The team did it best to get 1.0.0 out of the door on time, unfortunately some bugs managed to scape the last sweep before closing the release. This is a bug-fix only release, no new features have been added. Particularly applet and webstart modes did not work correctly when launching applications. Griffonsh also had trouble running applications

though all other commands ran fine.

Dependencies for all sample applications have been upgraded too. If you encounter a problem running any of the sample applications caused by an incompatible plugin version the please read the following entry to find out how to overcome the situation: http://www.jroller.com/aalmiray/entry/the_griffon_trove_troubleshooting_plugin

Full listing of bugs fixed in this release can be found at [<http://jira.codehaus.org/secure/ReleaseNote.jspa?projectId=11833&version=18574>]

Full release notes are located [here](#).

You can download the Griffon distribution from the [download page](#).

Thanks to all who contributed to this release!

Keep on Groovyng!

Andres Almiray posted on Jun 13, 2012

Griffon 1.0.0 released!

The Griffon team is happy to announce **the final release of Griffon 1.0.0!**

Griffon is an application framework for developing desktop applications in the JVM, with Groovy as the primary language of choice.

It took some time but the team has finally taken the required steps to make Griffon go the distance.

This release adds support for framework plugins. These are plugins that can be installed at the framework level. A good example is the [Git](#) plugin which lets you submit & tag plugin/archetype projects to Git repositories. There are also two new sample applications that show Griffon and JavaFX working together. The first one is GroovyFXPad, an exploratory tool for JavaFX in the same way as SwingPad is for Swing. The second one is FxBrowser which is a trivial webbrowser based on JavaFX's WebView component.

A list of bugs fixed in this release can be found at [<http://jira.codehaus.org/secure/ReleaseNote.jspa?projectId=11833&version=17846>]

Full release notes are located [here](#).

You can download the Griffon distribution from the [download page](#).

Thanks to all who contributed to this release!

Keep on Groovyng!

Andres Almiray posted on Apr 04, 2012

Griffon 0.9.5 released

The Griffon team is happy to announce the release of Griffon 0.9.5!

Griffon is a Grails inspired framework for writing RIAs on the JVM using Groovy.

A list of bugs fixed in this release can be found at [<http://jira.codehaus.org/secure/ReleaseNote.jspa?projectId=11833&version=18365>]

Full release notes located [here](#).

This release fixes all bugs reported for 0.9.5-rc2, there are no new features others than those introduced by 0.9.5-rc1 and 0.9.5-rc2.

Additionally, there are more than 120 plugins already published at <http://artifacts.griffon-framework.org> with more to come as we continue migrating plugins from the old legacy repository.

You can download the Griffon distribution from the [download page](#).

Thanks to all who contributed to this release!

Keep on Groovying!

PS: Can't wait for an issue to be fixed? Have an idea for a cool feature? Fork us at [github](#) and send us a pull request :-)

Andres Almiray posted on Mar 08, 2012

Griffon 0.9.5-rc2 released

The Griffon team is happy to announce the release of Griffon 0.9.5-rc2!

Griffon is a Grails inspired framework for writing RIAs on the JVM using Groovy.

A list of bugs fixed in this release can be found at [<http://jira.codehaus.org/secure/ReleaseNote.jspa?projectId=11833&version=18330>].

Full release notes located [here](#).

This release fixes some issues found in the build system after the big overhaul we did in the last release. Offline mode is finally available. When engaged all dependencies will be resolved against the local caches, also no remote nor legacy artifacts repositories will be reachable, only local ones.

Speaking of artifact repositories, we'd like to announce the immediate availability of the new Griffon Artifact Portal located at <http://artifacts.griffon-framework.org>. The code that runs this portal is Open Source and can be obtained from <https://github.com/griffon/griffon-artifact-portal>; and as a matter of fact is the only way for now to setup a remote artifact repository. This portal contains a minimum set of features that we wanted to expose for the first release. Rest assured we'll update it as time goes along.

The Griffon runtime has a few new features too. It's now possible to disable the application's event bus for a while, useful to avoid an event storm for example. Also, MVC groups have the capabilities to skip firing MVC lifecycle events, as well as instantiation events.

Finally, all threading methods have been renamed. this is an important step into providing better threading options as the previous method names were confusing for many. The old method names are still available but have been marked as deprecated; they will be removed by the time Griffon 1.0 is released so take note and update your code as soon as possible.

You can download the Griffon distribution from the [download page](#).

Thanks to all who contributed to this release!

Keep on Groovying!

PS: Can't wait for an issue to be fixed? Have an idea for a cool feature? Fork us at [github](#) and send us a pull request :-)

Andres Almiray posted on Aug 15, 2011

Griffon 0.9.3 released

The Griffon team is happy to announce the release of Griffon 0.9.3!

Griffon is a Grails inspired framework for writing RIAs on the JVM using Groovy.

A list of bugs fixed in this release can be found at <http://jira.codehaus.org/secure/ReleaseNote.jspa?projectId=11833&version=17427>.

Full release notes located [here](#).

This release is the culmination of the 0.9.3 series (we had two betas released in the past months). In particular Java support has been vastly improved.

It's now possible to write a Griffon application in 98% Java. All artifacts, lifecycle handlers and event handlers can be written in Java (even another alternative JVM language like Scala, Mirah and Clojure for that matter). Configuration files remain in Groovy source form. Additionally, the jumpstart archetype introduced in the betas is fully aware of Java source code now.

IDE support has improved as well. Griffon includes GDSL and DSLD descriptors for both IDEA and Eclipse.

In the runtime aspect, the size of the core library has been reduced by roughly 25% while new features were added. Giving you more bang for your bandwidth. Plugin authors will benefit from automatic registration of artifact handlers should they decide to create a new artifact definition. Specific platform customizations, like the handling of the unified menubar in OSX, are now easier to make.

There's a new set of AST transformations too. These transformations will inject the required code for making a class an MVC group handler or a threading handlers, like if it were a Griffon artifact.

Lastly, there's a number of breaking changes explained in the [release notes](#).

You can download the Griffon distribution from the [download page](#).

Thanks to all who contributed to this release!

--The Griffon Team