Developer tools for building better Web sites

by Carolyn Kraut

[Editor's note: This article resides in “Iris Today”, the technical Webzine located on the http://www.notes.net Web site produced by Iris Associates, the developers of Domino/Notes.]

If you've been paying attention to the technological developments as they relate to the Web, you know that there is much that can be done by using native Notes to create an effective Web site. You also know that there is much that can be added to a Notes Web site to increase its functionality, its usability, and give it more pizzazz. With technology changing at a breakneck speed, how can you meet the demands of your customers be they internal or external? Rather then spending a lot of time learning Notes development, learning HTML, learning Java, and learning a series of foreign languages before you can develop an effective Web site, you can use the following Lotus development tools with Notes Designer -- the Lotus BeanMachine, Domino.Action, and Notes Global Designer.

Lotus BeanMachine

The Lotus BeanMachine is a visual authoring tool for the creation of Java applets. Effective Web sites incorporate a variety of components depending on the purpose of the Web site. While the majority of a Web site may be text and images, it can also include sound, animation, and other special effects. More and more, Web sites are interactive. You can request information, make purchases, download software, and so on. In these instances, Java applets can provide the glue between what the site visitor does and what the Web site does. A simple Java applet could display a message acknowledging the site visitor's action. On the other hand, it could be as complex as your imagination can make it.

So how can you incorporate a variety of objects in your Web application, including Java applets? You can learn to program in Java. You can learn HTML. There are also a variety of authoring tools just waiting to make it just a little easier and faster than the "from scratch" method of creating a Web site. One of these tools is the Lotus BeanMachine 1.0, currently designed to run in a Windows 32-bit environment.

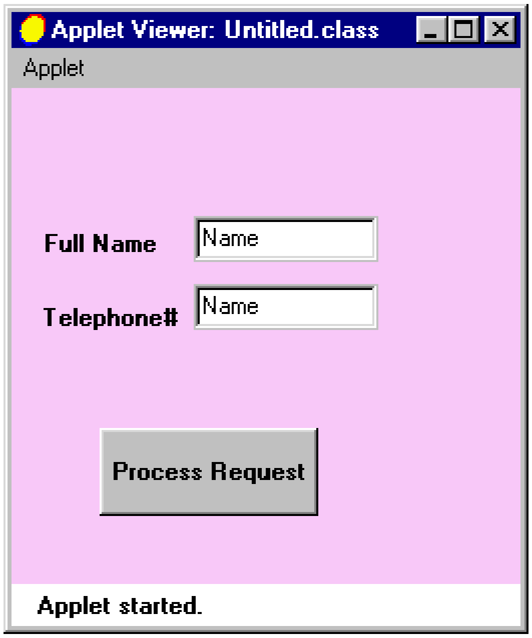
So what beans are included for brewing in the coffee pot?

There are beans for brewing multimedia-like animation, audio, and special text effects, to mention a few. There are beans for brewing standard application controls like checkboxes, list boxes and drop-down lists. There are beans for brewing database connections, links to URLs and mail links.

Each of these *parts*, as they are called, have associated properties, actions, and events. In the category of multimedia there are nine parts. The animation part will take a sequence of graphical images and make them into a movie. The audio part gives you the ability to incorporate sounds and music. The clock part displays either a digital or analog clock. The timer part can be used in your applet to keep track of elapsed time. The image part lets you display a GIF or JPEG image in your Java applet. Images can be displayed using special effects (transitions). Some of the transitions you can select are dissolve, wipe down, ripple, and shear. You can also set the speed at which the transition will occur. The rollover part acts as an image toggle for GIF and JPEG images. When the mouse pointer passes over the image, or the mouse button is pressed, the image will change. The rollover images can be displayed in one of two ways: without a border, or with a 3D border. When the 3D border is applied, the image looks like a graphic button. There are also text parts included in the multimedia category -- Text, Nervous Text, and Ticker Tape. The text part allows you to incorporate rich text in your applet. Nervous Text is text that bounces around in a nervous way. You can set the caffeine level of the text to decaffeinated (calm), caffeinated (hyper) and high test (manic). Ticker Tape scrolls text across the screen in the same way the Windows scrolling marquee screen saver does.

In the Network category there are three parts -- Database, Mail, and URL Link. The database part allows you to create connections to ODBC databases as well as DB2 databases. With the database part you can create powerful applets that access, display and modify information stored in a variety of databases, including Oracle, Sybase, DB2, and Microsoft Access. You can then display the applets in a variety of formats, including ones you design yourself. However, due to a restriction in Java 1.0.2, the database part can not be used across the Web. When using the database part in an applet, the database being referenced must be located locally with the applet. Randy Eckhoff of IBM's Multimedia Authoring group has indicated that this restriction has been removed in Java 1.1, and will therefore be removed in the next release of the BeanMachine. The mail part will display the user's browser e-mail dialog, so they can compose and send an e-mail through the Internet. The URL part gives you the ability to create a link to and display a new Web page. This part also supports frames and gives you the ability to indicate which frame the new page is displayed in.

The Controls category includes parts for creating standard Windows-like controls in your Java applet. These controls are Button, Checkbox, Choice, Label, List, Panel, Text Area, and Text Field. Button and Checkbox are what you would expect -- standard Windows-like controls. Choice is a drop-down list box that can contain up to 15 choices. Label lets you create a static text label anywhere in your applet. List is just that, a scrollable list of items. Panel is a container for other BeanMachine parts, and can be used in the same way Notes layout regions are used. Here you see a preview of a Java form created from the panel part (Lavender background), the label part, the button part, and the text field part.



Full Name Name

Telephone# Name

Applet started.

As you've probably already guessed, the text field part is used to display editable text, or receive text input from a user. The last part in the Control category is the Text Area part. This part creates a scrollable text box for entering and editing multiple lines of text.

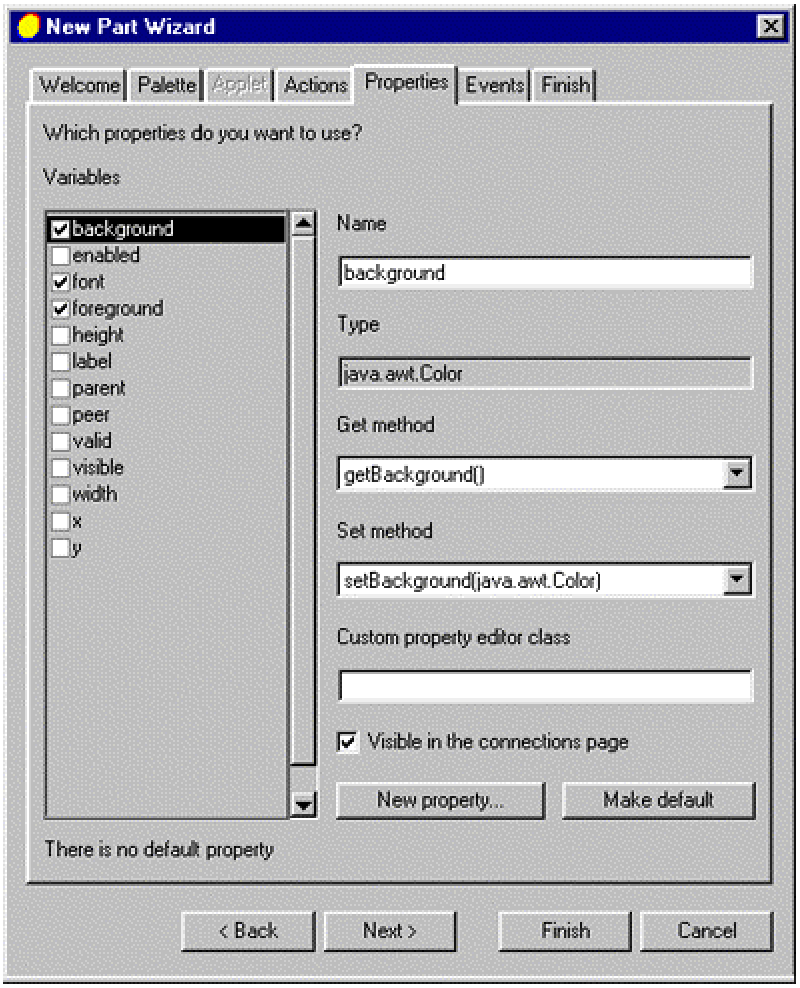
Cappuccino, Espresso, Café Latte...

While BeanMachine provides a terrific set of basic parts for creating Java applets, you may want to create something more exotic, more flavorful. With the BeanMachine you can not only create your own parts from scratch, but add them to the part palette as well.

Growing your own beans

With coffee being ever so popular these days and prices sky high, you may decide you want to grow your own beans. Or maybe you have a friend who imports beans from all over the world, making some very rare beans available to you. Can you use imported beans with the BeanMachine? You can with a couple of caveats. imported JavaBeans should be created following the Sun Microsystems specification for JavaBeans. They should be compiled using Java 1.0.2. JavaBeans compiled with Java 1.1 are not guaranteed to work with BeanMachine, and BeanMachine will not warn you if you are importing a class not compiled with version 1.0.2.

BeanMachine contains a New Part Wizard that makes incorporating your Java classes and applets a snap. Open this wizard by choosing Options - New Part Wizard. Seen here is the Properties tab of the wizard.



Which properties do you want to use?

Variables

☑ background
□ enabled
☑ font
☑ foreground
□ height
□ label
□ parent
□ peer
□ valid
□ visible
□ width
□ x
□ y

Name: background

Type: java.awt.Color

Get method: getBackground()

Set method: setBackground(java.awt.Color)

Custom property editor class:

☑ Visible in the connections page

There is no default property

You can select properties from the list at the left, or define a new property. You also have the same flexibility when choosing Events to be supported by the code you are importing. Once you have specified how the part should appear in the palette, and the default applet parameters, actions, properties and events, BeanMachine compiles the code and places the part in the palette, making it available for immediate use.

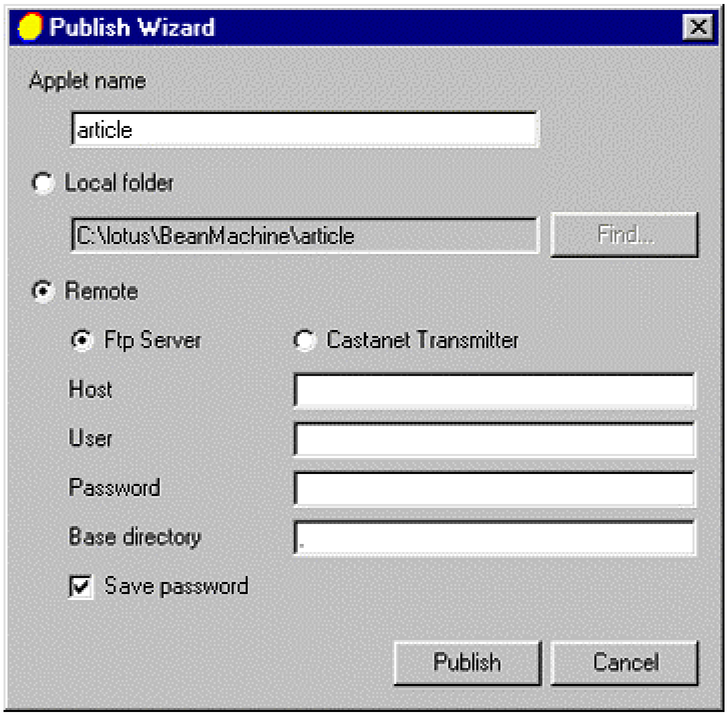
Creating a house blend

When you come to the point where BeanMachine's built-in parts and connections don't meet all of your needs, you can combine the BeanMachine functionality with your knowledge of Java. You do this by writing your own code in the BeanMachine Java window. There, you can weave together existing parts -- BeanMachine's and your own -- with new methods.

Publishing an applet on the Web

Once you have designed and tested your applet, and everything is the way you want it, you will want to publish it. After all, why spend all that time creating a masterpiece if no one else is going to see it?

BeanMachine provides the publishing wizard to makes things easy for you. You fill out the form, and BeanMachine does the rest.



Publish Wizard

Applet name

article

○ Local folder

C:\lotus\BeanMachine\article

● Remote

● Ftp Server ○ Castanet Transmitter

Host

User

Password

Base directory .

☑ Save password

All the files associated with the applet will automatically be located and placed in the correct directory structure at the location you've specified.

For more information about the Lotus BeanMachine, visit these Web sites:

- http://www.lotus.com/developers/tools/beanmachine.nsf
- http://www.ibm.com/java/appletauthor/

For more information about the JavaBeans specification, visit http://www.javasoft.com

Domino.Action

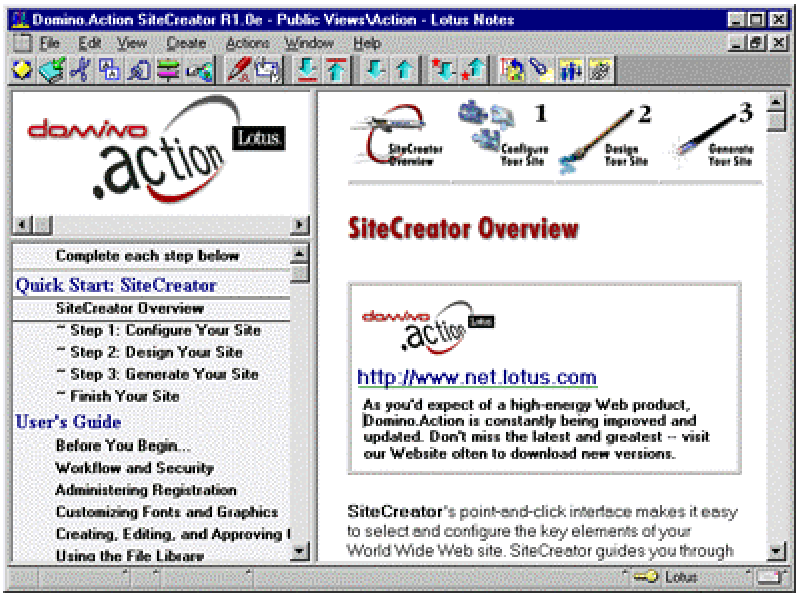
The decision has been made. Your company's Internet/Intranet strategy will revolve around Notes and Domino. The first order of business it to put up a Web site. The schedule is aggressive, the resources are limited, and to make matters worse, you haven't really had time to learn the ins and outs of Notes development. How to proceed? Proceed directly to Domino.Action. Domino.Action consists of two Notes templates -- SiteAct.ntf and LibAct.ntf. From these templates you create the Domino.Action SiteCreator, and the Domino.Action Library. From there, it's a simple matter of reading, pointing, clicking and previewing. No special development skills are needed.

When you use Domino.Action to create your Web site, you can create a site that contains features like a home page, corporate policies and procedures, and job postings. Of course you may not want all of the available features in your Web site, at least not at first, and with Domino.Action, you can pick and choose the ones you want.

In addition to choosing which features will make up your Web site, you can also customize the look and feel, and the security of the Web site. Once you have made all your selections, Domino.Action will generate the databases specified, and include the graphics, security and other options you have selected while designing the site.

The SiteCreator

The SiteCreator seen here, is a database in which you specify the features you want to include in your Web site. The creation process consists of three categories of steps: site configuration, site design, and site generation.



SiteCreator Overview

http://www.net.lotus.com

As you'd expect of a high-energy Web product, Domino.Action is constantly being improved and updated. Don't miss the latest and greatest -- visit our Website often to download new versions.

SiteCreator's point-and-click interface makes it easy to select and configure the key elements of your World Wide Web site. SiteCreator guides you through

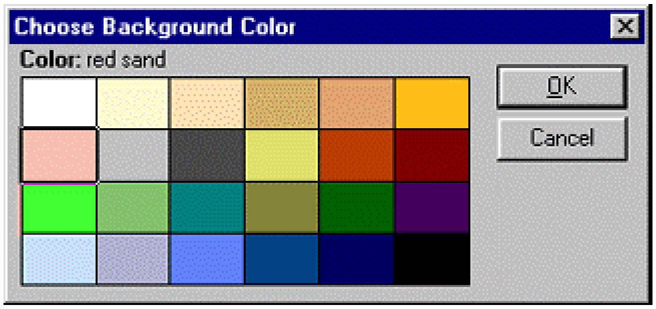
The first step, configuring your site, consists of indicating the areas you want included in your Web site. Area choices are:

- About the Corporation
- Corporate Policies and Procedures
- Discussion
- Document Library
- Feedback
- Frequently Asked Questions (FAQ)
- Home Page
- Job/Career Opportunities
- Products/Services
- Registration
- White Papers

Also important to the site configuration process is specifying who manages the site, where they are located, and how they can be contacted. Finally, you will specify physical details about the site -- the location of the server, the site directory, the site library location, and the site library filename. That's it. Just answer a few questions, and the basic configuration of your site is done.

The next challenge is to design your site. When designing your site, you can elect to use the default design elements (Quick Design) or you can be choosy (Customize Design). Needless to say, selecting Quick Design is the fastest, easiest solution. When you choose Quick Design, Notes agents retrieve the configuration information you entered, and then create the area documents using the default values contained in the SiteCreator database. You can then review these documents to see if you still want to retain the default settings, or enter new ones.

If you're like me, you'll select Custom Design. When you click the Custom Design button, a Notes agent runs and identifies the areas you chose to include in your site during the configuration process. For each area, you can specify the text that site visitors will click on to get to that area of the site. You are also able to "design" the look of the area. For each area to be included in the site, you are able to choose the look of the page, background color, and background image. Domino.Action SiteCreator includes 26 layout styles. The styles fall into one of three categories: Basic, Contemporary and Corporate. Choose your page look from one of these. The background color is selected from this pop-up palette:



Choose Background Color

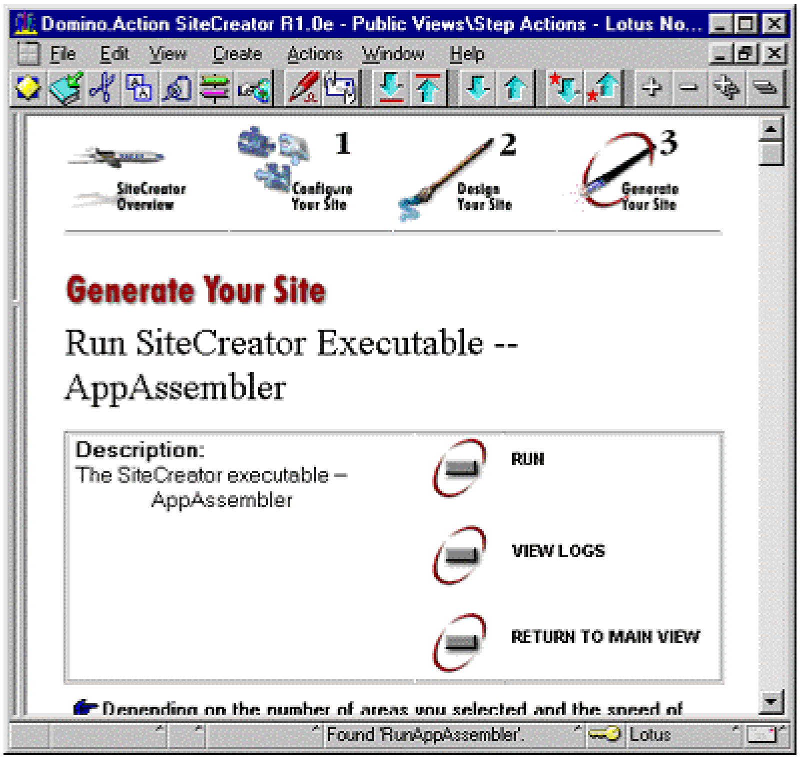
Color: red sand

There are four choices for the Background Image for each style category (16 total): Top, Bottom, Side, and None. In addition, you can customize the copyright information displayed on the pages of each area.

Each area also includes security and approval settings. Security information includes the names of those authorized as content composers, and specifies whether to allow anonymous access to the site, whether visitors can author pages in the area of the site, and how many approval steps make up the Web page creation approval process.

Site Generation

When you tell Domino.Action to generate your site based on your selections, it first runs Notes agents that refresh the documents that either you have created or the SiteCreator has created for you. This ensures that all of the latest changes are incorporated into the design of your Web site. After this initial processing, the "Generate Your Site" document seen here, will be displayed.



Domino.Action SiteCreator R1.0e - Public Views\Step Actions - Lotus No...

Generate Your Site

Run SiteCreator Executable -- AppAssembler

Description:
The SiteCreator executable -- AppAssembler

Depending on the number of areas you selected and the speed of

Found 'RunAppAssembler'.

When you select RUN, an external program (appassmn.exe) created with the HiTest API toolkit, and located in your Notes directory, will execute. AppAssembler creates a separate database for each area of your Web site and populates it with some basic documents. These databases can be viewed or edited through a Notes client or Web browser. Before making the site accessible to the public, you should go into each area of the site and fine tune the security as well as some of the documents. You may also want to enable and/or modify one or two agents. Once this is done, you're ready to go live.

The Library

In addition to functioning as a warehouse for the GIF files used in the creation of site styles, the Domino.Action Library is used to monitor the approval process, process requests, store the names of registered users, and to log system activities. This database is the Webmaster's friend and should be monitored regularly.

Extending Domino.Action

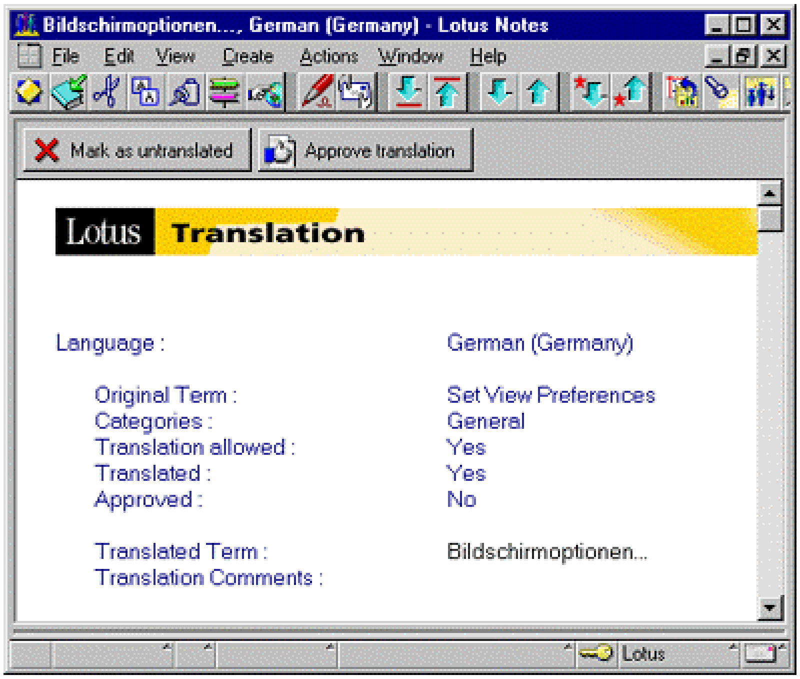
Its highly conceivable that you would like to include an area in your Web site that doesn't already exist in SiteCreator. Or maybe you just want to add new page types to the areas already included. Since SiteCreator is a Notes database whose design is not hidden, its possible to do this. However, you should not attempt to do this unless you are an experienced Notes developer. SiteCreator's design is rather complex. You will need to proceed with caution and develop an understanding of the relationships between the existing database structure and the elements you wish to add. For additional information on extending Domino.Action, visit the Domino.Action Web site at http://www.net.lotus.com/

Notes Global Designer

Multinational corporations and third party Notes developers face the challenge of making their applications usable around the world. One of the major challenges facing these developers is the translation of their applications into one or more languages. What about creating an application that can run using different languages based on user preference? How is this done? Well, up until now, not very easily. What makes it easier now is the availability of the Notes Global Designer (NGD). The Notes Global Designer 4.6 makes it possible to automate or semi-automate the translation of new or existing Notes databases into one or more languages. It also makes it easy for Notes application developers to create applications that function in more than one language while on the same server (multilingual database), allowing each user to decide which language they will work in. The primary users of the Notes Global Designer are Notes application developers and translators. Notes application designers will work primarily with the WorkBench and the Populator, while translators will focus their talents on the creation and enhancement of glossaries to be used during the translation process.

Glossaries

Notes Global Designer comes with four different glossaries. Three of them consist of terminology applicable to Notes database development. There is a glossary of Notes 3.0 terminology, another one containing Notes 4.5 terminology, and a third containing Lotus terminology. The fourth glossary contains generic terminology. Unlike most glossaries you may be familiar with, these do not necessarily translate individual words into foreign languages, but instead translate phrases. A typical glossary entry is the one seen here for the term "Set View Preferences."

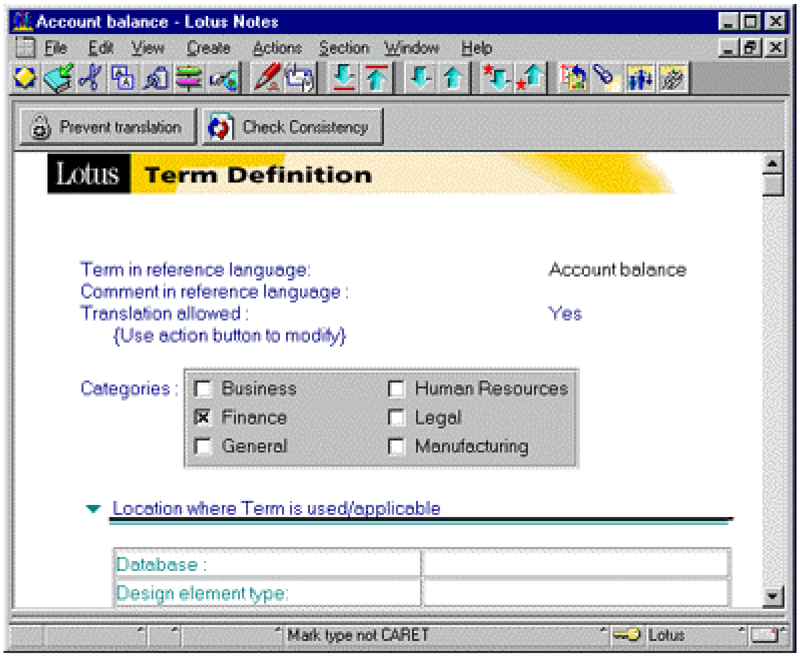


Language :	German (Germany)
Original Term :	Set View Preferences
Categories :	General
Translation allowed :	Yes
Translated :	Yes
Approved :	No
Translated Term :	Bildschirmoptionen...
Translation Comments :	

The pre-populated glossaries that come with Notes Global Designer are not meant to be the only glossaries you will need for translating your application. They are meant to give you a start. These glossaries are contained in easy-to-update Notes databases. In addition to creating new entries and editing existing entries directly in the glossary databases, you can import translations from other sources, and export the contents of existing glossaries using standard Notes import/export formats. You can also change the contents of a glossary by using the WorkBench or Populator tools that make up the Notes Global Designer. The Notes Global Designer also includes a glossary template. Therefore, you can create as many new glossaries as you need, incorporating them as needed into your translation projects.

Not only does the glossary provide language translations, but it allows you to provide context for the terms being translated. In many languages the same word can mean different things depending on how it is being used. In some languages different words are used to address a person depending on the familiarity of the speaker with that person. An example of this is the English word "you." In French, "you" can be translated as "tu" or "vous." Another example in French also shows the importance of context for any language translation. In English, the word "state" can mean a condition. It can also refer to a politically organized body of people within a defined territory a la Massachusetts. In creating a glossary entry for such a word, you may want to capture more than one translation. The way to distinguish between the translations is to indicate the category (context) in which they fall.

Context information can also pertain to the way a term is used in a particular application. You can provide this type of context by indicating the database for which the term applies, the design element (Form, View, and so on) to which this term applies, the design element name (Memo, All by category, and so on) to which this term applies, the design element subtype (static text, column header, and so on) to which this term applies, or the design subtype name to which this term applies. This information is specified in a Term Definition entry in the glossary. The term defined here is located in the Generic Terminology glossary. There is no specific database context for the term, but it is categorized as a financial term.



Term in reference language: Account balance
Comment in reference language :
Translation allowed : Yes
{Use action button to modify}

Categories :

☐ Business	☐ Human Resources
☒ Finance	☐ Legal
☐ General	☐ Manufacturing

Location where Term is used/applicable

Database :	
Design element type:	

Context information is then used to provide the best translation for a term based on the application being translated.

In addition to storing text translations, glossaries can be used to translate bitmaps. This is really useful when database navigators contain bitmap hotspots with words. While Notes Global Designer can't alter the original bitmap to translate the words embedded in the image, replacement bitmaps can be stored in the glossaries to be exchanged with the original during the translation process.

The Populator

The Populator is the tool you use to design a Notes database with the Notes Global Designer translation process in mind from the very beginning. The Populator allows you to select and load the glossaries needed to translate the application, set the glossary lookup options and tag text elements of the design as they are created. Alternatively, you can create the database in your native language, and then use the AutoPopulate process (described below) to tag the database.

Tag -- Your It!

What is a tag? Tags are unique identifiers that get assigned to terms found in a Notes database. They are 17 characters in length, plus the start and end delimiters specified by the user. The tag has this syntax: {~[17chars]term being tagged~}, where "{~ is the start delimiter and ~} is the end delimiter. Because the tag is so long and Notes design elements have length limitations, a tagged term may appear truncated in the master database. However, this does not affect translation. The translated or target databases that you build will contain complete terms. One thing to keep in mind when designing a database for translation in other languages, is the length of the terms when translated. Translated terms may be longer than the original terms and exceed Notes R4 limitations. In those cases the terms will be truncated in the translated database.

Terms can be just about any text string found in a Notes database. NGD will translate everything -- the Notes formula language, embedded Web elements, computed text, field labels, button labels, infobox text, and bitmaps (in rich text fields). It will not translate LotusScript, or private views and folders. When you use the Populator to tag your design elements, you are tagging the database manually, looking up each term as it is created. Included with NGD are several SmartIcons that you can add to your workspace. There are three associated with the Populator: Launch Populator, Lookup in Populator, and Retrieve from Populator.

These icons allow you to interact with the Populator while designing your Notes database. You highlight the term to be translated, launch the Populator and then click the Lookup in Populator SmartIcon. If the term is not in the glossary, you will be able to add it. If the term is found, you will be able to place the tagged term in your design using the Retrieve from Populator SmartIcon. This may sound like smoke and mirrors, but it's actually accomplished with a Dynamic Data Exchange (DDE) link between the Populator and the Notes database you are working in.

The WorkBench

The NGD WorkBench is the primary tool for Notes developers when translating an application. From the WorkBench, developers are able to launch the build process and identify the source database. They are also able to identify and describe the target or end-result database(s). Describing the end result starts with selecting the glossaries to be used during the translation process. Once the glossaries are selected, the languages the application should be translated into can be selected. You then choose other identifying characteristics: unilingual versus multilingual, replica or new copy or altering the current database, and original ACL versus different ACL. Finally, the WorkBench builds the translated application.

A unilingual database is one that is translated into a single language. A multilingual end-result contains design elements in more than one language. Multilingual databases allow end-users of an application, running on the same server, to work in different languages. When a user opens a multilingual database for the first time, they are prompted by the About Database document to install and prepare the multilingual addin needed to switch between languages. The installation process adds the following to the user's NOTES.INI file: EXTMGR_ADDINS=EMINTN01.DLL. The user is also asked to specify their language preference, which is also stored in their NOTES.INI file. The entry looks something like this: $BabelPref=FRFR;ZHCN. The addin checks this variable in the NOTES.INI file. Then, it hides all the design elements not in the preferred language and displays those elements in the preferred language. There is one limitation in regard to multilingual applications -- data content entered by end-users remains in the language in which it was entered.

AutoPopulate

The AutoPopulate process is the recommended way to use NGD with existing Notes applications. It is faster than using the Populator, and allows you to tag a Notes database selectively. You can choose all, one, or several design elements of the database to be tagged. AutoPopulate goes through the design elements you've selected, looking for terms that can be but aren't tagged. When it finds such terms, it asks you what you want to do. You can either skip the term or add it to a glossary.

Another difference between AutoPopulate and the Populator is that the former does not generate a separate target database. Instead, it makes the changes directly in the source database. AutoPopulate can also be an interactive process. You can set it to prompt you before tagging anything in the database its operating on. The one thing that you can do with the Populator, and not with the AutoPopulate process, is translate view selection formulas.

UnPopulate

When you choose to unpopulate an application, you are asking NGD to remove all the tags from the master database. This operation breaks the connections between the master database and the glossaries used when the database was tagged. The result is a standard, untagged Notes database. As with the AutoPopulate process, the UnPopulate process can be set up to work only on selected design elements in the specified database.

Command Line Options

The WorkBench can be launched from DOS to perform batch build operations. Command line parameters tell the WorkBench what to do. From the command line you can tell WorkBench to autopopulate a file, build a translated application or unpopulate a file.

Web Applications

What does all this mean for Domino applications designed for the Web? Well it means that now you can create Web applications that are truly worldwide with relative ease. This is a truly novel idea in a world where Web sites are generally displayed in the native language of the site's location. Already I know what you're thinking. Great! I can have a huge multilingual database that displays forms and views in the preferred language of the user. Unfortunately, this feature is not available for Web applications. You can however enable language switching from the application's home page.

In addition to native Notes design elements, your Web applications will probably include HTML. While the Notes Global Designer can recognize HTML, it will not tag it. Currently, any strings embedded in HTML need to be translated manually.

Notes Global Designer in action

Domino.Action 1.0c was translated to 10 European languages by the Domino.Applications Localisation Team in Dublin. The languages are French, German, Italian, Spanish, Brazilian Portuguese, Danish, Dutch, Swedish, Finnish and Norwegian. As described earlier in this article, Domino.Action includes two Notes template files -- Siteact.ntf and Libact.ntf. These templates posed some technical challenges for the team as they are more functionally complex and considerably larger in size than any templates previously translated in-house. These two templates are the core of a highly functional application.

Carmel Somers, the Localisation Project Manager for Domino.Applications, says "Notes Global Designer was the obvious tool for us to use to translate Domino.Action as it handled the Libact.ntf (3mb) and Siteact.ntf (8MB) files without issues, and allowed us to translate the databases without breaking product functionality or requiring us to re-engineer the US product in any way during the translation process." Using NGD, they identified and tagged the user visible text which required translation, and extracted this to a glossary template file. A copy of the glossary file was sent to each of the language translators, who used NGD's Translator View to translate the text. Upon receipt of the translated glossaries, Lotus software engineers used NGD to build translated Domino.Action templates using the original "tagged" template and the translated glossaries. One of the biggest benefits of this method is that NGD offers a complete environment for the translation of the templates. "This gave us total control of the process and any issues found are central to one tool and one overall process," says Somers. "If something goes wrong, it's easier to detect and address, as it is isolated to NGD and the NGD process."

The alternative to NGD is to use several tools. There might be one for extraction of the text, one for storage and translation, and another to reapply the translations to the template. This usually necessitates changes to more than one process and involves multiple tools. Another advantage of using NGD becomes apparent when the next release of Domino.Action is ready to be translated. NGD allows for the reuse of the existing translated glossaries. This preserves some of the investment in the translation costs incurred during the translation of the first release.

About the limitations of NGD, Somers says "The one drawback we had when using Notes Global Designer 1.0 was that it did not support the translation of LotusScript, GIF files and data documents (Notes content documents). We developed some tools in house to support the translation process of these items. As an NGD customer we are working with the NGD Development Team to ensure support of these objects in future versions of NGD." Currently, NGD is unable to translate LotusScript or the contents of file attachments. In the case of Domino.Action, the file attachments were GIF files. The data documents contained in Domino.Action include the help documents and documents containing sample information.

System Requirements

While the Notes Global Designer requires a Windows 32-bit environment when translating Notes database applications, users of translated applications are able to run them on any platform supported by Notes. Because glossaries are Notes databases, translators responsibile for populating glossaries are also free to work on any Notes-supported platform. Additional system requirements are:

Developer's workstation

1. Windows 95
   RAM: 12MB or more recommended.
   Disk Space: 15MB or more recommended. 160MB if installing shipping glossaries.

2. Windows NT 3.51 or higher
   RAM: 24MB or more recommended or more recommended.
   Disk Space: 15MB or more recommended. 160MB if installing shipping glossaries.

3. Notes Release 4.5
   Any language supported by Notes.

Translator's workstation

1. Any operating system supported by Notes Release 4.1 or higher.
   Same system requirements as the Notes client for each platform.

2. Notes Release 4.1 or higher
   Any language supported by Notes.

User's workstation

1. For Unilingual databases
   Any platform supported by Notes clients
   Same system requirements as the Notes client for each platform.
   Notes Release 4.1 or higher
   Any language supported by Notes.

2. For Multilingual databases
   Windows 3.1 or Windows 95 and higher, Windows NT, OS/2 Warp or Mac.
   Same system requirements as the Notes client for each platform.
   Notes Release 4.1 or higher. Notes Release 4.5 is recommended but not necessary if using multilingual Notes databases.
   Any language supported by Notes.

For additional information about the Notes Global Designer and Web site internationalization, visit these Web sites:

- http://www.lotus.com/developers/tools/ngd.nsf
- http://www.alis.com