

Introduction

Using Natural for Ajax, you can create rich internet applications which use the Ajax (Asynchronous JavaScript and XML) technology. This enables Natural users on Windows, UNIX and mainframe platforms to develop and use Natural applications with a browser-based user interface, similar to GUI desktop applications.

This chapter covers the following topics:

- What is a Rich Internet Application?
 - Rich Internet Applications with Natural
 - Mixed Applications
-

What is a Rich Internet Application?

Classical HTML- and browser-based applications suffer from known disadvantages. The server responds to each user interaction with a new page. This may lead to long response times and new rendering in the browser and thus to a discontinuous workflow for the user. The possibilities offered by DHTML overcome these disadvantages, but they are complicated to use and make it hard to build a comfortable user interface. The user interface is therefore often simpler and less comfortable than users are accustomed to from their experience with desktop applications. Although it is possible to provide complex controls and features like drag-and-drop, this is hard to implement - especially if compatibility with all commonly used browsers is required. Classical GUI applications also have the disadvantage that a client component of the application must be installed on each client machine.

Rich internet applications that use the Ajax technology overcome these disadvantages by combining the reachability of browser-based applications with the rich user interface of GUI applications. Software AG provides support for the development of rich internet applications with Application Designer. Natural for Ajax combines the user interface capabilities of Application Designer with the application development capabilities of Natural.

Rich Internet Applications with Natural

At runtime, a rich internet application with Natural has the following structure:

- A Natural host session on a Windows, UNIX or mainframe server runs the application code. Other than with a map application, the application does not deal with user interface issues. It contains only the application logic and communicates with the user interface layer by sending and receiving data. The data is displayed in page in a web browser. Events - such as button clicks - that the user raises in the web browser are passed back to the application code. Along with an event, the application code receives also the data that the user modified in the web browser. It processes the event and the data and returns modified data back to the web browser page.
- Natural for Ajax, which is running on an application server, merges the data received from the Natural application into a DHTML page and delivers the page to the web browser. In the inverse direction, Natural for Ajax forwards events that the user raised in the web browser along with the modified data to the Natural application.

- A web browser renders the DHTML page. JavaScript code on the page processes local user interaction and exchanges data with Natural for Ajax as needed. It uses Ajax technology to exchange data with the Natural application in the background without having to re-render the page as a whole.

At development time, a rich internet application is created with Natural in the following way:

- Application Designer is used to develop the user interface layout of a web page and to bind the controls on the page to data elements in the application. Application Designer is contained in the Natural for Ajax module running on the application server.
- When the user saves the page layout, a Natural module of type "Adapter" is generated. The adapter serves as an interface between the application code and the page layout. It contains:
 - A data structure that describes the data that the Natural application has to deliver to the application server in order to populate the web page.
 - The Natural code necessary to transfer the data structure to the user interface and to receive modified data back.
 - A code skeleton, in the form of comment lines, that contains handlers for the expected events. The application programmer can copy this code skeleton into the main program to implement the event handlers.
- Then a main program is implemented that exchanges data with the web page using the adapter and handles the events. The event handler code has no knowledge of the web page layout, but operates only on the page data that is sent and received through the adapter.
- The navigation between different pages is implemented. A rich internet application navigates between pages in the same way as a map application would navigate between maps.

Mixed Applications

With the support of Unicode, Natural has introduced the Natural Web I/O Interface which renders Natural maps in a web browser. Typically, if you are running map-oriented applications and wish to change them to rich internet applications, you will do this gradually. In certain parts of an application, maps might be replaced by rich GUI pages, other parts will possibly be left unchanged. Therefore, Natural supports running mixed applications which consist of both maps and rich GUI pages. With maps, the application controls the page layout, and the rendering mechanism therefore respects the layout information that the application provides. With rich GUI pages, the application does not control the layout; the layout is controlled by Application Designer. However, for the users of an application the switch between maps and rich GUI pages is seamless.