Concept

This chapter covers the following topics:

- Layout Creation - Binding to Application
- Server Side Architecture

Layout Creation - Binding to Application

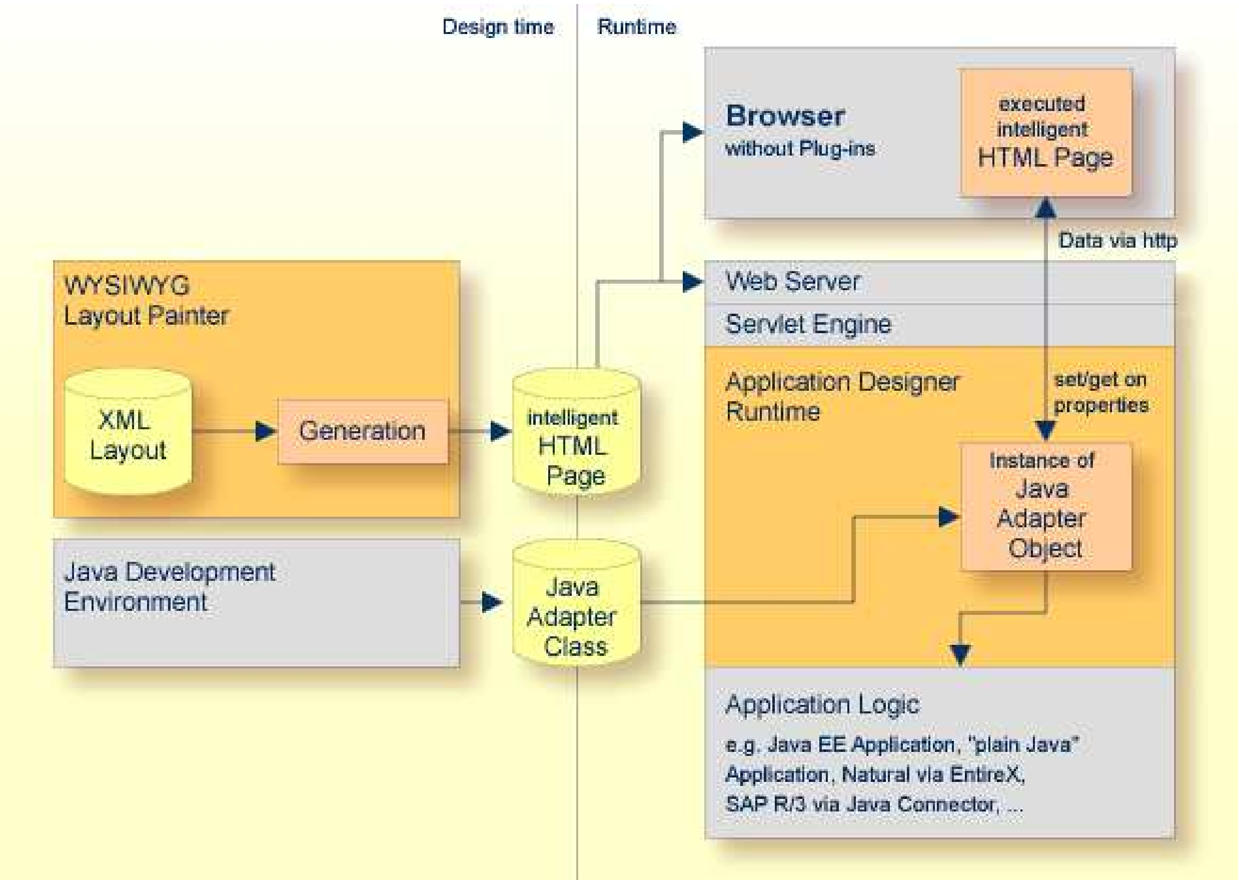
The Application Designer concept follows the commonly accepted paradigm of separating the layout (view) of an application as much as possible from the application logic itself.

To write a page (view) as a front-end to an application, you proceed as follows:

- You design the layout of the page using the WYSIWYG Layout Painter. The result is kept in an XML layout description file. The Layout Painter provides a code generator that you can use to generate an HTML page from the XML layout description. The layout definition both contains the controls used on a page and keeps information on how these controls are bound to properties and methods of a server-side Java object. The generated HTML page contains HTML and JavaScript statements - and is called "intelligent HTML page" in this documentation. The page is executed within a browser at runtime.

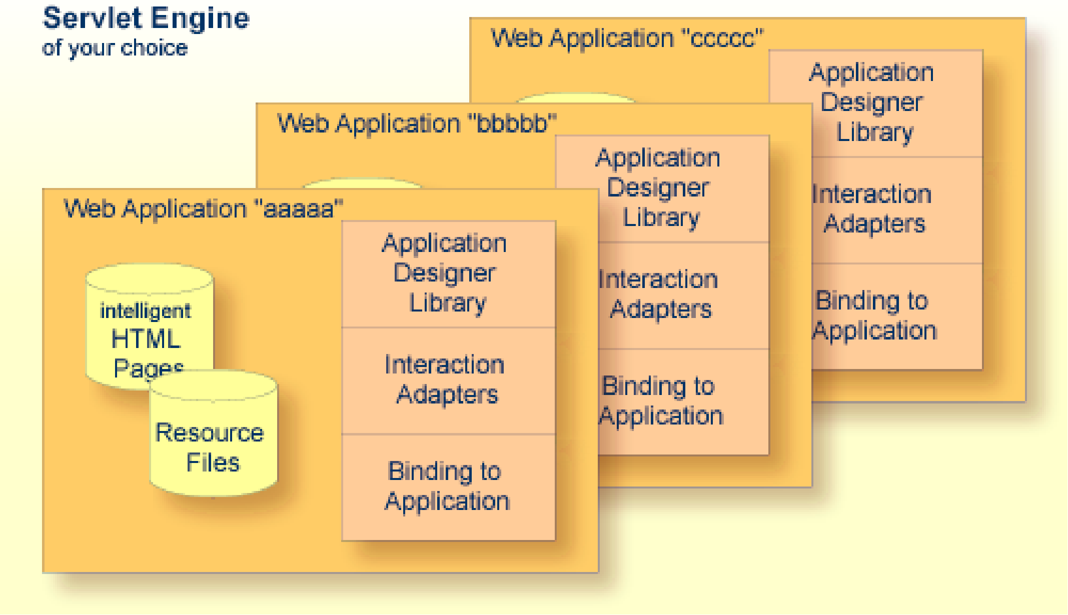
- You write a small Java class. This class handles all properties and methods defined within the layout definition. The class is executed in the Application Designer server environment, i.e. it does not run on the browser client but on the server side.

The class does not provide a large amount of logic but connects to existing application logic. This is the reason why it is called the "interaction adapter" in this documentation.



Server Side Architecture

The Application Designer server runtime is a web application conforming to the Java Servlet specification. It can be deployed multiple times to a servlet engine of your choice. XML layout definitions, generated intelligent HTML pages, adapter classes and other resource files are part of this web application.



You either install Application Designer as a web application inside your servlet engine and build your application modules around it, or you add and integrate the Application Designer environment into an existing web application.

In case of a multiple-deployment setup, each deployed instance of Application Designer is a self-contained, independent environment, completely separate from the other instances. With the example shown in the graphics above, you have three independent instances. Pages of the first instance are, for example, accessible via *http://host:port/aaaaa/...*, pages of the second instance via *http://host:port/bbbbb/...*, and so on.