User Interface as a Service

visualforce
Contents

Executive Summary ........................................................................................................................................1
Deploy Any Application, on Any Device ..............................................................................................1
Developing with Visualforce ..................................................................................................................2
Working with Visualforce ......................................................................................................................2
Visualforce and Apex Code ....................................................................................................................2
Getting Started .......................................................................................................................................3
Executive Summary
Visualforce is a key component of Force.com, salesforce.com’s platform as a service (PaaS). With Visualforce, companies and their developers can create any kind of user experience and deliver it entirely on demand, without software.

By leveraging the database, integration, and logic capabilities at the core of Force.com, Visualforce provides complete control over the user interface of any applications build on the platform, so that it can be tailored for any requirement, user, or device. With Visualforce, the power of the PaaS model can extend to virtually any application and business requirement.

Visualforce lets developers use standard Web development technologies—including HTML, AJAX, and Adobe Flex—to create user interfaces for their on-demand applications. Through a modern Model-View-Controller (MVC) model, these interfaces can be wired together with Apex code—salesforce.com’s on-demand programming language. As a result, Visualforce offers not only pixel-level control over the definition of an interface, but makes it possible to create new and sophisticated user interactions based on wizards, branching, and other UI logic.

With Visualforce, Force.com is now on par as a development environment with any of the application servers or database tools traditionally used for application development. Many salesforce.com customers are using Force.com to customize existing applications and create new on-demand applications, both for CRM and to support other business requirements. This new, Internet-based model of creating and running applications on demand lets developers focus on productivity, not plumbing. For their companies, that means a better return on their IT investment.

Deploy Any Application, on Any Device
As the Internet and associated Web standards become ubiquitous—appearing not only on virtually every desktop, but increasingly on every device—the opportunities for delivering applications on demand and as a service are growing as well. Tools such as Visualforce support this trend by enabling user interfaces that support a growing set of contexts from which users access these applications.

For example, approaches to on-demand applications may range from simply customizing an order entry form in an existing CRM application to creating entirely new applications that will run on kiosks or mobile devices such as an iPhone. In each of these use cases—either extending the current Salesforce look and feel or creating a totally new user interface—Visualforce provides the features needed to make those tasks as easy as possible.

AT a high level, examples of applications well suited to Visualforce include:

- **CRM customization.** With over 25 predefined user interface components, Visualforce makes it easy to extend the standard Salesforce look and feel to include new kinds of user interactions, such as wizards and multi-object screens. Instead of implementing custom cascading style sheets (CSS) to emulate the Salesforce interface, developers simply invoke common components such as page headers, fields, and related lists via special tags.

  Just as useful is the capability to selectively override specific interface pages—such as a Contact edit page or a custom tab’s home page—while leaving the standard interface intact for all other functions. This fine-grained control over an application’s flow provides the best of both worlds, so that developers can choose the automatically generated user interface for some areas while using Visualforce to selectively modify others.

- **New Web and database applications.** Although there any kind of application that can be built on the database, integration, and logic core components of the Force.com platform, not all applications fit salesforce.com’s default, out-of-the-box “database browsing” metaphor. For such applications, Visualforce provides the full page control needed to create different kinds of metaphors. The resulting applications may look and behave differently from Salesforce CRM applications in every way, but they can share the same service delivery infrastructure and other services of the Force.com platform.

- **New Devices.** The full control provided by Visualforce is also ideal for bringing on-demand applications to a new generation of Internet-enabled devices. By being able to shape the presentation of data to a specific context—for example, by making it appear in one format when viewed on an iPhone and in another in a desktop Web browser—Visualforce makes it possible not just to deliver the right application to the right device, but to deliver a single application to multiple devices, with each device providing different views into the same data.
Developing with Visualforce

With Visualforce, salesforce.com created the first on-demand implementation of an MVC architecture—a widely used interface architecture based on the separation of data presentation from data manipulation. In such an architecture, the “model” represents the data model, the “view” represents the presentation of the data (user interface), and the “controller” represents the business logic that manipulates the data and controls the user interface.

In Visualforce, the MVC model is implemented with standard and custom objects—a staple of Force.com development—as well as with three new objects: pages, components, and controllers.

:: **Pages**, along with components, are the basic creative building blocks for application designers. Similar to a standard Web page, a Visualforce page uses HTML to specify the appearance of the application’s interface, with the option of using other Web technologies such as CSS, AJAX, and Adobe Flex for additional flexibility. Pages are referenced and invoked via a unique URL, just as they would be on a traditional application server.

This standard markup is complemented by special Visualforce components. These components, which are similar to tag libraries in other systems, make it possible to invoke complex components with a single line of HTML. Because these components are composed and rendered on the server and then delivered to the client, this approach provides better performance and enhanced functionality compared to client-only techniques.

:: **Components**, which are invoked with special HTML tags, are the key for reusing common interface elements and for binding both standard and custom elements to data. Visualforce initially ships with 25 predefined components that can be assembled with minimal coding, in building-block fashion. Some components implement common Salesforce interface elements, while others make available new features such as AJAX-based partial page refreshes. Components may provide various levels of granularity, such as displaying multiple lines of data or executing embedded functionality of an application’s meta data on a Visualforce page. All components can be captured in style sheets, to make it easy to change a component’s style regardless of how it was created.

:: **Controllers** are the basic functional building blocks that control the application’s logic. Implemented in Apex code, controllers provide the underlying business rules for the interface, as well as the “connective tissue” between the application’s page presentation and the underlying data. Any given page interacts with a given controller through components, which bring in the data to be displayed in the interface and send it out to be stored in the database.

The controller provides access to the data and specifies what happens when the user interacts with an interface component. Visualforce provides pre-built, standard controllers for standard interactions such as View, Edit, and Save, which can be implemented without additional coding. New behaviors that go beyond these predefined interactions—called custom controllers—can be programmed in Apex to access new data sources, navigation elements, and behaviors. Because custom controllers can maintain state across page interactions, it is possible to construct interactions such as wizards, whose logic spans steps on multiple pages.

Working with Visualforce

Visualforce markup defines which interface components are included on the page and what those components look like. Because Visualforce markup is ultimately rendered into HTML, designers can use Visualforce tags alongside standard HTML, JavaScript, Adobe Flex, or any other code that can execute within an HTML page.

Because developers can edit an application’s markup in the same window that shows how the page will be displayed, they can instantly verify how a particular edit affects the interface by simply saving the code. The Visualforce editor pane also includes auto-completion, syntax highlighting, and *quick fix* features so that developers can create components on the fly.

Visualforce and Apex Code

The introduction of Visualforce follows last year’s introduction of Apex, which gave developers a major tool for creating application business logic. Visualforce provides similar creative control over the user interface, with all the performance, maintenance, and security advantages of an on-demand service.

Visualforce and Apex complete the application development services of the Force.com platform. With these tools, developers have a full spectrum of customization and development options. They can create pages with salesforce.com’s point-and-click tools that rely strictly on the Salesforce application framework and result in
an interface with the standard Salesforce look and feel. For those who want to create a more customized or altogether different experience, there’s Apex and Visualforce—to be used either separately or together.

In this way, developers can easily combine the best of all worlds. For example, developers can use the standard Visualforce controllers to create a custom look and feel, coupled with standard application behavior. They can extend the appearance of default pages by altering the standard header and footer areas, or by overriding elements of the user interface by modifying the pages linked to one tab, but not others. Of course, developers can also create pages and behaviors that don’t resemble anything in the traditional Salesforce palette.

**Getting Started**
Announced at Dreamforce 2007, Visualforce is scheduled for general availability in an upcoming Salesforce release. However, members of the Apex Developer Network can get started immediately by enabling the Visualforce Developer Preview in their Developer Edition orgs.

For access to Visualforce and a for a free ADN membership, please go to [http://developer.salesforcece.com](http://developer.salesforcece.com). Here you’ll also find introductory and “deep dive” Dreamforce presentations about Visualforce, as well as official documentation, a “quick start” tutorial, and code samples ranging from “Hello World” to a three-page opportunity wizard.
For More Information
Contact your account executive to learn how we can help you accelerate your CRM success.