

# PivotProposal

## Abstract

Pivot (<http://pivot-toolkit.org>) is an open-source platform for building rich internet applications in Java.

## Proposal

Pivot combines the enhanced productivity and usability features of a modern RIA toolkit with the robustness of the Java platform. Pivot applications are written using a combination of Java and XML and can be run either as an applet or as a standalone, optionally offline, desktop application.

Like other modern development platforms, Pivot provides a comprehensive set of foundation classes that together comprise a "framework". These classes form the building blocks upon which more complex and sophisticated applications can be built.

Pivot was designed to be familiar to web developers who have experience building AJAX applications using HTML, CSS, and JavaScript. However, it provides a much richer set of standard widgets than HTML, and allows developers to create sophisticated user experiences much more quickly and easily. Pivot will also seem familiar to Swing developers, as both Swing and Pivot are based on Java2D and employ a model-view-controller (MVC) architecture to separate component data from presentation. However, Pivot includes additional features that make building modern GUI applications much easier, including declarative UI, data binding, effects and transitions, and web services integration.

## Background

The web has become the defacto standard method for application delivery. However, functional requirements for web applications have begun to scale beyond the capabilities of the browser. Even with the addition of scripting support, dynamic element manipulation, and asynchronous server communication, it is difficult to create a user experience in HTML that is truly on par with that of a desktop application.

Rich internet application (RIA) development platforms are a means of bridging the gap between the web and desktop experiences. Using browser plugins, these platforms allow developers to build applications that look and feel more like native desktop applications but are deployable via the web, like traditional, HTML-based web applications. RIAs also often incorporate visual effects intended to enhance the overall user experience, such as animations and other dynamic behavior.

Adobe Flex (<http://www.adobe.com/products/flex>) and Microsoft Silverlight (<http://www.microsoft.com/silverlight>) are arguably the most high-profile of these platforms; others include OpenLaszlo (<http://www.openlaszlo.org>) and Curl (<http://www.curl.com>). Pivot itself falls into this category.

## Rationale

Pivot was created for two primary reasons:

1. To provide a viable option for developers who want to build rich client applications in Java. Flex applications are written in ActionScript, an ECMAScript variant; Silverlight applications can be written in either C# or JavaScript; OpenLaszlo applications are written in JavaScript. Pivot allows developers to write rich internet applications in Java (or any other language that can run in a JVM).
2. Provide a freely-available, open source alternative for RIA developers. Flex, Silverlight, and Curl are all proprietary platforms. We believe that a large part of HTML's success was its due to its openness. While we certainly hope that developers will use Pivot to build revenue-generating products and applications, we believe that the platform itself should be free and driven by its technological merits, not by corporate objectives.

## Comparison to Other Java-Based RIA Platforms

### Swing

While it is technically feasible to build an RIA in Java using the Swing toolkit (<http://java.sun.com/docs/books/tutorial/uiswing>), Pivot offers a number of advantages that make it a more compelling, modern alternative:

- Provides an XML markup language called WTKX for simplifying user interface construction. Flex, Silverlight, and OpenLaszlo all offer a similar feature; web developers are comfortable with the markup metaphor, and it can considerably reduce overall development time.
- Components are not limited to an "atomic" preferred size; they are allowed to report a preferred size as constrained by either width or height - this facilitates such features as label wrapping, which Swing does not support.
- Defines a consistent data model that is used throughout the entire framework; for example, JSON data returned from a REST service is serialized into the same data structures used by a table view component to present data. No additional translation is necessary, which can significantly improve performance. A common data model also reduces the learning curve for new developers.
- Includes built-in support for REST-based data services, which Pivot calls "web queries". This provides parity with Flex, which comes with out-of-the-box support for RPC via the AMF protocol, and Silverlight, which supports both SOAP and REST-style services. Swing does not include any built-in facilities for server communication, making it less convenient to work with.

Note, however, that Pivot is not limited to REST for server communication. Because it runs in a JRE, a Pivot application can take advantage of any client/server protocol that has a Java API; for example:

- SOAP-based services via Axis (<http://ws.apache.org/axis/>)
  - Flex RPC using the BlazeDS AMF client (<http://opensource.adobe.com/wiki/display/blazeds/Java+AMF+Client>)
  - Google Contacts via the Google-provided Java client ([http://code.google.com/apis/contacts/docs/2.0/developers\\_guide\\_java.html](http://code.google.com/apis/contacts/docs/2.0/developers_guide_java.html))
- The current source includes examples of the latter two.

- Includes built-in data binding support, which allows data returned from web queries (as well as other types of data services) to easily be mapped to form contents.
- Includes platform-level support for visual effects and transitions (i.e. animations).
- Defines a single Application interface that is used for deploying both desktop and web-based applications - multiple codebases are not required.
- Takes advantage of newer Java language features such as generics, enums, for..each, and annotations.

## JavaFX

Pivot differentiates itself from the recently-released JavaFX primarily by allowing developers to build applications in Java, rather than the new JavaFX scripting language. Additionally, JavaFX's widget support is based on Swing, which suffers from the limitations outlined above.

In a sense, Pivot represents what we think Sun should have done instead of JavaFX.

## Google Widget Toolkit (GWT)

While GWT allows developers to use the Java language to write web-based applications, the runtime environment for a GWT application is the browser itself, not a JVM. This has a number of drawbacks:

- The compiled code executes as interpreted JavaScript, not bytecode.
- The only libraries available are those that have been ported to GWT by Google.
- All presentation must be done via CSS and DOM manipulation rather than via a true 2D drawing API.

Additionally, GWT does not support an XML markup language - all UI elements must be created programmatically.

Pivot allows developers to efficiently construct RIAs that can truly take advantage of the Java platform.

## Current Status

Pivot began as an R&D effort at VMware. It was announced as an open-source project in June of 2008 under the Apache 2.0 license. Version 1.0 was released in October, 2008, and version 1.1 is targeted for release in early 2009. Pivot is currently hosted at <http://pivot-toolkit.org> with development at <http://code.google.com/p/pivot>.

## Core Developers

Thus far, Pivot has been developed primarily Greg Brown and Todd Volkert of VMware.

## Community/Meritocracy

The Pivot team is actively striving to build a development community around the Pivot platform, consisting of both users and contributing developers. Although we are pleased with the progress we have made to date, Pivot is a large undertaking that needs the support of a larger developer base if it is to succeed and thrive.

We have written numerous articles and blog entries about Pivot to help generate interest in the developer community, and will continue to do so as new features are added and new versions released. We have established a Google Group for discussion of Pivot-related issues (<http://groups.google.com/group/pivot-toolkit>) where feedback is solicited and highly encouraged.

As developers (hopefully) begin to express an interest in using and helping to expand Pivot, their contributions will be evaluated and incorporated as appropriate. Several contributions from several external developers were actually accepted before we released version 1.0 - notably, both the Pivot Explorer tool and the `pivot.core.util.Resources` class used for localization were provided by outside developers who volunteered to join the project after reading our initial announcement.

## Alignment

We use Ant to build and deploy Pivot, and we host our web-based components in Tomcat.

The Apache developer community is representative of our target audience - developers who seek out and advocate high-quality, open-source software projects. We consider ourselves to be members of that group, and we are hoping to share our work with other like-minded developers.

## Known Risks

### Orphaned Products

This project will not be orphaned. The developers have already invested a significant amount of time in the project, much of it in off-hours. Greg has been working on Pivot since August 2007 and Todd since January 2008. We pay to host the project site, [pivot-toolkit.org](http://pivot-toolkit.org), out of our own pockets. Additionally, at least one production application is currently being developed with Pivot. We want to see this project become successful, and we will continue to invest whatever time is necessary to help make it so.

### Inexperience with Open Source

Pivot has been conceived as an open-source project since day one. However, though we are longtime users of open source software and have some understanding of the process, we have no prior experience managing an open-source effort. We are hoping that the community experience within ASF can help bring our understanding to the next level.

## Homogenous Developers

While Pivot's primary developers work for VMware in Boston, Massachusetts, Pivot is not a VMware project. Most Pivot development takes place outside of work hours.

Pivot is a highly-modularized framework that lends itself well to heterogeneous development. We have already received code submissions from developers elsewhere in the U.S., Germany, Italy, Canada, and the U.K. We are hoping that, as the capabilities of the framework expand, it will lead to an increase in overall developer interest and an acceleration of the development process.

## Reliance on Salaried Developers

Pivot began as an off-hours side project. It was officially supported by VMware for a short time, but was dropped in June, 2008 when the company decided to pursue an off-the-shelf UI solution instead of a homegrown one. It has continued as an off-hours project since then.

The "lack" of an ability to rely on salaried developers for development may actually be more of a risk to the project. Most work must be done in off-hours, which may occasionally limit our ability to make progress.

## Documentation

Additional information is available from the project site: <http://pivot-toolkit.org>.

## Initial Source

Source code for Pivot is currently hosted at Google Code: <http://code.google.com/p/pivot>.

## Source and Intellectual Property Submission Plan

VMware released Pivot as an open-source project under the Apache 2.0 license in June, 2008 but retains a copyright on the code. VMware has agreed to sign a software grant for the codebase; a signed copy of the agreement was faxed to the ASF on 12/22/2008.

## External Dependencies

Pivot is a Java-based toolkit and relies on a JRE for execution. Pivot's charting library currently relies on JFreeChart (<http://jfree.org>) 1.0.9; however, charting is an optional package and JFreeChart is not integral to the implementation (i.e. it could be replaced with another charting package with no impact to the Pivot charting API).

## Required Resources

- Subversion repository
- Issue tracking
- Dev., user, commits, and private mailing lists
- Wiki space
- Web hosting (for content currently hosted at pivot-toolkit.org)
- SSL certificate (for code signing)

## Initial Committers

- Greg Brown, Todd Volkert - VMware (<http://vmware.com>)
- Eugene Ryzhikov - Florida Power and Light (<http://www.fpl.com>)
- Christopher Brind - unknown
- Sandro Martini - Technomind (<http://www.technomind.it>)

## Affiliations

No relevant affiliations.

## Sponsors

- Champion: Niclas Hedhman
- Nominated Mentors: Niclas Hedhman, Martijn Dashorst
- Sponsoring Entity: Incubator PMC