Overview

This documentation provides information on the Application Designer control concept. It is recommended that you first become familiar with the "normal development" of screens inside Application Designer.

When do you need customized controls? In general there are two cases:

1. You want to combine existing controls to form complex controls with a certain dedicated task. Maybe you want to define an "address area" control which consists of a certain arrangement of fields and labels that form an address. This kind of building controls is called "composing controls" in this documentation - you take what is available and group it into certain units.

2. You want to create new controls - maybe you need some special kind of icon with a certain behavior.

While case 1 does not require to deal with JavaScript and HTML, case 2 requires knowledge of JavaScript and HTML and the use of the JavaScript library functions that are available via the Application Designer framework.

Due to the usage of XML as the layout definition format and due to an open interface for integrating control definitions into the page generation process of Application Designer, the Application Designer control concept is a flexible and open framework. Actually, all Application Designer controls are following the framework - there is no "special way" or "shortcut" that is internally used.

One part of the concept is the definition of controls, i.e. control tags with certain attributes, which you can integrate via a tag library concept into layout definitions. The other part is the binding of the control to server side adapter properties. Following the strict Application Designer architecture - that the GUI is a reflection of a "net data"/"model data", dynamic controls have to transfer their data at runtime to/from adapter properties. This binding concept is important:

- On the one hand, you want newly created controls to reference the adapter properties/methods.

- On the other hand, you want to compose controls (e.g. an address area) and want to bind them to complex objects (e.g. an address object) on the server side - already providing for a set of data and methods that fit to the control and provide some server side logic.

See *Binding between Page and Adapter* in the *Special Development Topics* for detailed information, especially about the hierarchical name binding concept (access path).