Wirehose-Dynamic Layouts

WireHose applications can have multiple user interface appearances. This can be used to support multiple branded affiliates from a single codebase, as in an application service provider environment, or to allow the user to personalize the look of their page in addition to personalizing its content. You can also use this ability to support multiple output formats, such as XML, HDML, SMIL, RSS, RDF, etc.

Layout dictionary

Each separate look in a WireHose application is called a layout. Layouts are defined in a file called a **layout dictionary**. You can provide a layout dictionary programmatically, or specify the location of a layout dictionary file on the command line. **WHSessionHelper** is the primary object used for accessing the layout dictionary.

By default, WireHose will use the value of the user's currentLayout property to determine which layout to use. It can also automatically determine which layout to use by sniffing HTTP request headers, which is useful for temporarily overriding the user's layout preference if they are using a handheld device or other non-HTML client.

Pages, wrappers and areas

Like all WebObjects applications, a WireHose application consists of pages and components. You can substitute pages and components for a given layout; the session helper uses the layout dictionary to determine which components to use. Each layout has an associated **WHWrapper**, which defines the look for that layout. Wrappers and pages also may define multiple areas on the page using **WHArea** components. For example, a three-column layout may define three areas, "left", "middle" and "right", while another layout may include only a "main" area. Channels are mapped to a particular area through their areaName property; you can map areas from one layout to another through entries in the layout dictionary.

Dynamic binding resolution

Components that inherit from **WHComponent** can optionally resolve their bindings through the layout dictionary rather than being set directly by a parent component. A component's color binding can resolve to "blue" in one area, and "green" in another, depending on the current area, page and layout.

Localization

WHSessionHelper provides methods which return localized strings depending on the session's languages array, and will look in several places to resolve the value. You can define strings for a particular component on a particular page, or for a particular component, or for a particular page. You can also provide default localizations for components which are in a framework, and those localizations can be overridden in a particular application. You can also define non-localized strings in the layout dictionary. **WHPropertyBinder** objects are used to get localized labels and values for binding to popup menus.

Cookies, URL rewriting and bookmarkable pages

WHDirectAction provides a mechanism for automatically detecting whether cookies are enabled in a client's browser, and controlling whether session IDs are visible in URLs accordingly. **WHHyperlink** is a dynamic element which includes support for URL-rewriting similar to Apache's mod_rewrite, but which is applied to URLs generated by your application. Together these two classes enable your applications to have clean, bookmarkable URLs such as " /WireHoseDemo/MyHomePage" and still provide personalized sessions.