

Introducing the Natural Web Interface

More and more organizations need to offer information or services via the Internet. Gone are the days where static HTML pages were sufficient for the daily visitors to a web page. Today, increasingly sophisticated HTML pages are competing in the web, and the demand for full access to business logic via the Internet is increasing tremendously. The database management systems containing business-critical information are mostly based on heavy-duty servers like mainframes.

This section covers the following topics:

- What is the Natural Web Interface
 - Architecture
 - Natural Web Interface Modules
 - Features
 - Functionality
 - Security
-

What is the Natural Web Interface

The Natural Web Interface is a link between a Web Server (more precisely: HTTP server) and your Natural environment. This can be on a separate server machine (such as a mainframe) or on the same machine as the HTTP server (e.g. Apache or Microsoft IIS).

Contents of web pages can easily be created dynamically by a Natural program. This is a basis for implementing a real interactive application on the web.

An interactive application enables users to input information and react by issuing output depending on that input. Examples of Web-based applications are order entry systems, travel booking services and parcel tracking systems. This considerably increases the scope of Natural applications. Not just in-house users, but also potential customers all over the world can now use the same application.

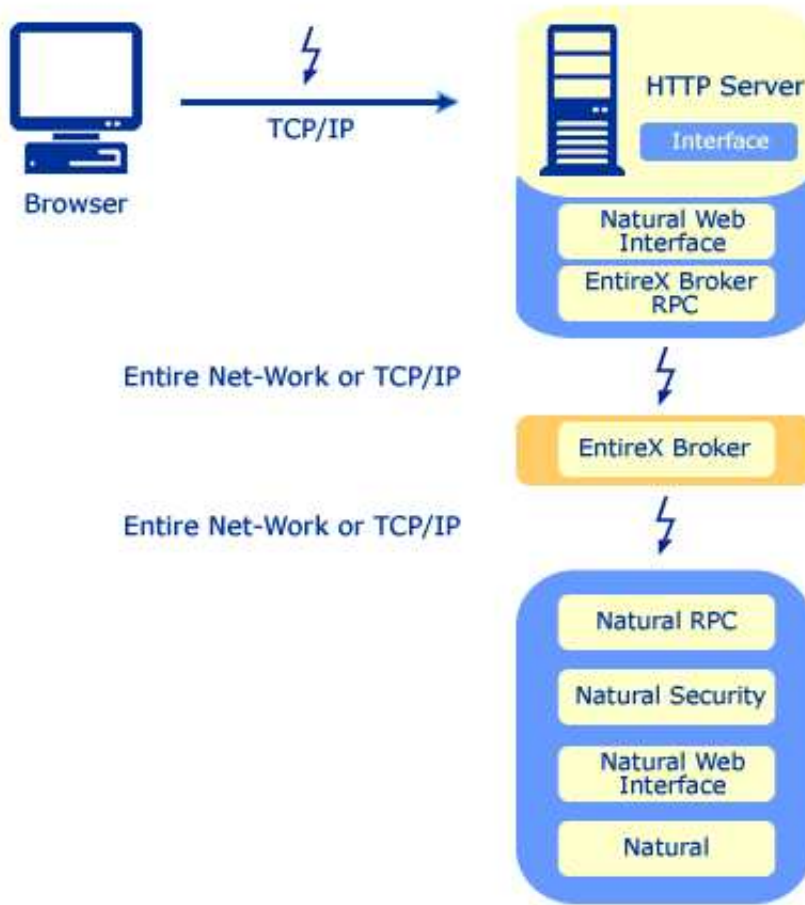
And best of all: to implement such an application, Natural users do not have to learn a new programming language. Navigation and user input/output are implemented fully in Natural (with some additional embedded HTML statements).

Architecture

The following topics are covered:

- Communication Using Natural RPC Techniques

Communication Using Natural RPC Techniques



Natural Web Interface Modules

The Natural Web Interface comprises three internal modules:

1. **Natural Web Interface**
the HTML API and the HTTP API of Natural
2. **Natural Web Server Extensions**
the part which provides the interface to the web server on the same machine
3. **Necessary middleware**
EntireX Communicator including EntireX Broker using RPC technology

Features

Calling Natural Subprograms from a Web Page

One of the main features of the Natural Web Interface is, that Natural subprograms can be called from a web page. This can be done using forms on a web page that contain input fields and buttons. Users can enter data and submit these data by clicking one of the buttons. This executes a Natural subprogram which passes the user data as parameters.

This allows easy access to application functions (= subprograms). Simple database access for retrieving data using SQL (and an ODBC driver) as offered by most Web Servers is not enough for implementing an interactive application. You also need business logic to ensure data consistency and processing of the user data.

Business logic such as consistency and plausibility checks usually already exist, as they were implemented for operational applications in the past. If they were implemented as separate Natural modules (such as subprograms, programs, or subroutines) they can easily be re-used and do not have to be re-implemented in a different environment or different language.

Therefore, no special interface program has to be written to connect the web server with the business functions. The Natural Web Interface is a standardized interface for that purpose.

No programming language has to be learned and existing skills can be leveraged (except for HTML statements to format the output pages).

Feedback to the User with a Formatted Web Page

The second important part of an interactive application on the web is the feedback to the user with formatted web pages. With Natural Web Interface these web pages can be formatted dynamically according to the application's needs.

A benefit is that the control of layout and contents of these pages is fully at the application/program level, not outside in separate directories.

And also: as Natural can gather data and information from a wide variety of sources (Adabas, RDBMSs, VSAM, sequential files, even system information with Entire System Server) the type of application is virtually unlimited - any application you can build with Natural you can integrate with the web.

Proven Middleware

The Natural Web Interface is based on the proven set of middleware products from Software AG: the Entire product family.

This allows seamless integration in an existing client/server environment. The web connection is just another client, which can be connected to existing Natural servers. If Entire Net-Work is installed, you do not need to install another set of middleware products.

On Natural for Windows, the interface can call Natural DCOM classes. The methods called, with a specific interface, can map to the same subroutines used through remote procedure call (RPC).

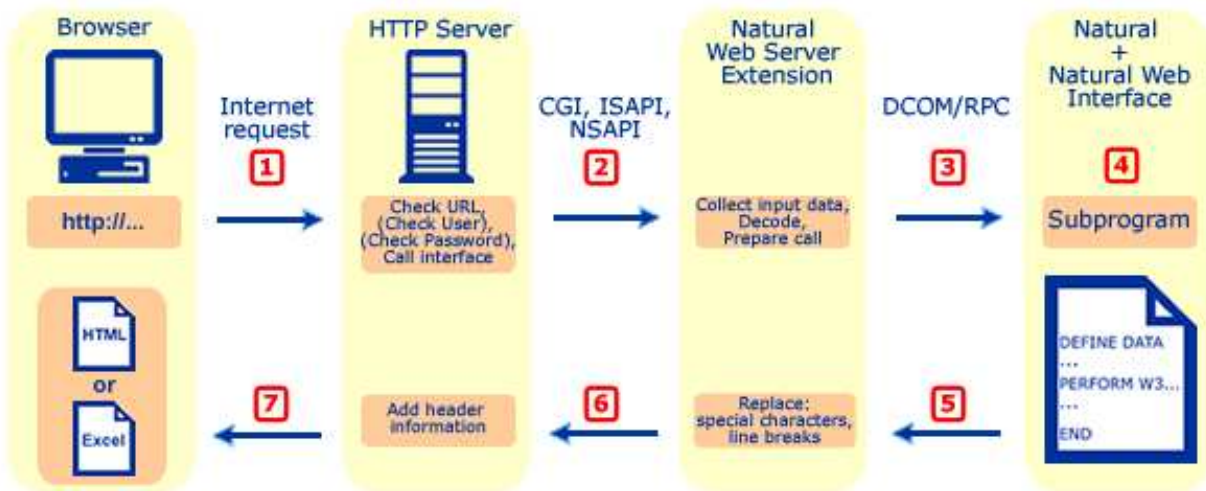
Web Page Creation

Web pages can be created with standard tools (e.g. Macromedia Dreamweaver or Microsoft Frontpage) or with the web page creation tool using the Natural generation functionality. From the Natural server, subprograms can be generated. There is no need to acquire knowledge about any other programming language or web-page creation tool.

Functionality

Requests from a web page in the user's browser are passed to the web (or HTTP) server. Provided that this was a form requesting execution of a Natural subprogram, this request is then passed to the Natural Web Server Extensions part which executes the Natural subprogram via EntireX RPC, PAL or DCOM. The program takes any user data as parameters and then issues a set of programs to provide the feedback to the user.

The following diagram illustrates how the Natural subprograms are called from an HTML browser. Each stage of the process is identified by a number; what happens at these stages is explained below.



1.

HTML Browser Requests URL.

Your browser requests a URL identifying the program you want to call on the server side.

2.

Web Server calls the Natural Web Server Extension CGI.

The web server takes the URL and calls Natural Web Server Extensions.

3.

Natural Web Server Extension converts the call to RPC.

The Natural Web Server Extension program "translates" the URL into a Natural RPC that invokes the Natural server program originally identified by the URL.

4.

Natural subprogram is executed and generates a return page.

The Natural subprogram on the server is executed and generates an HTML return page.

5.

Return Page is sent back to the Natural Web Server Extension.

The HTML return page is sent back as response of the subroutine call.

6.

Natural Web Server Extension sends back the return page to the Web Server.

The Web Server adds header information and sends it to the browser.

7. The browser receives the answer to what it was sent out as a request for an URL.

Note:

In the context of the Natural Web Interface, only external subroutines can return output.

Security

Pages called via Natural Web Interface can work together with Natural Security. This is accomplished as follows:

- First your Natural Web Server Extension has to be defined as restricted page at your HTTP server.
- If this is done, you will be prompted for user ID and password by your browser if you request a page.
- The HTTP server will now verify the given data with its database.
- If the user is authorized, Natural Web Server Extension is called with the remote user's name.
- If the Natural RPC server is started with Natural Security, the given name will be set as *USER.
- As an authentication is already done by the HTTP server, no password checking will be done on the Natural side. Therefore, the Natural RPC server has to be started with AUTO=ON.

A second scenario is that when the initialization file is started, a specific, fixed, defined user ID and password is set to communicate with a Natural RPC server with Natural Security. See also Communication with Natural Security.