



by
Sam
Bridegroom

Level: Intermediate
Works with: Designer 5.0
Updated: 09/01/2000

As application developers, many of us have found a variety of methods of using Notes to solve our business issues, and our existing applications reflect this. In a lot of cases, however, we didn't build the applications we're now supporting; we inherited them from developers of days gone by—many of whom were perhaps "slightly less than stellar" in their system documentation skills.

But whether they are applications we built or inherited, it's a safe bet that the user interfaces differ from application to application—and standardizing the look and feel of all of the applications would require making design changes to each application (or its respective design template). It's not very often such a task becomes a development priority in the eyes of the developer or of management, especially if the existing applications *are working!*

But what if there was a way to provide a standard navigational format without changing the design of existing applications? Well, there is—using the R5 framesets, outlines, and pages design components.

This article focuses on bringing all of these design elements together into a working business solution. It shows you how to achieve a standardized look and feel quickly by creating a standalone, user interface to standardize navigation around your organization's applications.

You can download a copy of the database used as the example in this article, [Acme Standard Interface](#) (homebase.nsf), from the Iris Sandbox. (Be sure to follow the download instructions and place both the Acme Standard Interface and Acme News databases in your Notes/Data directory so that they display properly.)

This article assumes a solid understanding of designing Notes/Domino applications using Domino Designer R5.

Why standardize with a standalone front end?

There are many reasons for standardizing your applications' user interfaces, but the best reason is the benefit gained by the end users. If your applications all have the same look and feel to them, it's easier for a user (a new user, in particular) to learn what buttons to push or how documents may be sorted and displayed in a view.

Usually, however, we tend to incorporate new user interface concepts into our latest applications as new functionality becomes available or to change the way information is presented as the programming staff changes—without considering existing applications and how they work.

If you create a standardized, standalone user interface to use as a front end for all of your applications, you're ahead of the game. The benefits of this approach include:

- **Independent design**

There's no longer a need to consider changing the code in production databases, which is good news for the programming staff. When production code works, we tend to leave it alone. A standalone user interface allows us to consolidate multiple databases' appearances

without tampering with tested and implemented designs or templates.

- **Flexibility to make quick changes**

Adding a new database or application? The independent design allows the developer or system administrator to quickly display the new link, without touching the end user's desktop or sending that confusing mail message that tells the user how to install the new application.

- **Dynamic content**

Adding the ability to display system-wide messages allows a system manager—who is not necessarily a programmer or system administrator—to update and manage daily bulletins or other news messages without compromising the new standard display.

- **Custom Welcome page**

Once the new standardized front end is designed and tested, it can easily be implemented as a user's Welcome page when he/she starts the Notes client. (For more information on this topic, see the *Iris Today* article [Customizing the Welcome Page.](#))

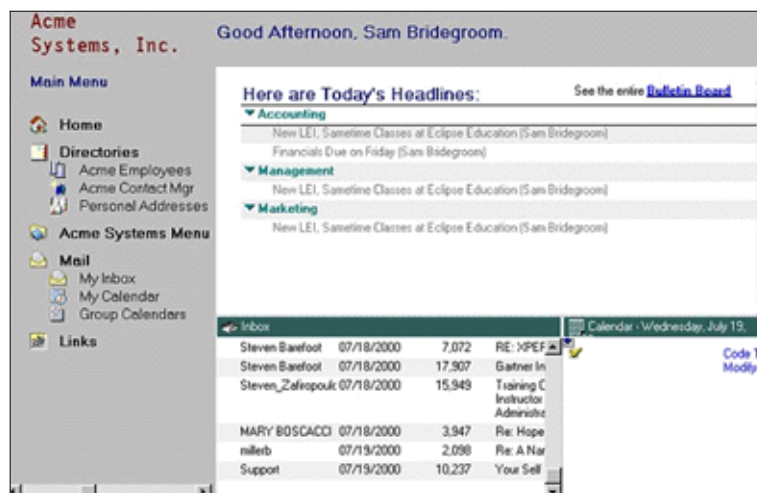
Don't misconstrue this approach as an excuse to write applications that don't conform to some sort of UI standard—that should be happening already. What this approach does is provide a quick and relatively painless shortcut to the process, particularly for existing applications. It gives programmers a little bit of breathing room to incorporate UI standardizations into future releases of their applications.

Creating the standardized front end

For this article, I've created the fictional Acme company. Over the years, there have been a number of developers (both in-house and contractors) who have created databases for Acme. Now they need to standardize them, using six steps:

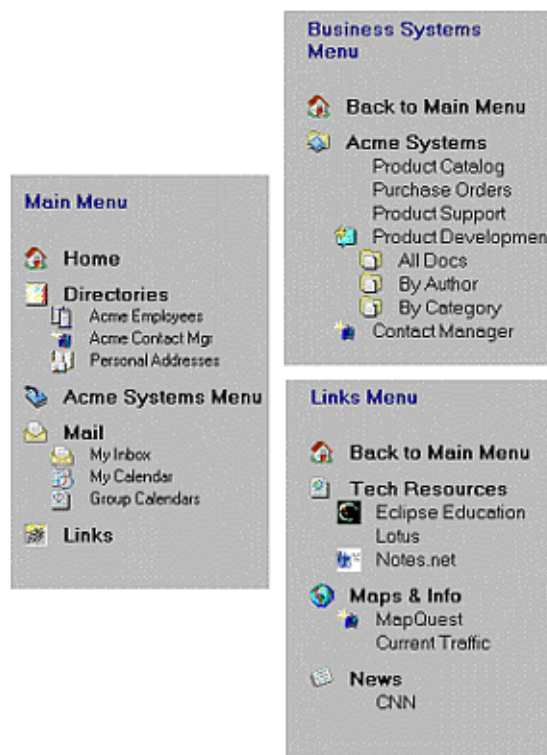
1. Identify the applications to include
2. Create a blank database
3. Import image resources and other items that can be reused
4. Create outlines and pages
5. Create the standard navigational frameset
6. Implement the standardization

By following these steps, I created the following standardized front end for Acme's applications, which has been specified as the user's Welcome page. The complete database, [Acme Standard Interface](#) (homebase.nsf), is posted in the Iris Sandbox.



Information such as the user's mail, headlines, and the content of databases appears on the right. The Main Menu on the left provides access to the user's mail and directory options as well as to submenus of the company's important

business applications and links to commonly used Web sites.



Step 1: Identifying the applications to include

The first step is to identify the applications that need to be included in the standardization. Other applications can be added to the list over time, but you need an initial list.

At Acme, there are a number of databases and information facilities that are used every day by nearly all users:

- Acme Customer/Contact Management database (which was written in version 4.6)
- Purchase Order database
- Product Catalog
- New Products Discussion
- Acme customer and vendor Web sites

In addition to their business databases, they want to include the Acme Domino Directory (which is used as their employee directory) and each user's personal mail databases and Address Books. This may sound like a challenge, but Designer will make this task very simple.

Step 2: Creating a blank database

Next you must create a blank database in a commonly accessible location. Don't worry about defining the ACL just yet (other than making the default something other than Designer); I'll go into more detail on that topic later.

Step 3: Importing image resources and other items you can reuse

During the course of designing your standard front end, there will be a number of items that you can reuse from the applications. Finding these elements will require a quick visit to the designs of the applications you are including, but you won't be modifying those databases.

Items that can be reused might include:

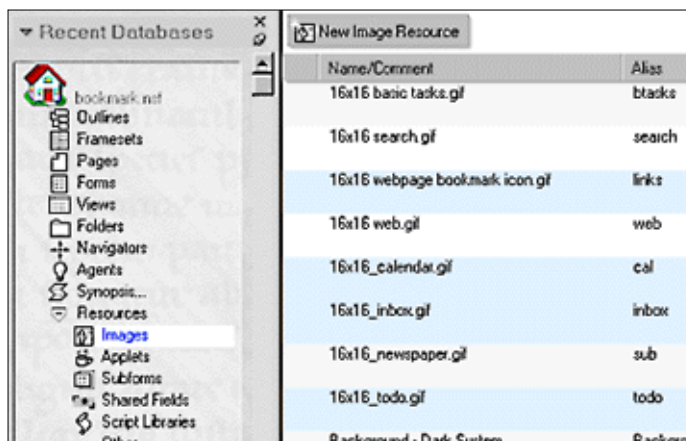
- Images that need to be incorporated into the image resources, such as

company logos, application logos, or other images that will make the user interface more graphical and user-friendly. These may exist as separate graphics files or be image resources in the applications' databases.

- Image resources currently in other Notes databases, for example, resources in the Bookmarks database.
- Outlines, pages, forms, and framesets that may serve as a framework or be useful in the new front end. For example, if an outline exists in one of the newer application databases that users like and are comfortable with using, you might use it as a starting point for the new standard user interface.

You can't "point to" image resources or other design elements—such as outlines, pages, forms, and framesets—that exist in the other applications' databases. Instead, you copy them from the existing database and paste them into the new Standard Interface database:

1. With Designer, open the database containing the element you want to copy.
2. Select the type of element to be copied. For example, to copy image resources, expand the Resources section of the Design pane and select Image.
3. Select the items to be copied. Remember that you can hold down the Ctrl key to mark multiple items.

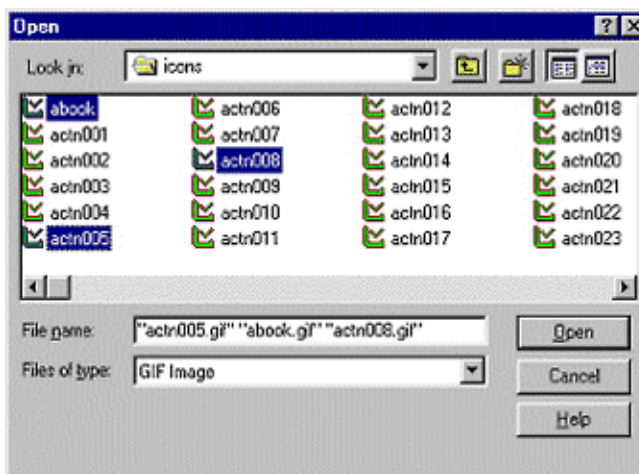


4. Copy the items to the Clipboard.
5. With Designer, open the Standard Interface database to the same type of element, and paste the Clipboard contents into it.

For the Acme example, I've copied a number of image resources from the bookmark.nsf file into the Standard Interface database. However, there were a few images from the Domino icons directory that I also wanted to use in the new interface. I needed to bring them into the database as new image resources.

To add new image files to the database as image resources:

1. With the Standard Interface database open in Designer, expand the Resources section in the Design pane and select Images. The list of existing image resources appears.
2. Click the New Image Resource button.
3. In the Open dialog box, specify the type of file you want to insert (GIF, BMP, or JPEG) in the Files of type field.
4. In the Open dialog box, select one or more graphic files you want to include as image resources.



5. Click Open. Designer creates an image resource entry for each file you selected.

Step 4: Creating outlines and pages

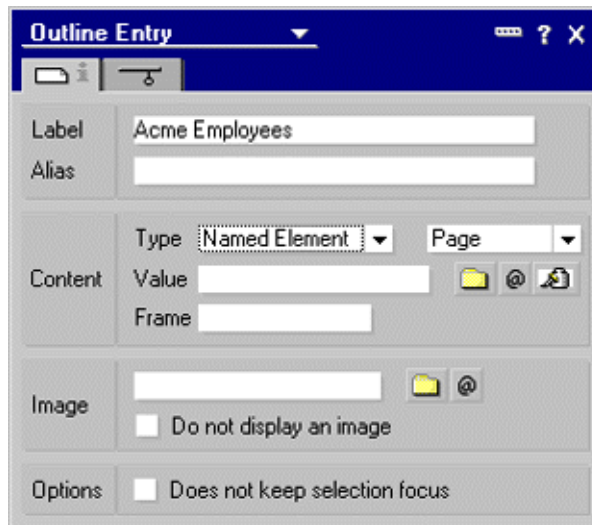
The outline design element in Domino R5 is your most valued organizational tool for this effort. As the title might suggest, outlines allow for the quick collection and organization of different navigational items in Notes/Domino applications. By creating an outline entry, you can create a single-click link to a view, form, page, folder, frameset, or navigator—all of the display components contained in a Notes/Domino application. (For more detailed information on designing and using outlines, see the *Iris Today* article [Domino Designer R5: The Outline Designer](#).)

One frequently overlooked option for outline entries is their ability to link to views, forms, and other navigational items in other databases, that is, outside of the current database. You can use this capability to create a navigational menu that links to the applications you listed in step 1.

Creating the outline entries

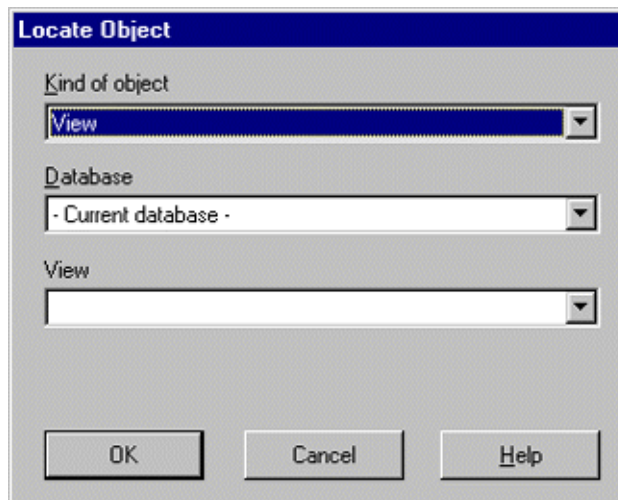
For example, to include an outline entry that navigates to the People view of the company's Domino Directory:

1. With the outline open in Designer, click the New Entry button to see the Outline Entry properties box.
2. Enter a label for the entry in the Label field.
3. Select Named Element in the Type field. Another field appears with a drop-down list of elements to which the entry can link, including page, form, view, frameset, folder, or navigator.



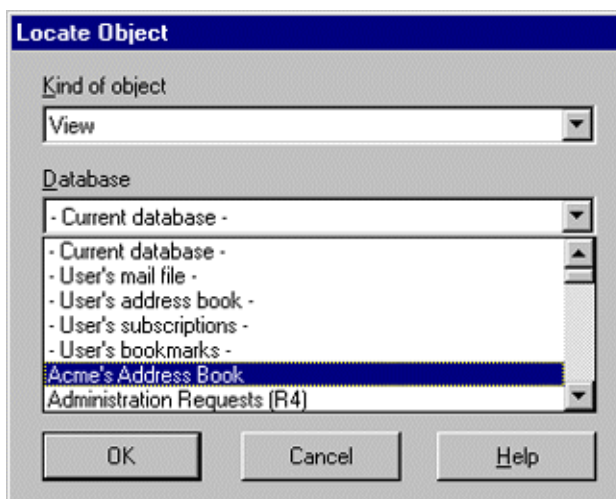
The 'Outline Entry' dialog box is shown with a blue title bar and standard window controls. It contains several input fields and checkboxes. The 'Label' field is filled with 'Acme Employees'. The 'Type' dropdown is set to 'Named Element'. The 'Page' dropdown is also visible. Under the 'Content' section, there are 'Value' and 'Frame' fields, both empty, with browse buttons (folder and @ icons) to their right. The 'Image' section has an empty field with a browse button and a checkbox labeled 'Do not display an image'. The 'Options' section at the bottom has a checkbox labeled 'Does not keep selection focus'.

4. Select the type of element you want to link to from the list. For this example, you would select View.
Note: Alternatively, you can skip this step and set the element type in the Kind of object field in the Locate Object dialog box shown below.
5. Click the Browse button next to the Value field to open the Locate Object dialog box.



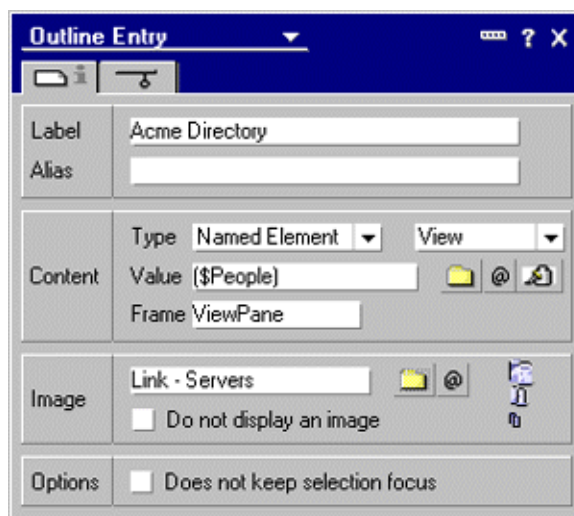
The 'Locate Object' dialog box has a blue title bar. It features three dropdown menus: 'Kind of object' (set to 'View'), 'Database' (set to '- Current database -'), and 'View' (empty). At the bottom, there are three buttons: 'OK', 'Cancel', and 'Help'.

The default Database option is Current Database, but there are more sources for objects available. The database list will include databases for which you have bookmarks. Designer also provides shortcuts to user-specific links such as mail, making the task of determining which user mail file or address book to open much less challenging for the programmer.



6. In the Database field, select the database you want to link to from the list, for example, Acme's Address Book.
7. In the View Name field, select the view you want to link to from the list, for example, (\$People).
8. Click OK.
9. In the Frame field, enter the name of the frame in which this outline entry will appear. In this example, I use the name ViewPane. I'll discuss the use of frames and framesets later in the article.
10. In order to display a graphic for this entry on the outline, you can specify an icon from the image resources by clicking the Browse button next to the Image field and selecting the appropriate graphic from the Insert Image Resource dialog box.

Here are the properties of the completed example outline entry:

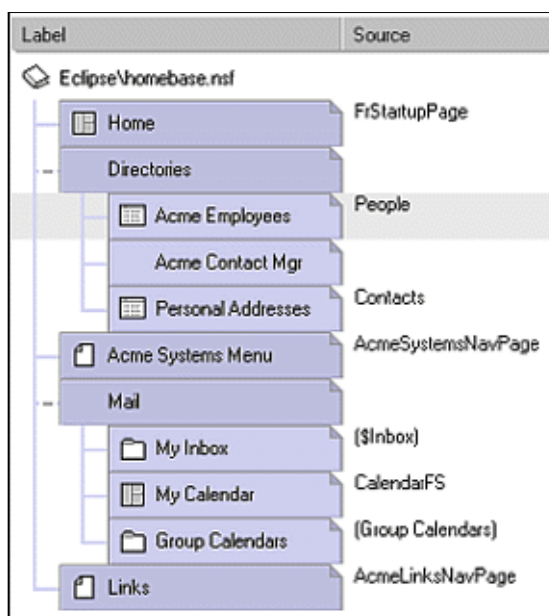


To build the outline that will navigate to the various company databases, you simply add this type of outline entry for each database item you want to include in your navigation. If the "mission" of this effort is to create a standard look and feel for all applications in the system, then it's probably a good idea to include outline entry links for all of the views in the included databases (or at least the ones that everybody uses). You also have the ability to selectively hide entries within the outline (I'll get to that shortly), so keep this in mind as you are deciding what links should or should not be included in your application outlines.

Developing multiple outlines

The outline you are developing becomes the menu for the interface. Ideally, all of the menu options could appear in one outline so that all of them are always visible to the user. However, you may find that you need more than one outline in your interface, to either keep your menu to a specific page length or to provide natural breaks in the functions to be displayed.

In the Acme example, including all of the links to the different Acme databases would cause a single menu to run beyond the bottom of the screen. Therefore, I created separate outlines for both the Acme Systems menu and the Links menu. To display these submenus, I created two outline entries in the Main Menu—Acme Systems Menu and Links, as shown in the following screen. These load the appropriate submenu pages. (These pages are discussed in the next section). The only significant variation for these two outline entries is the value in their Frame field; instead of using ViewPane, their Frame field is set to NavPane.



Creating the pages for the outlines

Outlines are great organizational tools, but they can't be displayed independently. They must be embedded into a page or a form. Once you're finished creating your outlines, you can create the pages and/or forms that will serve as the display "backgrounds" for your newly-created outlines.

Because the outlines for the example do not incorporate any fields, I chose to create pages to display them. To embed an outline in a page:

1. Open or create the page.
2. Position the cursor where you want the outline to appear.
3. Choose Create - Embedded element - Outline.
4. Select the outline you want to embed from the list in the Insert Embedded Outline dialog box and click OK. Designer embeds the outline on the page.
5. Choose the appropriate formatting options, such as alignment, font type, color, and so on. (For more information about the formatting options, see the *Iris Today* article [Domino Designer R5: The Outline Designer](#).)

Step 5: Creating the standard navigational frameset

Once you have assembled the outlines and their pages and forms, you need to create the framesets to display the information. (For information about

creating framesets, see the *Iris Today* article [Domino Designer R5: Framesets](#).) When planning and developing the framesets for a standard interface, you need to evaluate how your new navigational components—the outlines and pages—should be presented to users.

In this case, there needs to be one frameset to serve as the overall display structure. The menus should appear on the left side of the screen and the content associated with the menu selections, on the right. Therefore, the main frameset used in the Acme system is two frames wide; the left side is dedicated to navigation (titled NavPane), and the right (titled ViewPane) is for viewing what's selected in the navigation pane.

Remember that when you created the outline entries, you specified the target frame in which each element opens by setting the Frame field to ViewPane or NavPane. By setting that property for each outline entry, you control where the information is displayed—menus on the left (NavPane) and content on the right (ViewPane)

Additionally, to display a wider variety of information when the Standard Interface database is first opened, you can include a "startup page" frameset that incorporates information from a News/Headlines database, the user's Inbox, and the user's Calendar for the day. This startup page frameset can appear in the ViewPane of the main frameset. By using a frameset within a frameset, you can maintain the basic, two-sided display from the initial opening of the database through any selections of items from the navigational pane.

Step 6: Implementation

Now that all of the navigational pieces are in place, all that's left to do is set up the security for the database, set the database Launch options to display the appropriate frameset when the application is opened, and figure out a way to distribute the new Standard Interface database.

Setting the Launch Option is straightforward; simply go to the Launch tab of the database properties box, select Open designated Frameset, and specify the frameset. The other two items are a little more open to interpretation depending on your environment. Here are a few suggested guidelines.

ACL settings

The default ACL settings for the Standard Interface database should be Reader, but you should also designate someone to act as system administrator who has Editor rights to the database. This administrator is responsible for maintaining a System Profile document, which contains some group membership information that determines the user's ability to see certain menu options. (I'll cover this "menu filtering" a little later in the article.)

Deployment

In keeping with the "make it as easy as possible for the user" philosophy, one particularly effective way to deploy the Standard Interface database is by sending a mail message with a link to the database. Although the mail administrators may not be overly thrilled with this suggestion, the end user support/help desk staff should benefit. This offers the opportunity for explanation and instructions, without creating a great deal of work for your users and support staff.

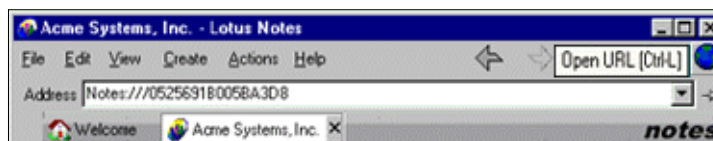
The link to the database can be delivered in several different ways:

- You can insert a **Database link** (similar to a Document link) into the mail message and instruct the user to click on the link to open the system.
- Embedding an **installation button** in the body of a mail message is also a way to deliver the database. The button can contain code (such as `@Command([AddBookmark])`) that will make it easy for the user to incorporate the database into their Bookmark bar (or a folder contained within the bar). For example, to install this item as a bookmark called

Acme Systems, the syntax for the button code would look like this:

```
@Command( [AddBookmark] ; "Notes:///0525691B005BA3D8" ; "Acme Systems" ; "" )
```

Note: You'll need to insert the appropriate URL string for your Standard Interface database, which can be obtained in two ways. You can use the Replica ID for the database, minus the separating ":" and preceded by "Notes://", or you can open the Standard Interface database to the default frameset, click the Open URL button in the upper-left corner, and then click the drop-down arrow twice to display the URL for the database.



- If your administrators have created **User Setup Profiles**, adding new bookmarks and replica databases to users' desktops becomes a very simple task. For details about creating User Setup Profiles and using them to add bookmarks, see the *Iris Today* article [Centrally Managing the Desktop](#).

Once the bookmark is added to their desktop, end users can make this item their Notes Home/Welcome page with just a few clicks:

1. Right-click the bookmark.
2. Select Set Bookmark as Home Page from the menu. The link will now be the user's starting point each time the Notes client is started.

Adding business value to the effort

Now that the basic application front end is in place, you can build some additional business value into the system, introducing items that help draw users to the new entry point for the business systems. This can be done in a variety of ways, depending on your business's needs. Here are a few suggestions.

The bulletin board

One quick answer might be to incorporate an electronic bulletin board offering a place to post company-wide memos or notices, and having the ability to identify specific documents as "headlines" to be displayed in the standard front end.

Using the Discussion template as a starting point, I created an [Acme News database](#) to function as a bulletin board. (It also is posted in the Iris Sandbox.) To keep matters simple, only a few changes were made to the Discussion template:

- A news/headline flag field was added to the Main Topic form.

- A hidden view was created to display only documents flagged for display in the Headlines view.

- The Interest Profile form was modified, adding a preferences field to allow each user to filter their headlines based on a news category.
- A new form containing the Headlines view as an embedded element was added.

The News database can be used as a repository for announcements, reminders, or anything else worth communicating to others in the company, without contributing to mail file database sizes. In addition, the task of maintaining the headlines listing can now be delegated to someone other than a developer. By restricting the headlines flag field's use to allow only Editors and above, you can allow anyone to post to the News database, but only those designated as Editors can enable postings for display within the new standard front end.

Web links

A separate outline containing URL links to other Web sites (with a reference entry on the Main Menu) might be of great use, particularly for gathering and organizing sites of interest to your business. Some items worthy of consideration might be:

- Industry news Web sites
- Customer or vendor Web sites
- Competitor Web sites, for keeping track of the latest press releases that may affect your business
- Reference sites, for quick access to weather reports, maps, or other useful information

You can see these types of items in the Links submenu of the Acme Standard Interface:



Menu filters

In many cases, certain menu items may need to be hidden based on the user of the system. You can easily manage this within the outline entries themselves, using the Hide tab in the Outline Entry's properties box.

To make the hide conditions a little more dynamic and less programmer-dependent for changes, you can use a System Profile document to house the names of the users who are granted access to "secure" menu options.

You can create a System Profile form and make it accessible from the Actions menu via an agent called System Administration. Then authorized users can create a System Profile document and assign access to the different secure menu items to the appropriate individuals and groups.

Acme Systems, Inc. System Profile Document	
Secure Menu Options Access Rights:	
Product Development Menu access is granted to:	Sam Bridegroom/Eclipse-Education
Management Menu access is granted to:	Sam Bridegroom/Eclipse-Education

Who can create and edit the System Profile document? Through the use of an ACL role (called ProfileMgr in this example), you can restrict create and edit capabilities to only the members of that role.

OtherDomainServers	<input checked="" type="checkbox"/> Create personal folders/views <input checked="" type="checkbox"/> Create shared folders/views <input checked="" type="checkbox"/> Create LotusScript/Java agent <input checked="" type="checkbox"/> Read public documents <input checked="" type="checkbox"/> Write public documents Roles: <input checked="" type="checkbox"/> (ProfileMgr)
Sam Bridegroom/Eclipse-Education	
Sherry Weinstein/Eclipse-Education	
Steven Barefoot/Eclipse-Education	
Tammy Dixon/Eclipse-Education	

Finally, you need to write the Hide formula in the Outline Entry properties box, to prevent those users who are not a member of the "secure" list (in this example, a member of the Product Development Menu access list) stored in the System Profile document from seeing that menu option.

Outline Entry
Hide outline entry from <input type="checkbox"/> Notes R4.6 or later <input type="checkbox"/> Web Browsers <input checked="" type="checkbox"/> Hide outline entry if formula is true Formula Window
<pre>!@IsMember(@UserName;@GetProfileField("SystemProfile";"ProductDevAccess"))</pre>

You may be wondering if you could use a Group document to restrict this access? Group security in many cases doesn't translate well to traveling replica databases, where a user may or may not have the appropriate groups in his or her Personal Address Book. Although using a System Profile document creates some dual maintenance of lists, it will perform more reliably when you have remote users than using a Group document.

Remote users

And speaking of remote users, what happens when a user takes this standardized front-end system on the road as a collection of local replicas? In most cases, the transition from connected to disconnected use will be nearly seamless. Remember, the outline designer uses the bookmarks list as its source for the database list. The Notes client's ability to adapt to its location and connection settings—and subsequently render the appropriate replica databases—makes your new, standardized front end a very portable solution with little or no programming effort.

Ongoing system maintenance

As we've gone through this exercise, you've probably noticed how easy it is to

assemble a front-end system like this. So what happens when a new database application is added to the environment and needs to be included in the standard front end? It's just a matter of adding new entries to the outline, a relatively easy task.

Programming caveats

Although this solution is a fast and relatively straightforward solution to standardizing the appearance of multiple applications, it doesn't come without its development "gotchas." Here are a few items to keep in mind as you develop your standardized front-end system.

Buried code in hotspots

Many applications that use navigators or hotspots often have code behind these elements, which may be critical to how a particular application is intended to function. Be sure to test your standardized user interface thoroughly, including having a few end users try it out, to uncover these types of items.

Web support

The Acme example is presented with the assumption that the company's applications are all Notes client-based. With that, there may be functions that don't translate well to the Web. When developing the front end, be sure to keep your client audience—whether it's Notes clients, Web browsers, or both—in mind. And be sure to test the application using the different clients to ensure everything is working properly.

Availability of Internet access

Finally, keep in mind that Web links won't work very well if Internet facilities are not available to the machine, unless offline browsing capabilities are configured on the client machine. This may be part of the installation instructions you send when deploying the database.

Conclusion

You now have a great starting point for standardizing your company's user interface for its business applications. Perhaps the feedback received from spending a day or two assembling this type of framework and testing it with the user community will be helpful in the design of future releases of your business applications. Will this prevent users from accessing the databases directly through File - Database - Open, the Notes workspace, or from an existing bookmark? No, it won't; but it will provide them with a much more structured, compact, and easy-to-use facility to access a wide variety of company data.

ABOUT THE AUTHOR

Sam Bridegroom is an instructor with [Eclipse Education](http://www.Eclipse-Education.com), a Lotus Authorized Education Center in Indianapolis, Indiana, specializing in Lotus Notes and Domino R5 courses.

Sam holds both a System Administration and Application Development CLP, and has recently fulfilled the requirements for his Certified Lotus Instructor status. While spending three years managing his own consulting and contract development practice (after two years of implementation and development of Notes at another company), he has managed or collaborated on a wide variety of Notes/Domino projects ranging from small development efforts to integration of back-end database systems with Notes and Domino. Sam has a B.S. in Finance from the Kelley School of Business at Indiana University. He and his wife Cathy enjoy hiking and bicycling, usually using those opportunities to expand their combined photography portfolio. You can reach Sam at Sam@Eclipse-Education.com.