Domino Designer Rnext Technical Overview

by
David
DeJean

Level: All
Works with: Rnext Preview Build 1
Updated: 01/22/2001

The vision of what constitutes a Domino application is expanding—it's not just an .NSF file anymore. Now it's data from relational sources, Java Server Pages, cascading style sheets, layered pages. Domino Designer is expanding to keep pace. That is what Designer Rnext is all about.

"Designer is an integrated development environment," says Martha Hoyt, Lotus Senior Manager for Collaborative Application Development. "It has to do two things. One is database development, building the way to get data into the application. The other is the graphical user interface for the presentation of the application's contents."

Maureen Leland, Iris Project Leader for Domino Designer, says that her team's goal for Designer Rnext has been to give developers "the ability to do whatever you want, the way you want to do it. If you want to work in HTML, you can. If you want to use a third-party application to edit a design element, you can."

When Leland and Hoyt talk about and demo Designer Rnext, what comes through are four major areas of focus for improvements in this new version:

- **Database development.** Many of the changes in Designer Rnext are intended to make it easier for developers to do the basic work of building an application—from small UI changes to major additions like the Data Connections resource type.

- **Presentation development.** The Web has fundamentally changed the presentation of content, and Designer Rnext includes many changes and additions that bring the creation and management of new presentation elements, like layers and style sheets, into the IDE. At the same time, they are intended to make Designer a tool that leverages the skills of Web designers, as well as Notes developers.

- **Reusability.** As Domino applications become larger and more complex more code can be reused. Designer Rnext includes many changes that make that reusability easier.

- **Managing complex applications.** Designer Rnext provides better support for managing applications that span multiple databases and include objects that aren't traditional elements of an .NSF file—and for third-party tools to use on the design elements of those applications.

Support for complex applications

When you open Designer Rnext, you'll notice major changes in Designer's Bookmarks pane. When you expand a database list you see several new types of database resources. And you can create nested folders that contain arbitrary sets of databases, design elements, and other kinds of objects that aren't traditional parts of an .NSF file.

This screen shows folders that have been created as children of a database, fishnet.nsf, to hold pages and views. These elements can come

from this database or other databases, and you can open them directly from the bookmarks.
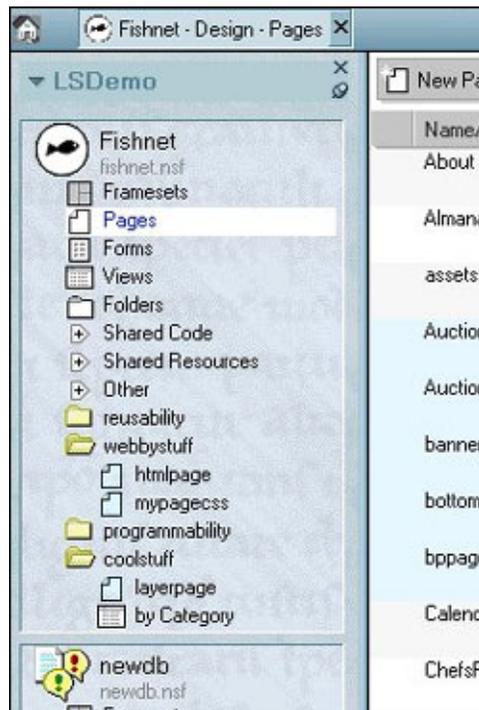
You can drag applications into Bookmarks, as well, and use it as a task bar for programs you run frequently. You can also create new folders here, and drag databases and other elements into them to group all the pieces of your projects together.

With the Shared Code and Shared Resources items expanded, you can see new resource types: Script Libraries now include JavaScript Libraries, and Servlets; Java Server Pages are added. Style Sheets and Data Connections have been added to Shared Resources.
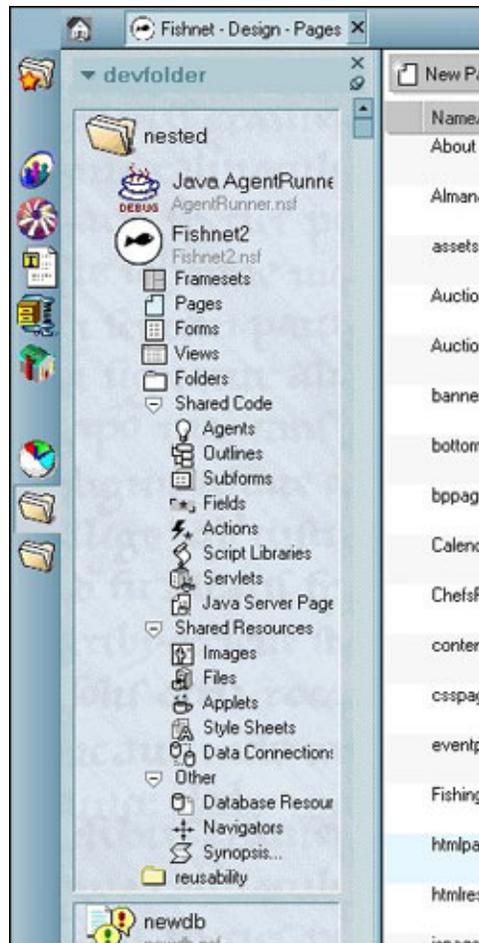
When you expand individual items in the list, you will see new icons and columns in the view pane, as well, that reflect new properties you can set for these elements. One of these is an icon to indicate whether an element has been locked, so that an element can't be modified by another developer until the lock is released by the developer who set it. This provides basic file-locking for applications that are being worked on by teams of developers, and the locking technology was implemented in ways that support third-party check-in-check-out and versioning products for Domino development.

In this element view, you can also select multiple elements for action such as setting the hide-when properties of a number of similar elements at once.

Reusability

Shared Resources brought a "create once, manage centrally, use everywhere" convenience to some kinds of resources and code segments in R5—image resources and agents and subforms, for example. A major goal of Rnext is to expand this kind of easy reusability to more kinds of elements. JavaScript Libraries are added to the library types, and several new kinds of shared resources show in the database design list:

- Files
- Style sheets
- Servlets
- Java Server Pages
- LayerPages
- Data Connections

You can make a bookmark link to a design element or code object in a different database, so that a frequently used block of JavaScript, or a standard view or subform, can be managed centrally across several databases.

A new "Insert resource" dialog makes it easier to reuse objects, as well:


This dialog is context-sensitive: You see only elements appropriate to the current object. If you are working in the JSHeader object you see only JSLibraries, for example. If you are working in the HTMLHeader you will see StyleSheets.

The selection dialog lets you select the database and the resource type, and displays the available resources plus any resource specific fields.

(This "Insert resource" dialog is the primary reason to select the appropriate type when you create a resource, by the way. You could save a style sheet simply as a file resource, but then it wouldn't be available in the "Insert resource" dialog in the HTMLHeader object.)

The ability to work with resources from other databases extends to design elements as well—in particular, views. In Designer Rnext you can embed a view from another database into a page or form—and in fact, you can embed multiple views. Outlines can be embedded in similar fashion.

Another reusability enhancement is a new set of actions for shared elements that show as a row of buttons across top of the view of all elements of that type:
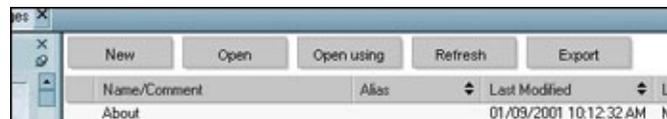
"New" is actually not new, but the rest are:

● When you select a resource and click "Open," Designer makes a temporary copy of the object, saves its pathname in the Design Note associated with the object, and opens it in whatever application is associated with the object's file extension on your workstation.

● "Open using" performs the same action as "Open," but allows you to choose the application. Designer saves the pathname of this application in the object's Design Note, so that the next time you "Open using" for that object the application you used previously will be the default.

● When you have edited and saved the object in the external application, return to Designer and click "Refresh." Designer follows the stored pathname to the temporary copy, replaces it into the resource, and removes the copy.

● "Export" takes advantage of DXL (Domino XML) and the fact that all Domino objects are expressible in DXL. Click "Export" and the object is exported in DXL format.

Presentation design

Release 5 of Designer made major steps forward in incorporating support for Web technologies into the IDE, and Rnext makes several more very significant steps. Some new features smooth out differences between developing for the Notes client and for a browser, while others provide ways of doing things and using tools in Designer that will be familiar to Web developers.

Editing a style sheet is an important example of the latter: Style sheets are a new resource type in Rnext. Rather than invent a new editor for them, the Designer team simply incorporated a style sheet editor, Top Style, that is popular with Web developers. The view of all style sheets carries an action button that says "Edit in Top Style." When you select a style sheet and click the button, Designer uses the technology built for "Open" and "Open Using" to open the style sheet in Top Style.

Designer Rnext also lets you edit and check the syntax of HTML, Servlets, Java Server Pages, and DXL. For these it uses the same editing control developed for LotusScript and JavaScript.

Editing HTML can be taken even farther in the new version of Designer. The Rnext Notes client and Designer both render pass-through and edited HTML. Designer lets you work with HTML code either as text, or in WYSIWYG display:
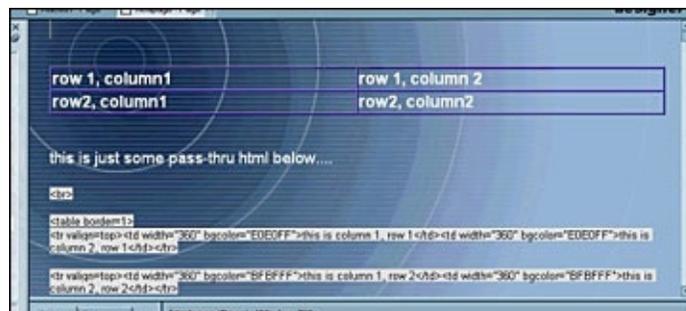
This partial screen image shows a page with a two tables—one a WYSIWYG table created in the Designer, the other in HTML code. Select the WYSIWYG table and select View - Display as HTML and the page is redisplayed:

The WYSIWYG view of the table is replaced by the HTML code. You can edit this selected HTML code in the HTML pane:
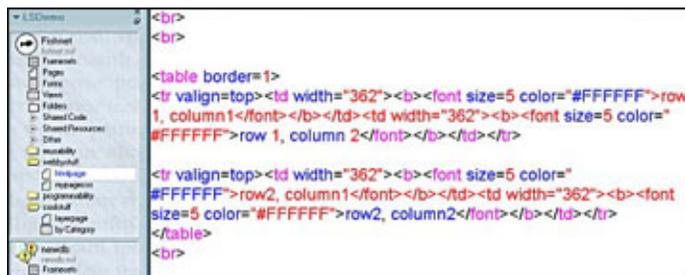

Finally, when you're finished you can select it again and turn it back into WYSIWYG display.

Designer Rnext incorporates similar support for an increasingly important Web design feature, layers, in ways Web designers will find familiar as well: You can save layers as resources, insert them into a page or form, manipulate them in WYSIWYG view, and edit their properties, setting absolute position and Z order, and nesting layers within other layers.

Database development

In Rnext, Designer moves to harmonize the development of applications for the Notes client and the Web. The Notes client's ability to render HTML is an important feature, because it makes it possible to write HTML that runs on both clients without recoding. Another new feature that will reduce the duplication of effort required to prepare an application for both Notes and the Web is an enhanced event model that does away with redundant events—one for Notes, the other for the Web.

When you open an event in Designer Rnext, two new drop-downs appear at the top of the programmer's pane:
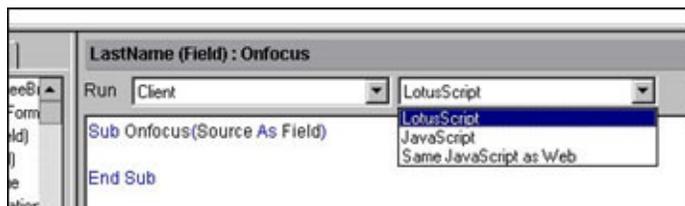
If you're writing LotusScript to run in a Notes client, you choose "Client" in the first field and "LotusScript" in the second. If you're writing JavaScript for a browser you choose "Web" and "JavaScript." You can write both kinds of code in the same event—JavaScript for the Web and LotusScript for the Notes client, and both will be stored and executed appropriately. Or you can write JavaScript and specify you want both clients to run it by selecting

"Same JavaScript as Web" for the Notes client (or "Same JavaScript as Client" for the browser if you wrote the script for the Notes Client).

Many of the enhancements in Designer Rnext extend the properties-box paradigm to make functions easier and more intuitive. One of the most powerful is the creation of a Data Connector and linked fields that connect a form to an external database. This brings a new ease to integrating Domino and relational data sources. It takes three steps:

First, in the database properties, select "Allow connections to external databases" as shown here:
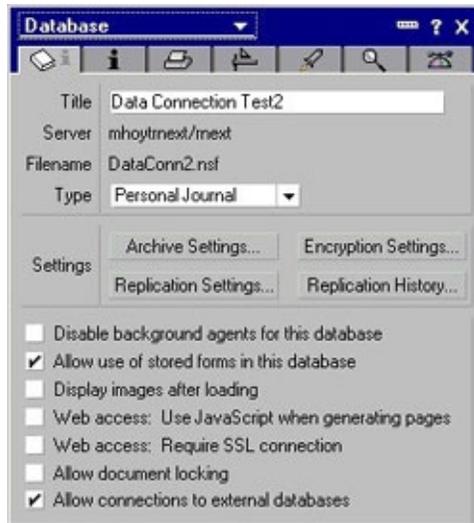
Second, create an ODBC data source (this example creates a connection to a DB2 table named Employee) and save it as a database resource:

Finally, set the properties for the fields that will connect to the external database. You'll need to specify the table and column names, the data connection, and whether the field is a key field or a data field. (One key field and at least one data field must be specified.):

What you've just done is configured a real-time connection using Domino Enterprise Connectivity Services (DECS) in a way that is easier than it's ever been.

Other cool stuff

Not every new feature in Designer Rnext is as big as the Database Connection. Many of them are small—but important to making a developer's life a little easier:

- LotusScript support includes a remote debugger capability for server agents.

- An Actions design pane displays an icon for each action, supports multiple selection, drag-and-drop to set the actions' order on the action bar, and double-click to edit in place. From this new interface you can delete system actions you don't need (and system actions are no longer added by default to new forms and views).

- A redesigned agent interface provides more screen real estate to write code. The agent properties have been enhanced, as well—the most important change allows you to make a shared agent private with a click in a checkbox—and the reverse.

- Dynamic named elements have been enhanced. The creation dialog allows you to compute not only the element name, which you could do in R5, but the database the element is stored in, and the kind of element it is—form, view, etc.

- You can customize the twisties that appear in views using an image resource that is an image well—the first state is displayed when the category is collapsed, the second when it is expanded.

- A "No refresh" selection is added to the database properties, which allows you to effectively lock the design of the database against changes.

- Type-ahead is enabled in the programmer's pane.

- You can print the code that appears in the programmer's pane
- The 64K limit on code segments has been removed.
- You can set persistent breakpoints in the LotusScript debugger.
- Views have a new property: "Hide column if formula is true."
- Color palettes have been made consistent throughout Designer Rnext.

Power for productivity

The changes in Designer Rnext are evolutionary rather than revolutionary, says Maureen Leland. The goal for this new version of Designer, she says, is to make it "fun to use." Then she amends that, adding, "...and powerful."

Designer Rnext is indeed powerful to use. It delivers features that make the development of increasingly complex applications easier—features that save developers from repetition and wasted effort, and makes them more productive. It makes it easy to use new presentation features like layers, and new programming features like database connections. It brings the IDE up to speed with the current state of the application-development art. And it really does make it fun.