Some Background Information

This chapter covers the following topics:

- Name Binding between Controls and Adapter

- Data Exchange at Runtime

- Files and their Locations

Name Binding between Controls and Adapter

Which are the critical parts when building the "Hello World!" application?

- The NATPAGE control in the layout points to the name of the adapter object (property `natsource`).

- The FIELD control in the layout points to the property name of the adapter (property `valueprop`).

- The BUTTON control in the layout points to the event `sayHello()` of the adapter (property `method`).

There is a name binding between the layout definition and its corresponding adapter. This is the simple and effective approach of the development process: The adapter represents a logical abstraction of what the page displays. All layout definitions are kept in the page - all the logic is kept in the adapter. (Or better: behind the adapter. The adapter itself should only be a facade to the "real" application logic.)

Data Exchange at Runtime

What happens at runtime?

- When the user starts a Natural session from the logon page, the Natural program that the user specified in the command line is started.

- The Natural program executes a `PROCESS PAGE` statement, using an adapter.

- The `PROCESS PAGE` statement passes the name of the HTML page to be used and the initial page data to the browser.

- The browser displays the page. JavaScript code on the page distributes the initial data to the controls.

- The user provides some input, for example, enters the name. The content change is stored inside the page. The Natural program is not yet involved.

- The user does something which causes a flush of the changes (for example, the user chooses a button). Therefore, all registered data changes are packaged and are sent through the adapter to the Natural program, including the information which event has been raised.

- The Natural program receives the modified data.

- The system variable `*PAGE-EVENT` receives the name of the raised event.

- The event handler in the Natural program modifies the data and resends it to the page using a `PROCESS PAGE UPDATE` statement.

- And so forth.

With a standard HTTP connection, only the changed content of the screen is passed when operating on one page. The layout is kept stable in the browser. Consequently, there is no flickering of the page due to page reloading.

All steps described in the list above are done completely transparent to your adapter; i.e. you do not have to cope with session management, stream parsing, error management, building up HTML on the server, etc. You just have to provide an intelligent HTML page by defining it in the Layout Painter and an adapter object.

Files and their Locations

Have a look at the files created for your "Hello World!" application and take notice of the directory in which they are located.

All files are located in the directory *<installdir>/cisnatural/cisnatfirst*. The *<installdir>/cisnatural* directory is the directory of the web application instance. The *<installdir>/cisnatural/cisnatfirst* directory is the directory that has been created for your new project.

- The XML layout definition is kept in the *<installdir>/cisnatural/cisnatfirst/xml* directory.

- The generated HTML page is kept directly in the project directory. There are possibly also some other files inside this directory that start with "ZZZZ". These files are temporary files used when previewing pages inside the Layout Painter.

- The generated Natural adapters are kept in the directory *<installdir>/cisnatural/cisnatfirst/nat*.

- In the directory *<installdir>/cisnatural/cisnatfirst/accesspath*, "access restriction" files are generated. If you view these files inside a normal text editor (such as Notepad), you see that one file is maintained for each page; it holds the information about which properties are accessed by the page.