PeopleSoft®

EnterpriseOne Form Design Aid 8.9 PeopleBook

September 2003

EnterpriseOne
Form Design Aid 8.9 PeopleBook
SKU RELEFD0309

Printed in the United States.

This documentation is subject to change without notice, and PeopleSoft does not warrant that the material contained in this documentation is free of errors. Any errors found in this document should be reported to PeopleSoft in writing.

Open Source Disclosure

PeopleSoft takes no responsibility for its use or distribution of any open source or shareware software or documentation and disclaims any and all liability or damages resulting from use of said software or documentation.

Table of Contents

Interactive Application Design

Interactive Application Design is the entry point to several tools for creating, generating, running, maintaining, and securing applications. Interactive Application Design includes Form Design for creating forms and Event Rules Design for attaching business logic through event rules. Use Interactive Application Design to do the following:

- Access Form Design for creating forms
- Run an application
- Create text overrides
- Browse ER
- Browse forms in an application
- View the version list
- Use Visual ER Compare to compare event rules between two versions of an application
- Use FDA Compare to compare one version of an application with another

You can use the J.D. Edwards toolset to create applications that run in a Windows client, an HTML client, or both.

To start Interactive Application Design, choose an application in Object Management Workbench and click the Design button. In Interactive Application Design, you can change the metadata for the application. To access application metadata, click the Summary, Category Codes, and Install/Merge Codes tabs. You can also attach text and files to an application by clicking the Attachments tab. You can access all other functions from the Design Tools tab.

See Also

- ❑ *Object Management Workbench* in the *Object Management Workbench Guide* for information about how to use the Object Management Workbench
- ❑ *Working with Attachments* in the *Object Management Workbench Guide* for instructions for attaching text and files to an application (or any OMW object)
- ❑ *Designing Web Applications* in the *J.D. Edwards Web Client Guide* for more information about developing Web applications

Understanding Applications

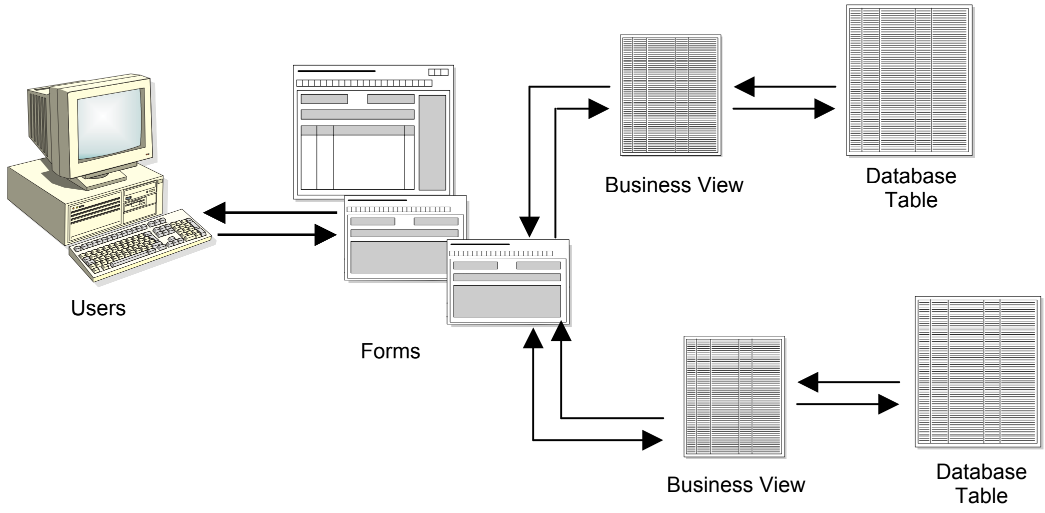
An application is a collection of computer programs that performs a specific task. An application retrieves and updates data within a database table. Examples of applications include P01012 (Address Book) and P4210 (Sales Order Entry).

J.D. Edwards software includes the following types of applications:

- Interactive applications

  An interactive application is the interface between a user and a database table, based on a business view. A user uses an interactive application to add, modify, or view data using a form.

Interactive Application



You use the same tool to create an application for Windows and for HTML.

- Batch applications

  A batch application processes automatically without user interaction. Table conversions and reports are examples of batch processes.

Adding an Interactive Application

An application is an object. Before you can begin developing an application, you must add it so that it exists as an object. You can add a new application or you can add a version of an existing application.

After the application exists as an object, you can start building the components of the application. You can use Form Design to design the first form in your application.

Interactive Applications

The Object Management Workbench (OMW) name for an application can be a maximum of eight characters. Although the software accepts up to 10 characters, if you enter more than eight characters the entry will be truncated. Format the name as *Pxxxxyyy*, where:

P = Application

xxxx = The system code

yyy = A next number, such as 001 and 002

Ensure that you provide a description of up to 60 characters. The description should reflect the subject of the forms within the application; for example, Companies and Constants.

Naming Conventions for Forms

Form Design Aid automatically assigns a name to the form using the format *WzzzzzzzzA*, where:

> *W* = Form.
>
> *zzzzzzzz* = The application name.
>
> *A* = The first form created in the application. It is usually, but not always, the entry point to the application. Subsequent forms are assigned sequential letters, such as B for the second form, C for the third form, and so on.
>
> For example, the application P0101 has two forms. The first form, Work With Addresses, is assigned the name W0101A. The second form, Address Book Revisions, is assigned the name W0101B.

Ensure that you provide a form description that is based on the form type. For example:

Find/Browse	*Work With* followed by the subject of the application, such as *Work With Companies*, *Work With Constants*.
Fix/Inspect, Header/Detail, and Headerless/Detail	A title that reflects the topic of the form, such as *Supplier Information*, *Item Master Revisions*, *Purchase Order Entry*.
Lower-Level Windows	A title that reflects the topic of the window, with the title of the calling form appended to it, such as *Enter Voucher - G/L Distribution*. When the title of a window includes a verb, use an active verb, such as *Work With Vouchers*, instead of a nominalization.

Form Interconnection Data Structures

The J.D. Edwards tool set automatically creates form interconnection data structures using the key fields in the business view.

You should change the data item name and description to describe the item that is passed between forms.

Because Message forms do not have data structures, you must add at least the one data structure.

► To create an interactive application object

In the OneWorld Menus task view, choose the OneWorld Tools node. Expand the OneWorld Tools node and then the Object Management node.

1. Launch Object Management Workbench.
2. On Object Management Workbench, click Add.
3. On Add OneWorld Object to the Project, choose the Interactive Application option, and then click OK.
4. On Add Object, complete the following fields, and then click OK:
    - Object Name

This field accepts up to 10 characters; however, if you enter more than 8 characters, the entry will be truncated.

- Description

  Provide up to a 60-character description. It should reflect the subject of the forms within the application, such as Companies and Constants.

- Product Code
- Product System Code
- Object Use

► To create an interactive version object

In the OneWorld Menus task view, choose the OneWorld Tools node. Expand the OneWorld Tools node and then the Object Management node.

1. Launch Object Management Workbench.
2. On Object Management Workbench, click Add.
3. On Add OneWorld Object to the Project, choose the Interactive Version option, and then click OK.
4. On Adding a Version, enter the object name of the application upon which you want to base this version.
5. On Version Add, complete the following fields, and then click OK:

   - Version

This field accepts up to 10 characters; however, if you enter more than 8 characters, the entry will be truncated.

- Version Title

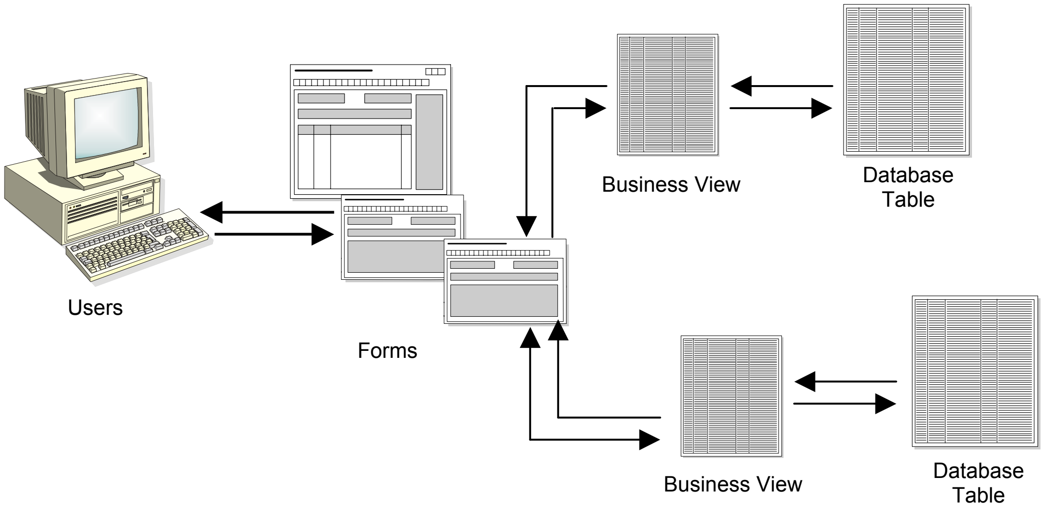
  Provide up to a 60-character description. It should reflect the subject of the forms within the application, such as Companies and Constants.

- Prompting Options
- Security

Form Design

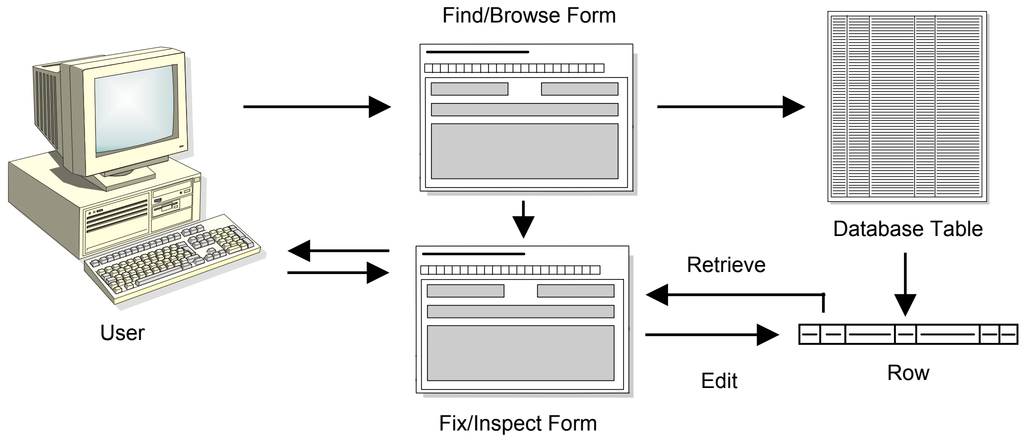
Use Form Design to create one or more forms for an application. A form is the interface between a user and a table. This interface should present the data logically and contain the functions that are necessary to enter and manipulate data.

Interactive Application



A single application can contain one or more forms. Usually, a Find/Browse form is the first form that appears in the application. It enables the user to locate a specific record with which he or she wants to work. Upon selecting a record, a subsequent form, such as a Fix/Inspect form, automatically appears, on which the user can view or modify data for that record.

Two Tiers of Forms



A form has the following elements:

Form Type

The form type establishes the basic purpose of a form. Each form type has default controls and processes.

Business View

In an application, a business view links forms and tables efficiently by providing access to only that data required by the application. For example, if you have two tables with twenty columns each and your application only needs to access one column from one table and two columns from the other, you can make a business view that contains only those three columns. Your application is more efficient because searches are limited only to those three columns, but your application still updates the actual tables when necessary. You must associate all forms, except the Message form, with a business view.

Controls

All objects on a form are controls. Controls include grids, check boxes, radio buttons, push buttons, and more.

Properties

In application design, the following types of properties exist: application, form, control, and grid. Properties define appearance and function.

Data Structure

A data structure defines the data that can be passed between forms within an application or between applications. After the data structure of a form is defined, use form interconnections to indicate the direction of the flow of data between forms.

Event Rules

Event rules can contain processing instructions for specific events. Events are actions that occur on a form, such as clicking a button or using the Tab key to move out of a field. Use event rules to attach business logic to any event.

Events are triggered either as a result of user interaction with a control, such as clicking a button, or as a result of a system-controlled process, such as loading a grid.

Creating a Form

You use Form Design to create one or more forms that appear in an application. These forms are the visual interface for the end-user of your application and enable that user to view, add, or modify data that is stored in one or more tables.

After you create the form, you can modify the system data for the form, such as its metadata, Help ID, and so forth.

Before You Begin

- □ Acquire a working knowledge of Windows and have an understanding of the J.D. Edwards application design process.
- □ Determine which data items are required for each form in the application.
- □ Determine whether any of the existing tables suit your needs. If not, create a table.
- □ Create a business view on which to base a form.

Recommended Form Design Configuration

Form Design offers a variety of toolbars and panes that you can place on the desktop while you work. These objects are described below. J.D. Edwards recommendations for using these objects follow.

Object	Notes	To View
Main Toolbar	Provides standard actions such as open, cut, and paste.	Choose: View > Toolbars > Main Toolbar
Control Modes Toolbar	Allows you to switch between modes while designing your form.	Choose: View > Toolbars > Control Modes
Layout Toolbar	Provides functions to align and space form controls precisely.	Choose: View > Toolbars > Layout
Insert Controls Toolbar	Allows you to place controls on a form with the push of a button.	Choose: View > Toolbars > Insert Controls
Status Bar	Displays a status bar at the bottom of the tool.	Choose: View > Status Bar
Application Tree View	Displays a pane showing the objects in the current application and their relationship to each other. You can open an object in this pane by double-clicking it.	Choose: View > Application Tree View
Property Browser	Displays a pane showing the properties of the selected object. You can change an object's properties in this pane.	Choose: View > Property Browser
Data Dictionary Browser	Displays a pane where you can search for data dictionary objects. You can drag and drop data dictionary objects from this pane to your form.	Choose: View > Data Dictionary Browser
Business View Column Browser	Displays a pane where you can search for columns in a business view. You can drag and drop objects from this pane to your form.	Choose: View > Business View Column Browser

Recommendations

- Display the Main, Layout, and Insert Controls toolbars at all times.
- Use the Insert Controls toolbar to insert controls on a form.
- Display the Application Tree View whenever possible. The pane helps you keep the entire application in mind, and it provides an easy way to open other objects in the application.
- Display the Property Browser at all times and use it to set the properties for all objects.

► To create a form

This task provides a general overview of the typical form creation process. Many of the steps in this task are described in greater detail in other topics.

In Object Management Workbench, create an application or open an existing one, and then start Form Design Aid. Alternatively, launch Form Design and create or open an application from the File menu.

1. On Form Design, choose Create from the Form menu, and then choose the type of form you want to create.

   The system creates a blank form of the type you chose and displays the Properties form for the form.

2. Complete Form Properties as appropriate and click OK.
3. If required, attach one or more business views to the form by choosing Business View from the Form menu and then choosing Add Business View.

   Headerless Detail forms allow you to attach two business views to them; all other forms, except for Message forms, allow you to attach only one. You cannot attach a business view to a Message form.

4. Add and configure controls to the form, as required.

   Set the Data Item Information property for a control to attach data dictionary items or business view columns to it. To attach event rules (ER) to a control, right-click the control and choose Event Rules. To set text variables for a control, right-click the control and choose Text Variables.

5. Arrange the controls on the form so that they line up and are equally spaced, and then resize the form to fit.
6. Add menu/toolbar exits to the form.
7. Save and test the form.

See Also

- ❑ *Form Properties* in the *Form Design Aid Guide* for a description of the purpose of the different property values, along with a discussion of why you might enable or disable some values.
- ❑ *Menu/Toolbar Exits* in the *Form Design Aid Guide* for instructions on how to add menus and toolbar exits to a form.

► To delete a form

1. On Form Design Aid, choose Destroy from the Form menu.
2. Confirm the deletion.

Caution

After you delete a form, you cannot retrieve it.

Alternatively, you can delete a form on Work with Forms by choosing the form to delete and then clicking Delete.

Form Properties

The property settings for a form control its appearance, how it displays errors, and how it interacts with its underlying tables. When you first create a form, the system prompts you to configure its properties. You can change a form’s properties later in the design process as well.

Forms have the following property values:

- Event Rules Title

  The user sees this text for the item as it is displayed in the application. The & indicates by the letter that follows it the keyboard shortcut. For example, $File can be activated by the user from the keyboard with Alt+F.

- Entry Point

  Check this box if the form is the first form a user sees upon launching the application. If you do not assign one of your forms as the entry point, the runtime engine loads the first form it finds in the application. The system does not allow you to set more than one form as an entry point. Fix/Inspect forms cannot be used as entry points.

- Enable In-Your-Face-Errors Display

  This property affects HTML forms only. Typically, the system indicates an application error by highlighting the Errors and Warnings hyperlink in the upper right area of the application. When enabled, however, the In-Your-Face-Errors property causes application errors to appear on the Web page. Enabling In-Your-Face-Errors has no noticeable effect on performance.

- Transaction

  When you enable Transaction, the system commits all changes at one time instead of individually. If the form makes a single database change or a group of unrelated changes, do not enable Transaction. However, if the form has a group of inserts that rely on each other and if the system should revert to the previously committed changes if a change fails or for another condition, then enable Transaction.

- Size
- Position
- Update on form business view

  When enabled, the system updates the tables underlying the form (except for those underlying the grid control) when the user clicks OK. If you want greater control over table updates, disable this option.

- Update on grid business view

  When enabled, the system updates the tables underlying the grid control when the user clicks OK. If you want greater control over table updates, disable this option.

- Fetch on form business view

  When enabled, the system updates the form based on information in the tables underlying the form (except for those underlying the grid control) when the user clicks OK. If you want greater control over form updates, disable this option.

- Fetch on grid business view

  When enabled, the system updates the grid based on information in the tables underlying the grid when the user clicks OK. If you want greater control over grid updates, disable this option.

- End form on add

  When enabled, the system returns to the previous form after a user adds a record and clicks OK. If you want greater control over form flow, disable this option.

- Form Guide

  Form Design indicates the optimum form size for different platforms by superimposing light blue lines that indicate height and width along the top and left-hand side of the form. You can choose from a number of different platforms, or you can indicate your own dimensions for the guide. You can automatically size your form to the guide dimensions by choosing Size to Guide from the Layout menu. Use the Form Guide option from the Layout menu to hide or show the form guides.

- Show Wallpaper

  When enabled, you can designate a bitmap to be used as the background image for the form.

Note

These properties are the ones that appear on the Form Properties form. You might see other property values for a form in the Property Browser. Click the + symbol at the bottom of the Property Browser to see a description for each property type, or right-click and choose What's This from the pop-up menu.

► To set properties for a form

You can access the Form Properties in several different ways. Create a form for your project or on an existing form and perform one of the following, to add or change the values for the form properties that you need.

- Create a form and Form Properties appears.
- Choose Property Browser from the View menu, and then click the form header (being sure not to click a control on the form) to display the form property values in the Property Browser. You can change the properties for the form directly in the Property Browser pane.
- Right-click the form (being sure not to click a control on the form) and choose Properties from the pop-up menu.
- Choose Application Tree View from the View menu, and then, in the Application Tree View pane, either double-click the form name or right-click the form name and choose Properties from the pop-up menu.
- Click the form and choose Form Properties from the Form menu.

► To revise system information about a form

You can assign and change system data for a form, such as the description of a form and its help identifier, as well as metadata, such as its product code.

1. On Interactive Application Design, click the Design Tools tab, and then click View Forms.
2. On Work with Forms, choose one of the following options:
   - Local Forms
   - Checked-in Forms

When you create a form, it is local to your machine until you check in the form. When you check in the form, the system updates the Form Information File table (F9865). You can view forms that have been checked in or forms that are local to your machine.

3. In the detail area, choose one of the available forms to view or revise.
4. On Form Information Revisions, review or revise any system information needed.

► To attach a business view to a form

You can attach two business views to a Header Detail form. You can attach one business view to all other form types. Use either of the following methods to attach a business view to a form.

In the Property Browser pane:

a. Click the Business View Name field.
b. Click the ellipses button that appears in the field.
c. On Individual Object Search and Select, in the detail area, choose the business view that you want to attach to the form and click OK.

Alternatively,

In the Form Design desktop:

a. Choose Business View from the Form menu, and then choose Add a Business View.
b. On Individual Object Search and Select, in the detail area, choose the business view that you want to attach to the form and click OK.

► To include a picture on a form

Insert an Image control on your form. When you set up the control properties, indicate the location of the picture.

With the Image control, you can place static or animated graphics on a form. You can also endow the graphic with special qualities, such as making it clickable.

► To allow users to attach objects to a form

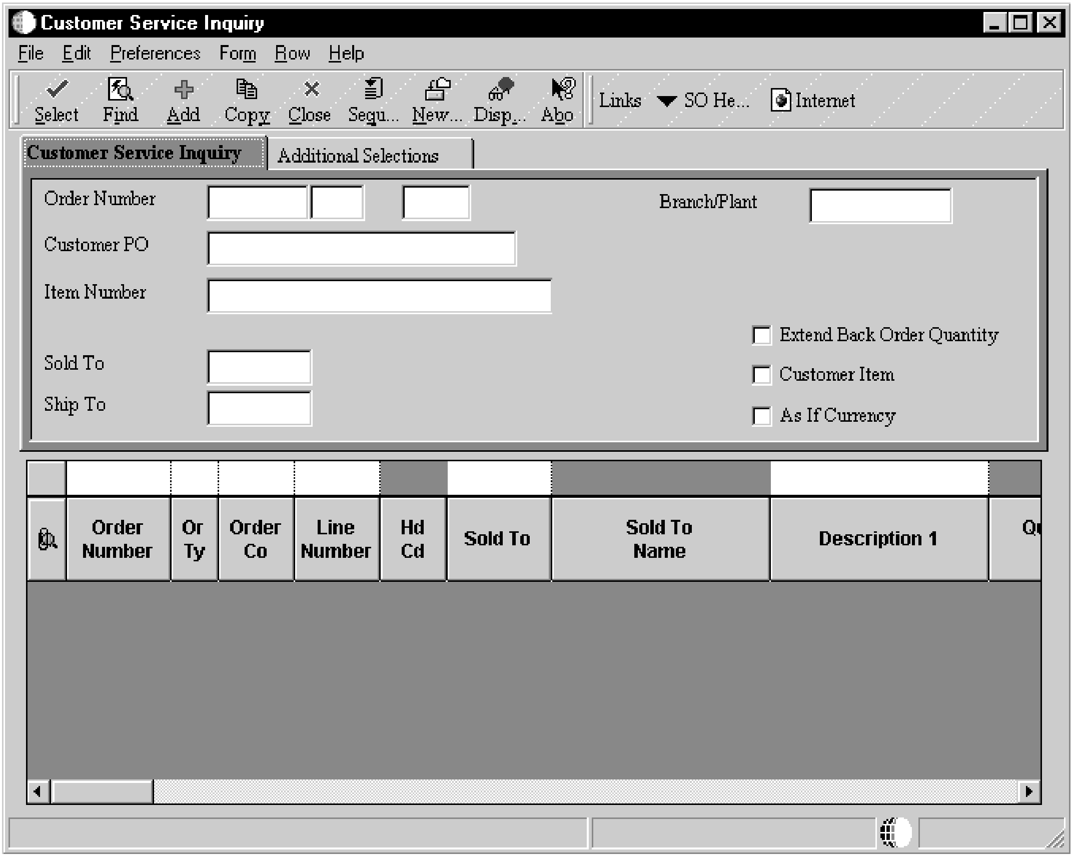
Insert a Media Object control on the form. This control allows users to attach text and other file types to a form for their own use. For example, a user might attach a job aid to a form.

Putting Tabs on a Form

You can create a control that allows you to split a form into different tabbed pages. This strategy allows you to use more controls for a single form. You can group your control functions by placing related controls on different tab pages for a single form. You can cut and paste controls from one page to other pages.

The form has a single business view. One commit for the form exists on the OK button. You can use system functions, such as Set Focus, to add additional functions for the tab controls. Each tab page has a Tab Page is Selected event and a Tab Page is Initialized event associated with it. You can attach additional event rule logic to these events. When you use tab pages in an application, you can focus on the upper-right corner of the tab page and move it around. This strategy allows you to see several pages at the same time.

The following illustration shows a form that has tab pages.



► To create a tab control

1. On the form with which you are working, choose Tab Control from the Insert Controls toolbar.

   Page Properties appears. It indicates which page of information you are on.

2. On Page Properties, complete the Event Rules Title.

   Your form appears with a tab at the top, named as you indicated.

3. Position and resize the control.

4. Choose Tab Page from the Insert Controls toolbar to add additional tab pages as necessary.

   The size of each page in the tab control is equal to the size of the entire tab control. You cannot resize an individual page to be bigger or smaller than the others.

   All tab pages appear as children of the tab control in the Application Tree View.

5. Add controls to the tab pages as if they were individual forms.

Quick Form

Quick Form allows you to place multiple database fields on a form faster than choosing each data item individually. Choose one or more data items, and Quick Form automatically places the fields on the new form simultaneously.

Depending on the number of selected fields, you might need to resize the form or move and align fields to achieve the desired layout.

Before You Begin

- ❑ Attach a business view to the form.

See Also

- ❑ *To attach a business view to a form* in the *Form Design Aid Guide*

▶ To use Quick Form

1. On Form Design, choose Quick Form from the Form menu.
2. On Quick Form layout, choose the number of columns of controls you want on the form and whether you want the columns arranged horizontally or vertically.
3. Choose the data items from the business view that you want to display and click the right-facing arrow to move them to the Columns in form pane.
4. Use the buttons under the Columns in form pane to order the data items the way you want.
5. Click OK to place the fields on the form.

   Quick Form remains open so that you can adjust the arrangement by changing the columns per row and the vertical and horizontal placement.

6. Click Close when you are finished.

Form Controls

You use form controls to provide specific functions within an application. For example, you can do the following:

- Insert field controls on forms to display data, enter data, calculate data, store data permanently or temporarily, or pass data between fields and forms
- Place check boxes on forms to provide for multiple selections, or radio buttons to indicate mutually exclusive selections

A maximum of 250 controls can exist on a form. You can use Form Properties to review the number of controls on a form. You receive a warning if you are near the 250-control limit.

Each form includes specific default controls, depending on the type of form that you are creating. However, you might need to add additional controls when you design the form. Choose from standard Windows graphical controls as well as J.D. Edwards custom controls. Available controls include the following:

- Push Button

  Use a push button to initiate an action or a set of actions.

- Check Box

  Use one or more check boxes to provide the user with options that are not mutually exclusive.

- Radio Button

  Use radio buttons to provide the user with sets of options. The radio buttons in each group are mutually exclusive.

- Edit

  Use edit fields to display data and to allow users to enter information for a specific instance of a data item.

- Static Text

  Use static text as labels on your form.

- UDC Edit

  Use this specialized edit field to limit the content of the field to a specific user defined code (UDC) table.

- Grid

  Use grids to display data and to allow users to enter information. Unlike an edit control, grid controls can show multiple data items and multiple table rows at once.

- Parent/Child

  Use parent/child controls to attach a tree view to a grid control.

- Media Object

  Use media object controls to allow users to attach text or files to a form.

- Group Box

  Use group boxes to create a box around a grouped set of controls to give users a visual cue that the controls constitute a group.

- Image

  Use image controls to place a static or animated graphic on a form.

- Tree Control

  Use tree controls to display the files and hierarchical structure of a directory.

- Tab Control

  Use tab controls to create a form with one or more pages, each with its own tab.

- Page Control

  Use page controls to add pages to a form that has tabs.

- Combo Box

  Use combo box controls to allow the user to choose from a predefined list of objects.

- Text Block

  Use text block controls to create different text segments and then attach attributes to them.

You can disable controls at design time and then enable them at runtime using event rules. If a control is disabled, it appears grayed out. If a control is read-only, you can set focus on it or use the Tab key to move to it, but you cannot enter values.

See Also

- ❑ *Data Dictionary Triggers* in the *Form Design Aid Guide* for information about overriding the default triggers for a data item associated with a control.
- ❑ *Text Variables* in the *Form Design Aid Guide* for information about creating and using variable text strings for field labels, column names, messages, and so forth.

► To insert a control on a form

In the Form Design Aid desktop, choose a form.

1. Click the button on the Insert Controls toolbar corresponding to the type of control that you want to add, and then click the form approximately where you want to place the control.

   This is the simplest way to add a control to a form.

2. Position the control using one of the following options:

   a. Drag-and-drop the position of the control.

   b. Alternatively, access the properties of the control either through the Property Browser or by double-clicking the control, and then on the Justification tab, choose the Left option to dictate the position of the upper left corner of the control.

Note Concerning Inserting a Control Using the Insert Control Toolbar
When you insert a control using the Insert Controls toolbar, FDA inserts a control that does not have a connection to a data item. You can attach a data item to a control after you create it.

Alternatively, if you display the Data Dictionary Browser or the Business View Columns Browser, you can choose the form and then double-click a data item from one of these browsers. Then click on the form to place an Edit control that is already attached to the data item, along with a Static Text control that displays the title of the data item.

► To move a control and its static text

1. On the form with which you are working, position the mouse pointer over the shaded area between the static text and the control box.
2. When the pointer becomes the hollow square design tool, press and hold the mouse button.

   When the crosswire appears, you can move both the static text and the control box together.

3. Drag the controls to where you want them. Release the mouse button.

If the static text and the control box are too close together, the hollow square design tool does not appear in the shaded area. If this occurs, move the text and the control farther apart. You also can use the pencil tool to draw a box around just one control and move it independently.

► To disconnect the data item label associated with a field

If you insert an Edit control on a form by double-clicking a data item in the Data Dictionary or Business View Columns browsers, then the system automatically creates a static text field, displaying the data item’s title, and associates it with the control. To move or delete the static text, you must first disconnect it from the Edit control.

Click the control and choose Disconnect from the Edit menu.

To reconnect the two fields, select them and choose Reconnect from the Edit menu.

► To display the title of a data item associated with a field

When you insert an edit field on your form and associate a data item with it, FDA does not automatically label the field. If you want to display the row description for the data item automatically on the form, perform this task.

1. Click the control, and then choose Associate Description from the Edit menu.

   FDA creates a static text block and attaches it to your cursor so that you can place it where you want.

2. Click the approximate location on the form where you want the description to appear.

When you associate a description with an edit control or UDC edit control, a description of the value in the control automatically appears next to it. Associating a description is optional but is useful as a visual aid on the form.

For example, suppose that the address book number, for example, 1001 appears in an edit control or in a UDC edit control. When the user presses the Tab key to move out of the control, the address book name XYZ Company appears next to the control.

▶ To disable a control

Select the control and then set the Disabled property value to Yes.

A disabled control appears to be grayed out. You can permanently disable a control by manually changing the Disabled property value. You also can programmatically disable and enable a control by changing this property value with Event Rules (ER) based on specific conditions.

▶ To hide a control

Select the control and then set the Visible property value to No.

A hidden control is invisible to the user. You can permanently hide a control by manually changing the Visible property value. You also can programmatically hide and show a control by changing this property value with Event Rules (ER) based on specific conditions.

▶ To make a control keyboard accessible

You can configure a control so that a user can invoke it directly from the keyboard without using the mouse.

Change the title of the control by placing a & symbol in front of the letter you want to use as a keyboard shortcut.

The & indicates by the letter that follows it the keyboard shortcut. For example, a control with the title $Help can be activated by the user from the keyboard with Alt+H.

Grouping Controls

Sometimes, you want a set of controls to be taken as a whole by the user, the runtime engine, or both. The classic example is when you want the user to choose one item from a list of items, and this situation is typically resolved by creating a radio button for each option and then grouping them so that the user can choose only one at a time.

When you group controls, the individual controls in the group are no longer eligible to be a part of the tab sequence for the form. That is, the user cannot use the Tab key to access members of the group, only the first item in the group. However, if you make the first control tab accessible, the user can use the Tab key to access the group and then can use the arrow keys to move among the controls in the group.

To provide the user with a visual cue that the controls are grouped, you can place a group box around the controls. To do so, choose all of the controls in the group, and then click Group Box on the Insert Controls toolbar. FDA draws a box around the controls. Use the Title property for the group box to label the group on the form.

Tab Sequence of Controls on a Form

The tab sequence of the controls determines the order in which the cursor travels through the controls on your form.

Each control that is designated as a tab stop is numbered according to the way the cursor travels. For example, when the user opens the form, the cursor appears on the control labeled number one. When the user presses the Tab key, the cursor moves sequentially to the next control (tab stop) on the form.

Only those controls that are designated as tab stops in the control properties are affected by the tab sequence. Only the first control in a set of grouped controls can be designated as a tab stop.

The default tab sequence is the order in which you place the controls on the form.

The following two properties can override the first tab stop if they are enabled for a control:

- Default cursor on add mode
- Default cursor on update mode

► To change the tab sequence

1. On the form with which you are working, choose Tab Sequence from the Form menu.

   FDA displays each control so you see its number in the tab sequence instead of its name.

2. Click the controls in the order in which you want the cursor to travel.

   The first control that you click becomes number one, the second becomes number two, and so on. To reset the numbers to their original values, right-click anywhere in the window.

   To set the tab sequence for a group of controls (such as a set of radio buttons), the Tab Stop and Group properties must be turned on for the first control in the group. All other controls in the group should have these properties turned off to ensure that the cursor tabs to the first control in the group and skips the others.

3. To hide the tab sequence, from the Form menu, choose Tab Sequence again.

Data Dictionary Triggers

You use triggers to initiate display and edit routines that are associated with data items at application runtime. When you create a data dictionary item, you define its default triggers at that time.

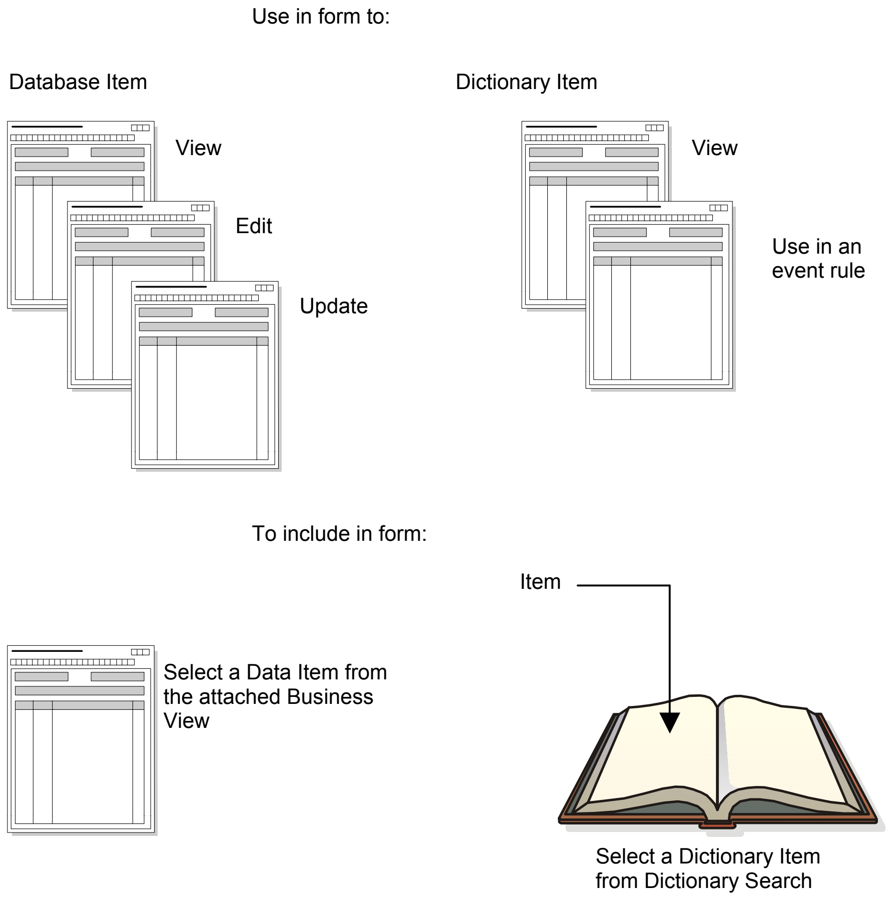
You can override previously defined triggers at the application level, which allows you to further customize how data items behave at runtime. Use the disable option to turn off data dictionary triggers for a specific item or to override triggers to change to a new value or option.

Attaching Data Items to a Control

On a form, you can use any data item that is in the business view and any data item from the data dictionary. Database items are in the business view associated with the form. Data dictionary items are not in the business view. Use a data dictionary item to store information that is not contained in a database field. Only database items are updated to the database. On a form, database items appear with a blue box in the left corner, and data dictionary items appear with a yellow box in the left corner. Use data dictionary items as display-only fields in a form or as a reference for an event rule. Data dictionary items do not retrieve or update data in a table.

The following illustration compares the use of database items and data dictionary items in an application.

Database Item and Dictionary Item



Radio buttons, check boxes, edit controls, and UDC edit controls can have associated values. A push button cannot have an associated value. Database items and data dictionary items can be associated only with check box or radio button controls. You can review the properties for a control to determine the data dictionary item with which the control is associated.

If you want the user-entered value for a check box or radio button to update the record, then the control must be associated with a database item.

For example, a check box can be associated with the database field called Taxable. In the check box properties, the Checked value should be Y and the Unchecked value should be N. You can use these values in event rules, and they will update the Taxable database field in the table.

Because data dictionary items never update the table, you can use them only in event rules. In the data dictionary item, the user can enter values that will be used in event rules, or an event rule can produce a value that appears in a data dictionary item.

► To attach a data item to a control

1. Click the control, and then click Data Item Information in the Property Browser and click the ellipses button that appears in the Data Item Information field.

   Alternatively, double-click the control and then click the Data Item tab on the Control Properties form.

2. On Control Properties, perform one of the following tasks:
   - To attach a data item from the business view attached to the form, click Business View Items and double-click a data item.
   - To attach a data item from the data dictionary, click Data Items, search for the data item you want, and then double-click it.
3. To override the data item name as it appears on your form, click the General tab, click Override Text, and then enter the name you want to use in the Event Rules Title field.

   The Text is Overridden property in the Property Browser changes to Yes. To return the display name to the data dictionary name, change Text is Overridden to No.

4. Click OK.

See Also

- ❑ *Data Dictionary Triggers* in the *Form Design Aid Guide* for information about specific data dictionary item trigger types, their functions, and how to configure them.

► To override a default data dictionary trigger

1. On your form, double-click a control with an attached data item.
2. On Edit Properties, click the Overrides tab, and then click Data Dictionary Overrides.
3. On Data Dictionary Overrides, to disable one or more triggers, choose the ones you want to disable in the Disable box.
4. To override a trigger, click the applicable trigger type and complete the form that appears.

   For example, if you click Default Value, the Override Default Value form appears and you would make you changes there.

5. Click OK.

6. When finished, on Data Dictionary Overrides, click OK.
7. On Edit Properties, click OK.

Push Button Controls

Use a push button to initiate an action or a set of actions. You can designate a single push button to be the one that is activated when the user pushes Enter by changing the Default Pushbutton property value to Yes.

Check Box Controls

You use check boxes to indicate choices. A check mark in a box indicates that you have made a choice. Check boxes are not mutually exclusive. You must associate a check box with a database or data dictionary item. You can use a check box to pass a value to an event rule.

Radio Button Controls

You use radio buttons to indicate choices. A filled radio button indicates that you have made a choice. When radio buttons appear in a group box, they should always be mutually exclusive.

You can associate radio buttons with a user defined code field, where each button has a value from the User Defined Code table. You associate a radio button with a database or data dictionary item.

Edit Controls

Edit controls are generic input fields and have no associated text. You should associate edit controls with data items.

Note

If you want to limit a field’s values to a specific user defined code (UDC) table, use a UDC Edit control instead of an Edit control.

If you associate an edit control with a data item from the form’s associated business view, then the value entered by a user at runtime updates the table. If you associate an edit control with a data item not associated with the business view, then the value entered at runtime is for display only. However, you can make a field read-only for business view data items by setting the Read Only property value to Yes.

Edit controls have a type-ahead feature. When a user enters a character in the field, the system searches a history list for a match. If a match exists, it appears in the field and is highlighted. This feature is particularly useful for data entry work because it can reduce the amount of typing required. The history list is stored in alphabetical order in a local file in the

Windows root directory. A user can enable or disable type-ahead editing in User Preferences. Type-ahead is disabled for double-byte languages and multiline edit controls.

As you add controls to the form, you can indicate how the runtime engine filters the incoming records from the database. For example, if your Find/Browse form has two controls on which you want to filter, the resulting SQL statement that you generate will be an `AND` condition for each condition. For example, if you have Search Type and Alpha Description as the controls on the form, the filter criteria for Search Description should be >= and the Search Type control should be =. If a user types D and puts a V in the Search Type field, the resulting SQL statement looks like the following:

```
SELECT * FROM F0101 WHERE (ABAT1 = "V" AND ABDC LIKE "D%") ORDER BY
ABAN8 ASC
```

You might also want to apply filters to edit controls when records need to fall between two values. In this case, you use a range filter. For example, in distribution, a status is assigned to each line of the order. One status is the current status and the other status is the next status. In this example, you filter records that are greater than or equal to the present status and less than or equal to the next status. You drop one value, filter the value, and then drop the next value.

Creating UDC Edit Controls

You use a user defined code (UDC) Edit control for a field that accepts only the specific values that appear in a UDC table. You associate a UDC Edit control with a data item.

The Search button appears adjacent to the UDC edit control field. When you click it, the attached Search and Select form displays valid values for the field.

To create a UDC edit control, you must do the following:

- Associate the data item with a specific UDC table in the data dictionary.
- Create a Search and Select form for displaying valid values from the UDC table.

► To filter database items

1. Double-click the control and choose the Filter tab on Edit Properties.
2. Choose one of the filter criterion.
3. Click the Wildcard Display option to turn it on, if desired.

   This option displays an asterisk in the field by default.
4. Click OK.

► To mask characters in a field

Sometimes, you do not want an edit control to display the input as a user enters it. For example, if you provide an edit control for users to enter their password, then you might not want to display the user's password on the screen as they type. In these cases, you can substitute an asterisk for the characters, thereby masking the characters.

Set the Password property for the edit control to Yes.

▶ To make an edit control a required field

You can require that an edit control contain a valid value before any runtime processing occurs.

Set the Required entry field property to Yes.

Static Text Controls

Use a static text control to display descriptive text, such as a title or instructions. This text is not associated with a control, and the user cannot change it. You can change it using the Set Control Text system function in event rules. You also can make static text clickable by setting the Clickable property to Yes. Use the event rule, Text Clicked, to respond to clickable text.

▶ To display the title of a data item associated with a field

When you insert an edit field on your form and associate a data item with it, FDA does not automatically label the field. If you want to display the row description for the data item automatically on the form, perform this task.

1. Click the control, and then choose Associate Description from the Edit menu.

   FDA creates a static text block and attaches it to your cursor so that you can place it where you want.

2. Click the approximate location on the form where you want the description to appear.

When you associate a description with an edit control or UDC edit control, a description of the value in the control automatically appears next to it. Associating a description is optional but is useful as a visual aid on the form.

For example, suppose that the address book number, for example, 1001 appears in an edit control or in a UDC edit control. When the user presses the Tab key to move out of the control, the address book name XYZ Company appears next to the control.

Creating UDC Edit Controls

You use a user defined code (UDC) Edit control for a field that accepts only the specific values that appear in a UDC table. You associate a UDC Edit control with a data item.

The Search button appears adjacent to the UDC edit control field. When you click it, the attached Search and Select form displays valid values for the field.

To create a UDC edit control, you must do the following:

- Associate the data item with a specific UDC table in the data dictionary.
- Create a Search and Select form for displaying valid values from the UDC table.

See Also

❑ *Edit Controls* in the *Form Design Aid Guide* for information and tasks that apply to all Edit control types in general

Grid Controls

Use grid controls to display one or more table records. You also can use grid controls to allow users to edit table records. When you make the grid control, you place data items in the grid. Each data item becomes a column.

In addition to the standard formatting options you have for all controls, such as height and location, you can do the following with property values:

- Hide or show row headers, row and column lines, and the Query by Example (QBE) line
- Allow users to resize columns and rows and to select more than one row at a time
- Control grid population and refresh in the following ways:
    - Populate the grid at the rate of one page at a time (the runtime engine only loads enough data to fill the grid, and then loads enough data to fill the grid again if the user clicks Next) or all at once
    - For grids on Find/Browse and Headerless Detail forms, populate the grid automatically upon form open and whenever a user tabs to a new line in a grid while in update mode
    - For grids on Find/Browse forms, refresh the data automatically to reflect any changes the user made on another form before being returned to the Find/Browse form
- Control how a user enters data into a grid in the following ways:
    - On Header and Headerless Detail forms, allow the user to continue adding lines to the grid after clicking OK (Clear Grid After Add)
    - Prevent users from adding rows to a grid, although the user can change existing rows (No Adds On Update Grid)
- Cause the runtime engine to process all of the rows in the grid, not just the rows that have been added or changed

► To add columns to a grid control

1. Perform one of the following tasks:

- To add a data item to the grid from the business view associated with the form, double-click a data item in the Business View Columns Browser.
- To add a data item to the grid that is not in the business view associated with the form, search for a data item in the Data Dictionary Browser, and then double-click it.

2. Repeat step one until you have added all of the columns you need.
3. Double-click the grid control.
4. On Grid Properties, click the Columns tab and arrange the column order by choosing a column and then using the Up and Down buttons to shift its position in the list.
5. Click the Sort Order tab and set the order in which the system will sort returned records by following these steps:

   a. Choose a data item in the Unsorted Columns list and click the right arrow. Repeat for each data item on which you want to sort.

   The data item moves to the Sorted Columns list.

b. To toggle between displaying by ascending and descending order, click the data item in the Sorted Columns list.

   The letter A next to the data item indicates that the system will display records in ascending order. The letter D next to the data item indicates that the system will display records in descending order.

c. Arrange the column sort order by choosing a column in the Sorted Columns list and then using the Up and Down buttons to shift its position in the list.

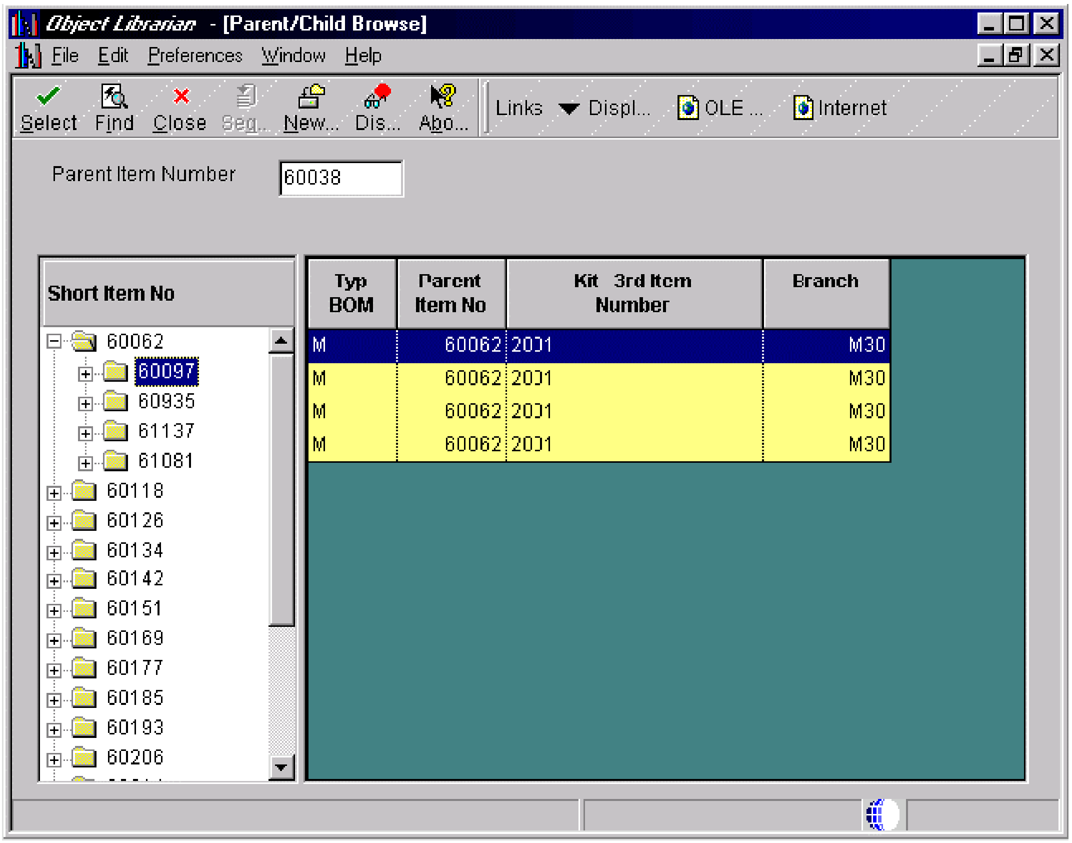
6. Click OK.

Working with Columns in a Grid Control

You can control how a column looks and behaves by changing its property values. In addition to the standard formatting options you have for all controls, such as width and visibility, you can apply most of the same kinds of controls that you can for any edit field, such as making it read only, making it clickable, making it a required entry, and so forth.

Parent/Child Controls

The Parent/Child form has a unique parent/child control that automatically appears with the form. The control is a composite control with a tree on the left and a grid on the right. The tree and grid portions use the same business view.



Parent Item Number 60038

Short Item No
60062
60097
60935
61137
61081
60118
60126
60134
60142
60151
60169
60177
60185
60193
60206

Typ BOM	Parent Item No	Kit 3rd Item Number	Branch
M	60062	2031	M30
M	60062	2031	M30
M	60062	2031	M30
M	60062	2031	M30

You can add data items to the grid. Because of its similarity to a Grid control, many of the properties for Parent/Child controls are the same. In addition to setting the properties, you can do the following:

- Show the tree expanded upon form open (Show Details of All Tree Nodes)
- After choosing which column to display in the tree control, hide or display it in the grid
- Enable or disable functions, such as drag-and-drop

After you configure the control, you set up either a parent/child relationship or use event rules to load child nodes to the tree. The method that you use depends on whether your table has an inherent parent/child relationship. In the following examples, the first example describes a situation in which an inherent parent/child relationship exists in a table. The second example describes a situation in which the data does not have an inherent parent/child relationship by keys, but can appear hierarchically to enhance readability.

You can use event rules and system functions to customize the way in which your parent/child control functions. For example, you can change the top node of a tree or change the node that appears as the first child on a tree.

See Also

- ❑ *Grid Controls* in the *Form Design Aid Guide* for more information about adding columns to the grid portion of the Parent/Child control and about setting properties for the grid and for its columns

Example: Inherent Parent/Child Relationship

This example describes a case in which an inherent parent/child relationship exists in a table. For instance, in the Bill of Material Master File table (F3002), IXKIT is the short item number for the kit. Kits are composed of multiple items within IXKIT, represented by IXITM, the short item number for the item.

Kit item 60038 includes items 60062, 60118, 60126, and other items as its children.

IXTBM	IXKIT (Parent)	IXITM (Child)	IXKITA	IXMMCU	IXAITM
M	60038	60062	220	M30	2001
M	60038	60118	220	M30	2006
M	60038	60126	220	M30	2007

Each one of the items (IXITM) can be a kit within itself.

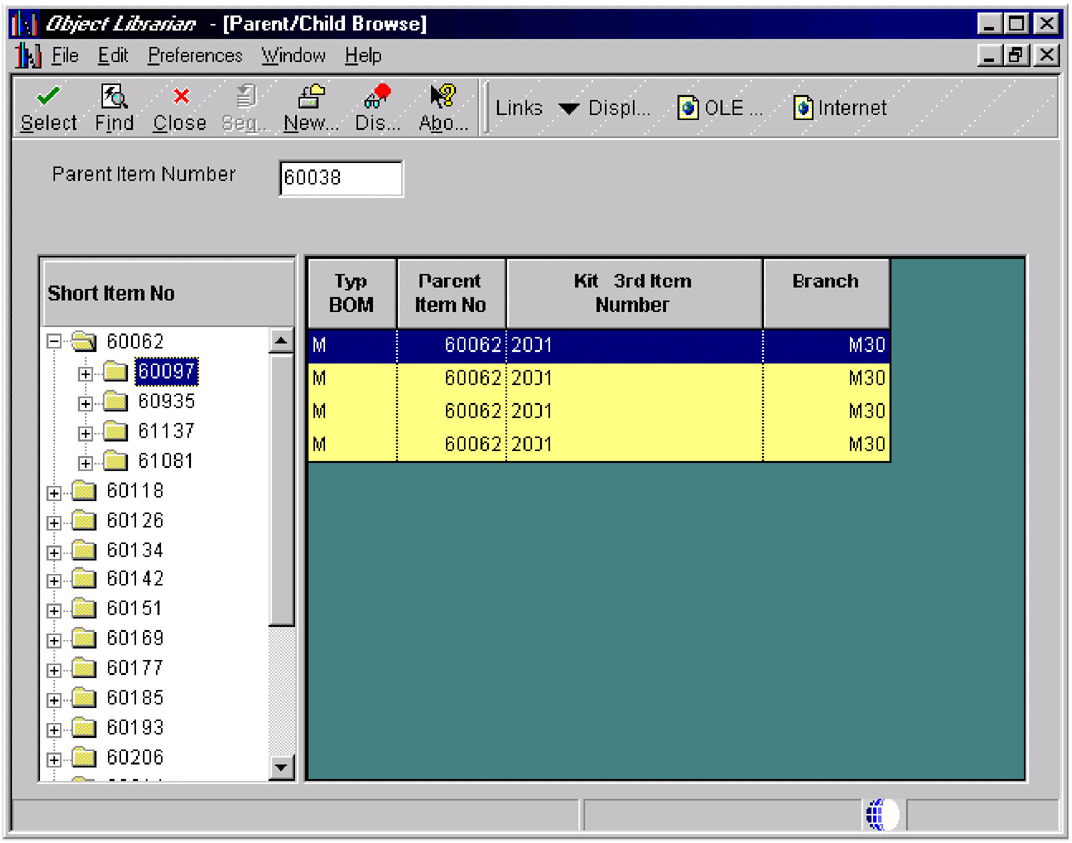
Item 60062 includes 60097, 60935, 61137, and other items as its children.

IXTBM	IXKIT (Parent)	IXITM (Child)	IXKITA	IXMMCU	IXAITM
M	60062	60097	2001	M30	2004
M	60062	60935	2001	M30	9011

M	60062	61137	2001	M30	9031

Because this inherent relationship exists within the table, you can develop a parent/child application over the Bill of Material Master File table (F3002) using the V3002A business view.

The following form represents this data.



Parent Item Number 60038

Short Item No

- 60062
  - 60097
  - 60935
  - 61137
  - 61081
- 60118
- 60126
- 60134
- 60142
- 60151
- 60169
- 60177
- 60185
- 60193
- 60206

Typ BOM	Parent Item No	Kit 3rd Item Number	Branch
M	60062	2001	M30
M	60062	2001	M30
M	60062	2001	M30
M	60062	2001	M30

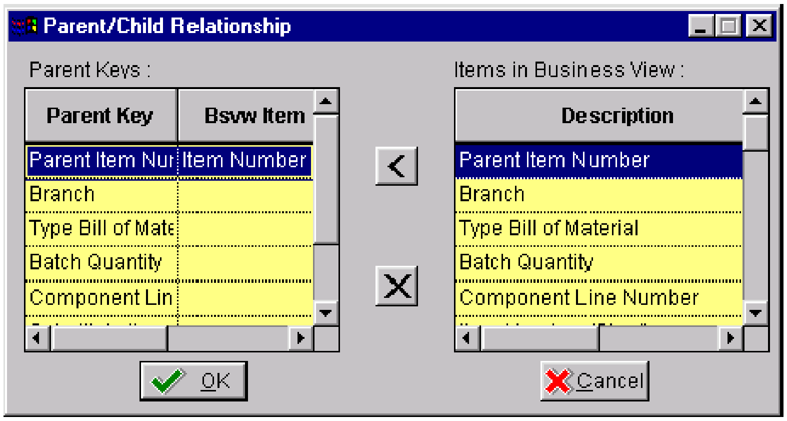
This structure represents data from the same table shown in the parent/child format. When you first inquire on item 60038 in the filter field, you generate the same SQL statement as you would on a normal Find/Browse form.

When you click the + node next to item 60062, the runtime engine takes the value of the Item Number and moves it into the Parent Item Number, making it the parent for the fetch.

Complete the following steps to set up the parent/child relationship to be used when the runtime engine loads nodes to the tree:

1. Focus on the parent/child control.
2. From the Form menu, choose Parent/Child Relation.

   The following form appears:



Parent Keys :

Parent Key	Bsvw Item
Parent Item Nur	Item Number
Branch	
Type Bill of Mate	
Batch Quantity	
Component Lin	

Items in Business View :

Description
Parent Item Number
Branch
Type Bill of Material
Batch Quantity
Component Line Number

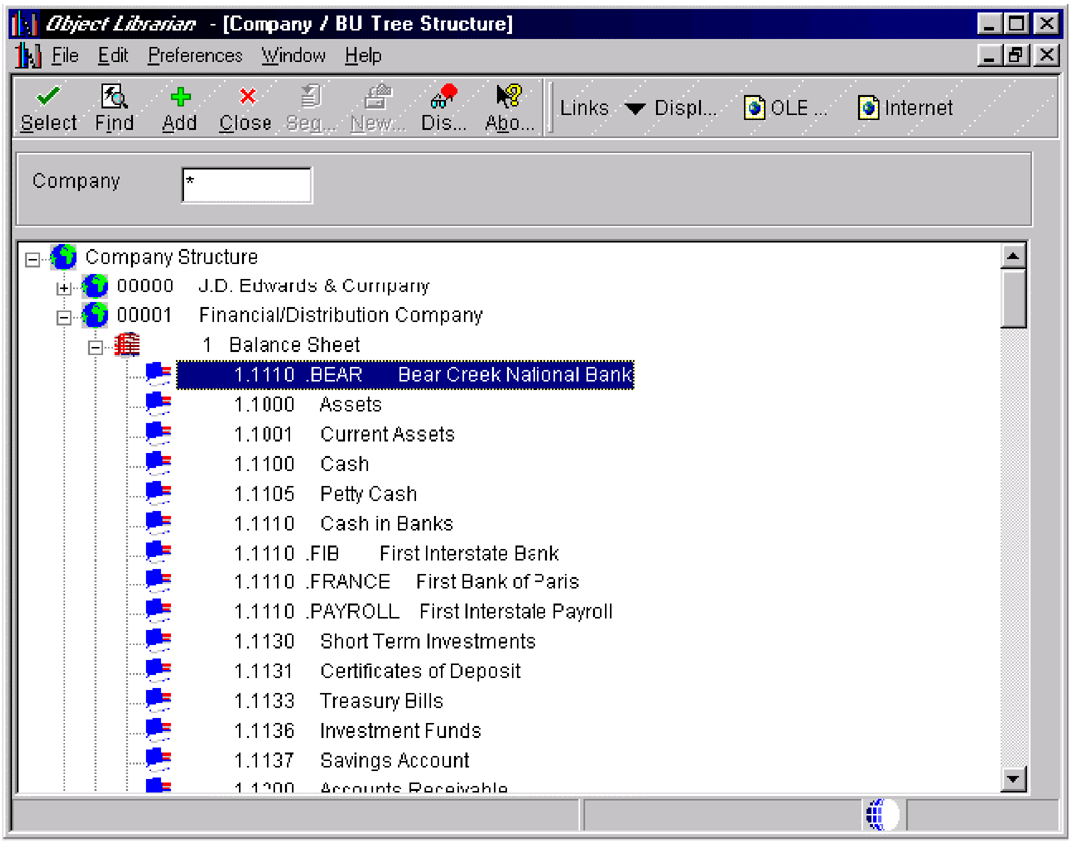
3. Indicate which child item should be used as the parent for loading the next level of nodes.

A Parent/Child form is an extension of the Find/Browse form, where the tool performs fetches using different keys to access its data. Some things to consider when you create a Parent/Child form include:

- You usually use the parent field for the filter.
- You usually use the child for the left side of the parent/child control.
- The data item type between the parent and the child must be compatible.
- You must establish the parent/child relationship correctly. You usually move the child item into the parent field.
- The tool performs subsequent (RESET) fetches only when a node is clicked. The tool takes the value of the child at that node and substitutes it into the parent field for the fetch. After the tool performs a fetch for that node, that information is saved in a runtime structure. If the same node is clicked again, the data is not fetched from the database; it is fetched from the runtime structure.

Example: No Inherent Parent/Child Relationship

This example describes a situation in which no inherent parent/child relationship exists. In this situation, the runtime engine does not automatically fetch records because the parent/child relationship is not inherent, and the relationship is between data items of different types. In this case, the data item that you chose to be in the left side of the parent/child control is usually a string (alphanumeric) variable that holds any data type.



Company *

```
⊟ Company Structure
    ⊞ 00000    J.D. Edwards & Company
    ⊟ 00001    Financial/Distribution Company
        ⊟        1  Balance Sheet
                    1.1110 .BEAR      Bear Creek National Bank
                    1.1000    Assets
                    1.1001    Current Assets
                    1.1100    Cash
                    1.1105    Petty Cash
                    1.1110    Cash in Banks
                    1.1110 .FIB       First Interstate Bank
                    1.1110 .FRANCE    First Bank of Paris
                    1.1110 .PAYROLL   First Interstate Payroll
                    1.1130    Short Term Investments
                    1.1131    Certificates of Deposit
                    1.1133    Treasury Bills
                    1.1136    Investment Funds
                    1.1137    Savings Account
                    1.1200    Accounts Receivable
```

The parent/child relationship is not inherent to a particular table because the application is accessing multiple tables, such as Company Master, Account Master, and Chart of Accounts, to get information.

Page-at-a-Time Processing in Parent/Child Controls

Page-at-a-time processing ensures that each fetch fetches only one page of data. Page-at-a-time processing is the default mode for all parent/child forms.

During page-at-a-time processing in standard mode, the page size is the number of nodes that can fit in the current view. When the Find process begins, only one page of first-level nodes is fetched from the table and inserted in the tree. When the user scrolls down, a new page of data is fetched.

Page-at-a-time processing for the expand-all style of Parent/Child forms is similar to that for standard mode. When the Find process starts, one page of first-level nodes is fetched from the table and inserted into the tree. Then, all of the first-level nodes are expanded. Each expansion fetches only one page of data. Because the tree expands exponentially in expand-all mode, a very deep tree might affect performance.

Tree Nodes

When the parent and child nodes come from different tables or are of different data types, the parent/child relationship is not automatically set up. In this case, the runtime engine does not automatically fetch the child database records because it does not know the table from which to retrieve them.

If possible, use the runtime engine to load the initial set of parent nodes to the tree for you. You do this by using the based-on view, which is a view over the table for the upper-most node. You can use a parent filter in the control, and the runtime engine loads the first level nodes to the tree. If you cannot do this, you must insert the first-level nodes yourself. To do this, you typically use Table I/O on the Button is Clicked event of the Find button. You use the same methods that you use to insert child nodes. Use a Suppress Find system function to stop the runtime engine from attempting to load any nodes.

Whenever a node is expanded, the system function Suppress Fetch on Node Expand is called from the event Tree Node Is Expanded. This function tells the runtime engine not to do any fetches because event rules will handle the loading of the child nodes. Tree Node Is Expanded is the main event of the application. This event occurs when the tree node is expanding (such as when the plus next to a child node is expanded for the first time). You place event rules on this event to read the next records to be loaded to the tree as children of the expanded node. You can use table I/O or business functions to retrieve these records. Often the children come from different tables, based on the type of parent node that is expanded. If possible, you should perform a `SELECT` and then use the `FETCH NEXT` command to retrieve records in a `DO WHILE` loop. The `GB` runtime data structure is populated with data from the records read in the loop, and then an `INSERT GRID BUFFER ROW` system function is called. This parent child system function is different from the `INSERT GRID BUFFER ROW` in the normal `GRID` section. At this point, you also can set your custom tree bitmaps using the `SET TREE NODE BITMAP` system function.

Example: Using the Tree Node is Expanding Event

In the following example, event rules are attached to an application on the *Tree Node is Expanding* event:

```
Suppress Fetch on Node Expand(FC Parent/Child)

//

// Here are the variables to get out of the account and the business unit loop

// being initialized.

VA frm_OutOfLoop = ”0”

VA frm_ExistAcctLoop = ”0”

//

// If Loop looking at the GC Business Unit Field.

If GC BusinessUnit is equal to <Blank>

//

// Select the F0006 differently if there is one company or all companies

//

F0006.Open

If VA frm_AllCompanies is equal to ”1”

VA frm_CurCompany = GC Co

F0006.Select

Else

VA frm_CurCompany = BC Company

F0006.Select

End If
```

```
//
// While Loop which fetches all the all the business unit for a specific company.
//  If the company changes we get out of the loop.
While VA frm_OutOfLoop is equal to  <Zero>
// Fetch the records from the F0006 Table.
F0006.FetchNext
GB BusinessUnit = GB Companies
If SV File_IO_Status                  is equal to CO SUCCESS
GB Co = BC Company
VA frm_PreCompany = BC Company
VA frm_ConcateBuDesc = ”   ”
VA frm_ConcateBuDesc = concat([VA frm_ConcateBuDesc],[VA frm_NameOfBU])
GB CompanyStructure = concat([GB Companies],[VA frm_ConcateBuDesc])
GB Companies = concat([GB Companies],[VA frm_ConcateBuDesc])
//
// Tells the Level of the Tree Structure.
GB LevelOfTreeInt = ”1”
//
Insert Grid Buffer Row(FC Parent/Child, <After Last Row>, <Yes>, <No>, <No>, <No>,
<No>, <Yes>)
Set Tree Node Bitmap(FC Parent/Child, <Last Grid Row>, BussUnit.bmp, <Yes>)
//
//
If VA frm_PreCompany is not equal to VA frm_CurCompany
VA frm_OutOfLoop = ”1”
End If
Else
VA frm_OutOfLoop = ”1”
End If
End While
F0006.Close
Else
// Loop thru the F0901 to get the corresponding accounts.
//
VA frm_CURBU = GC BusinessUnit
F0901.Open
F0901.Select
```

```
If SV File_IO_Status                  is equal to CO ERROR
//

End If

//

// While Loop to pick up the accounts till the Fetch Fails.

While VA frm_ExistAcctLoop is equal to  <Zero>

F0901.FetchNext

GB Sub = VA frm_DBSUB

GB ObjAcct = VA frm_DBOBJ

GB Name = VA frm_AcctDesc

GB BusinessUnit = VA frm_CURBU

VA frm_AcctDesc = concat([VA frm_BLANKS],[VA frm_AcctDesc])

GB AccountID = VA frm_AIDF0901

GC AccountID = VA frm_AIDF0901

If SV File_IO_Status                  is equal to CO ERROR

VA frm_ExistAcctLoop = ”1”

Else

GC Companies = ” ”

GB Companies = ” ”

Business Unit, Object, Subsidiary Merge

GB Companies = concat([VA frm_DBANI],[VA frm_AcctDesc])

GB CompanyStructure = concat([VA frm_DBANI],[VA frm_AcctDesc])

GC Companies = VA frm_AIDF0901

//

// Tells the level of the Tree Structure ’2’

GB LevelOfTreeInt = ”2”

//

Insert Grid Buffer Row(FC Parent/Child,  <After Last Row>, <Yes>, <No>, <No>, <No>,
<No>, <No>)

Set Tree Node Bitmap(FC Parent/Child, <Last Grid Row>, accounts.bmp, <Yes>)

End If

End While

End If

FC BUFrom = ” ”
```

Dragging and Dropping

When a Parent/Child form is created, both move and copy drag-and-drop operations are enabled. You can turn these options off in the properties for the form. When drag-and-drop is turned off and a user attempts to drag a node, the cursor indicates that dragging is not allowed, and none of the drag events executes. You can control operations to the database using the following drag-and-drop events:

- *Begin Drag* - If you allow drag-and-drop, then, on the Begin Drag Operation event, GCs are copied to GBs.
- *Drag Over Node* - You can attach event rule logic to this event to verify that the node on which the dragged record is about to be dropped is a valid situation. If it is not, you can use a system function to change the cursor to a No Drop cursor to indicate that dropping the record there is not allowed. If the cursor is not the No Drop cursor, when the record is dropped the event *End Drag* runs.
- *End Drag* - You can attach event rules to this event to update or insert information that has been moved or copied via the drag. Be aware of the effect of using Insert Grid Buffer Row in the End Drag event, as well as deleting the grid row dragged if the operation is a Move.
- *Drag Mode* - You can verify the Drag Mode system value at any time to determine the type of drag that a user is performing. For example, you can determine whether the drag is a move or a copy.

Tree Node is Selected Event

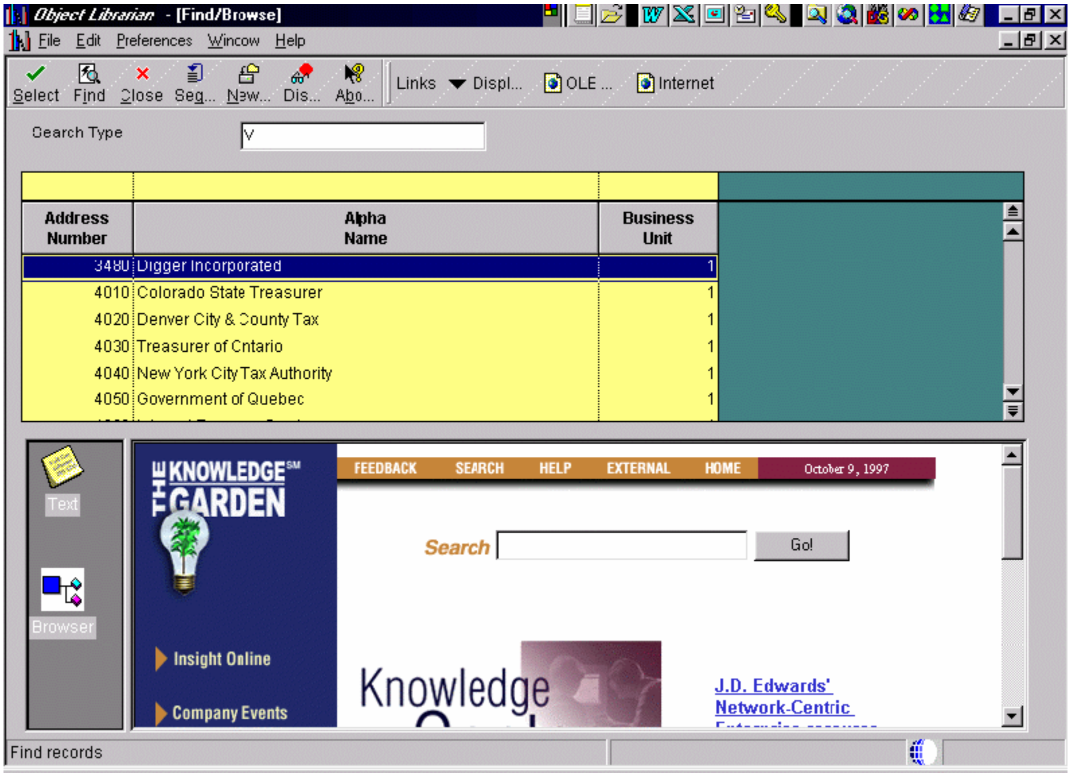
This event runs every time a user selects a tree node, either by clicking it once with the mouse or by moving an arrow up and down the nodes. You can place event rules that need to run when a mode is selected on the *Tree Node is Selected* event. The Work Center application uses this feature to load the media object that appears next to the tree with the message information for each node. You can also use this event when you need to protect controls or exits, based on the kind of node that the user selects.

Media Object Controls

The media object control is a specialized control. You can use this control to allow the user to enter text or attach objects. You can place this control on any form type except the message box.

You can use the media object control in a variety of ways. You can add images, shortcuts to other applications, and text. You can add multiple media objects to a form. You can add multiple text objects to a single media object. You also can add generic files or URLs.

If you want to display a file that is available on the Internet, you can attach the media object control to the form and create a link to the Internet.



Search Type: V

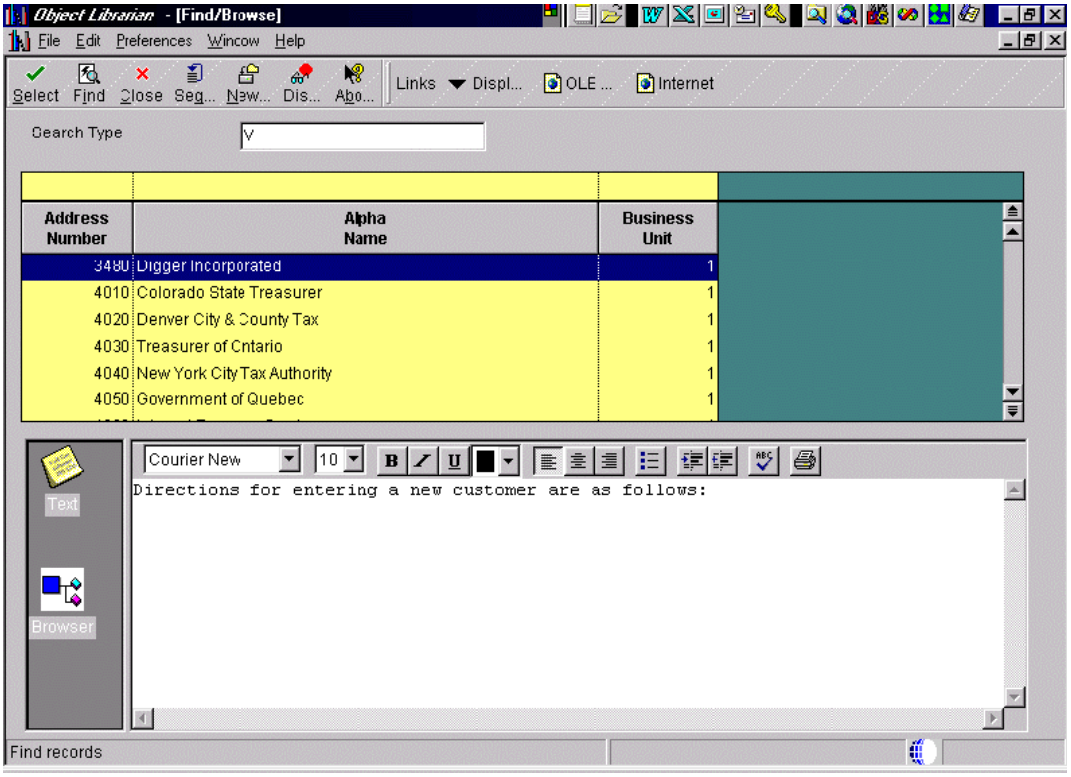
Address Number	Alpha Name	Business Unit
3480	Digger Incorporated	1
4010	Colorado State Treasurer	1
4020	Denver City & County Tax	1
4030	Treasurer of Ontario	1
4040	New York City Tax Authority	1
4050	Government of Quebec	1

Find records

After you define your media object queue for the Internet and include a valid HTTP address, you can use the Start Web Browser system function to open the control and display the Internet file.

For example, you might use this control when you need to verify the Web page for a shipping vendor so that you can track the status of shipments. You can look up the shipment directly within the media object control.

You also can use the Text feature of the media object control. For example, you can use the media object control to display instructions that are specific to a particular form.


Object Librarian - [Find/Browse]
File Edit Preferences Window Help
Select Find Close Seg... New... Dis... Abo...
Links Displ... OLE ... Internet
Courier New 10
Text
Browser

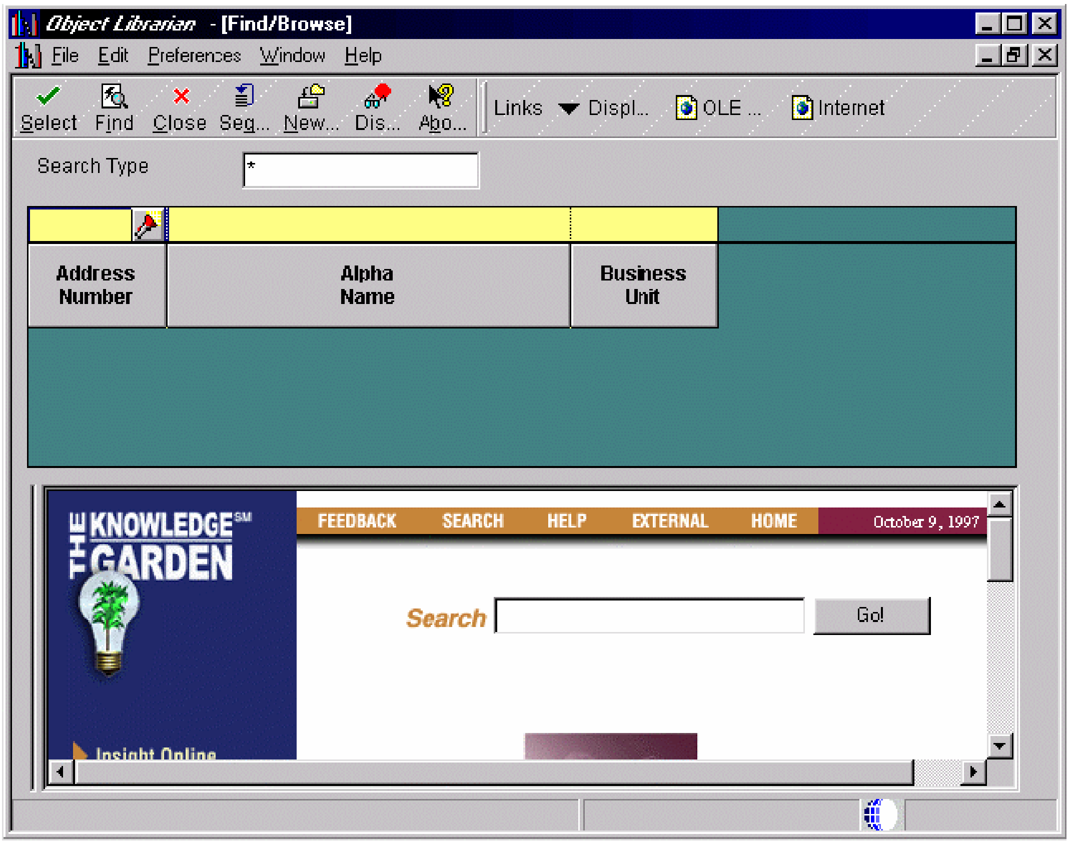
Search Type V

Address Number	Alpha Name	Business Unit
3480	Digger Incorporated	1
4010	Colorado State Treasurer	1
4020	Denver City & County Tax	1
4030	Treasurer of Ontario	1
4040	New York City Tax Authority	1
4050	Government of Quebec	1

```
Directions for entering a new customer are as follows:
```

Find records

You can use the Hide Splitter Bar system function if you do not want your employees to have access to other media object functions.


Object Librarian - [Find/Browse]
File Edit Preferences Window Help
Select Find Close Seg... New... Dis... Abo...
Links Displ... OLE ... Internet
THE KNOWLEDGE GARDEN SM
FEEDBACK
SEARCH
HELP
EXTERNAL
HOME
October 9, 1997
Search
Go!
Insight Online

Search Type *

Address Number	Alpha Name	Business Unit

You also can use the media object control to display the employee queues to which messages are sent. A media object control appears next to the tree control on a Parent/Child form. Next, event rules establish a relationship between the tree side and the media object side. For example, if a message is highlighted in the tree, then the corresponding message text appears in the control.

Image Controls

You can use a bitmap control to create a control that looks like a picture or other artwork. You can then attach event rule logic to the control. For example, you can attach logic to the Button Clicked event on the control so that when a user clicks the control, the application automatically links to a different form.

You also can use the bitmap control for an animated gif instead of a bitmap. An animated gif is particularly useful for Java and HTML applications. The animated gif is animated in Java and TML applications; in Windows applications, however, it does not appear animated and only the first image of the animated gif file appears.

Creating a Tree Control

You can use the tree control on any form type. Because the tree control is not limited to a particular business view and does not have any default database function, you have more flexibility when you load tree node data. Tree control system functions are also available to further customize tree control functions.

You can use tree control system functions to add logic to your control. You can use these system functions for actions such as contracting and expanding nodes or setting bitmaps for nodes.

Combo Boxes

You can use a combo (list) box to display a list of items from which the user can choose. The combo box includes the type-ahead feature. You can associate the combo box with a data dictionary item so that it preloads with UDC values. When you create a combo box, use a data dictionary item that has the value that you want and use meaningful UDCs. You also can use event rule logic in your application to load the values.

Creating Text Blocks

You can use a text block control to create different text segments and then attach attributes to them. You can format the text segments differently so that each segment looks different. For example, you can create a clickable text segment and add event rules to the *Click* event so that you can click the text to connect to a different form. You can also use several system functions with this control. The text block control is particularly useful for Web applications.

Text Variables

Text variables (TVs) are stored as strings and can be used as an alternative to hard-coding text strings in assignments. Because text variables are not hard-coded, they are easier to maintain. Following are some of the ways in which you can use text variables:

- To reuse forms

  You can reuse message forms and display the appropriate message depending on specific conditions.

- To reuse grid columns instead of hiding and showing them by changing the column heading text

To use a text variable, you must first create the variable. Then you can apply it to specific fields and columns with event rules.

► To create a text variable

1. In Form Design, on the form for which you want to add a text variable, choose Text Variables from the Form menu.

   The system displays the current text variables that are associated with the selected form.

2. On Text Variables in the Text String column, choose the last empty row and enter the text string that you want to use.
3. Use the Tab key to move to the next row to create another text variable.

   If you try to create a text string that is named the same as another variable on the same form (or on the same section in Report Design), an error occurs. Rename the variable to clear the error.

4. Click Save to save your variables.

Note

You can modify and delete text variables. To modify a text variable, type over it and save the new text. If you delete a text variable that is referenced in event rules, you must also delete its reference in event rules.

► To attach a text variable

1. To attach text to a control, on the form with which you are working, choose event rules from the Edit Menu.
2. On Event Rules Design, choose the event from the events list to which you want to attach the text variable.
3. Click the System Function button.
4. Click the Function Selection tab and expand the Control folder.
5. Double-click the Set Control Text system function.

The system displays the Parameter Mapping tab with information appropriate to the Control Text system function.

6. In the Parameters page, choose Control.
7. In the Available Objects page, choose the control to which you want to attach the text variable and click the right arrow between the two pages.

   The control you chose appears in the Parameters pane in the Value column, next to Control.

8. In the Parameters pane, choose Text.
9. In the Available Objects pane, scroll to the Text Variables section of the list and choose the specific text variable that you want to attach.
10. Click the right arrow between the two panes.

    The text variable you chose appears in the Parameters pane in the Value column, next to Text.

11. Click OK.
12. Complete the event rule, as required, and click OK.

Processing Media Objects

Media objects are used to link information or attachments to application transactions. For example, you can attach drawings of products to a form. You use a media object data structure to pass information between your application and the media object table.

You can use standard processing for media objects to bypass all event rules that are required to implement media objects. All of the required information for a form is gathered in Form Design and does not require the entry of any event rules. Standard processing does the following:

- Allows you to implement media objects at the form level with no event rule coding required
- Standardizes how you use media objects across forms
- For any grid, places a paper clip icon on the row header if a media object is defined for that row
- For a form, places an icon in the status bar if a media object is defined for the form
- Allows you to attach documents to the form or to a row in the grid
- Allows you to double-click the paper clip icon in a row to start media objects for that row
- Allows you to click the paper clip icon in the status bar to start media objects for the form
- If you choose not to use standard processing for a form, allows you to use event rules and system functions to make media objects work

The Media Objects Storage table (F00165) stores link records for media objects and imaging. You must define your media object data structure using a unique key structure so that the F00165 table can store data correctly. The layout of the F00165 table is as follows:

`GT`*`xxx`* `|| F4211Keys ||` *The media object text*

`GT`*xxx* is the naming convention that you use when you define a media object data structure. `F4211Keys` is information the system uses to access the unique media object attachment for that record. These keys typically match the unique key for each detail line in the F4211 table. The media object text is the actual text attachment that stores information that the user typed. The F00165 table chains records for text that is greater than 30 kilobytes. A sequence field indicates the order of the text and that the text is stored in two files.

The Media Object Categories table (F00166) contains a record with characterization values for each media object that has been characterized. Each record contains the characterization values for a single media object, such as a text or Word document. Multiple F00166 records can exist for each F00165 record. The key is the same as that of the F00165 plus the Object ID.

The Media Object Category Constant table (F00167) contains a record that indicates which categories are used to characterize setup for each GT structure. Only one record exists for each GT structure.

The Media Object Queues table (F98MOQUE) is used for multiple OLE queues. This table allows you to include media objects in several different queues. It includes the following:

- Online/offline queue paths

- Secondary text queues
- Queue status information, such as read-only and read/write information
- Longer queue names that allow FTP access by the Java Application Server

Before You Begin

❑ To display the media object paper clip column on your form, turn off the Hide Row Numbers option in the Grid properties for the form in Form Design.

► To use standard processing for media objects

1. On Form Design, choose Media Objects Setup from the Form menu.
2. On Media Objects Setup, click the Enable Automatic Media Object Functionality option to turn it on.

   This option enables imaging and opens the other fields on the form.

3. Click one of the following options:
   - Media Objects Only
   - Document Handling Only

     Use the Document Handling Only option if you are developing a form that is enabled for media objects via functionality in event rules and you want to bypass standard processing. If you want to enable standard processing later, you must delete all of the event rules for media objects and click the Media Objects & Document Handling option to turn it on.

   - Media Objects & Document Handling
4. Click Edit mode or Display mode.

   Edit mode allows the user to make changes; display mode is read-only.

5. Click Define Form Key.

   The System Functions form appears. This form is identical to the parameter definition form that you use to define system functions in Event Rules, except that it includes only the Media Objects header.

   The key structure stores link records in the Media Objects Storage table (F00165). These links are critical to proper functioning of imaging.

6. Expand the Media Object Structures folder.
7. Choose the appropriate structure and define it on the Parameter Mapping tab.
8. Click OK.
9. On Media Objects Setup, click OK.

► To add a language-specific media object attachment

1. On the application that you want to use, type a language in the filter field.
2. Click Add.
3. Add multiple records if you want the attachment for multiple languages or base.

If you create a custom application that you want to enable for media object language handling, you must include a data item language preference (alias LNGP) in the generic text data structure that you create.

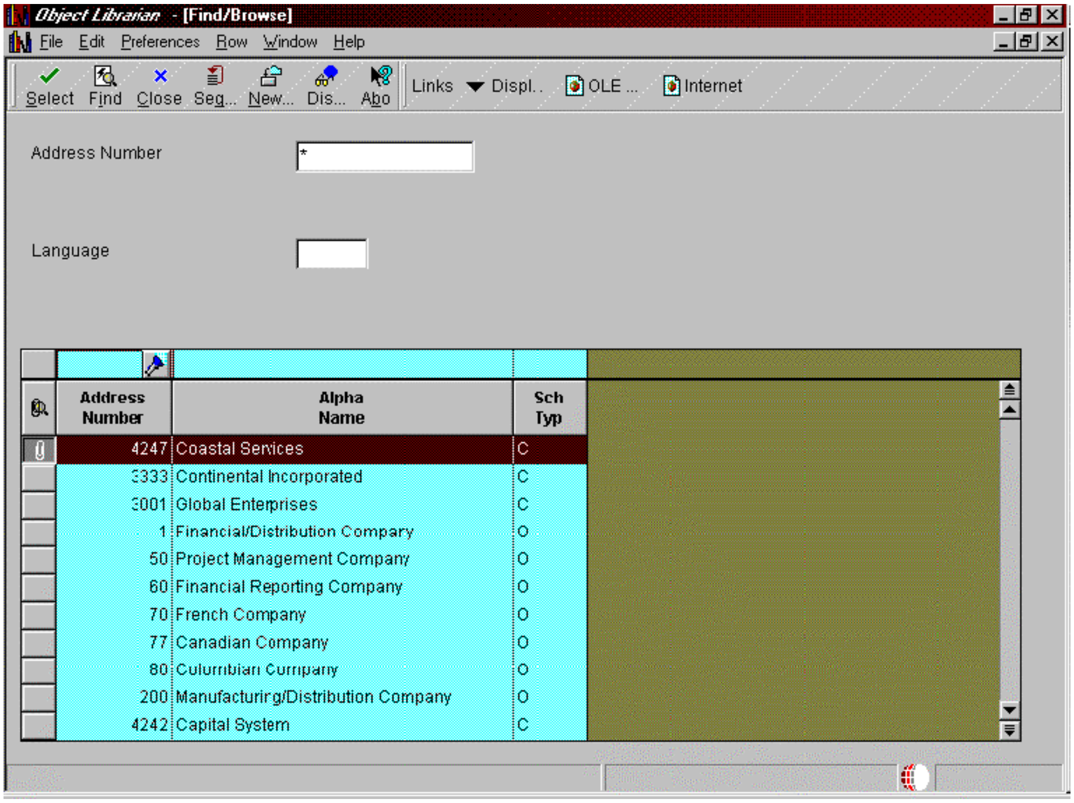
Example: Language Considerations for Media Objects

When you design an application, you can allow the end user to add separate and unique media objects to the same record or different records, based on the language chosen.

If language (LNGP) is not a database column, then you define your media object (GT) data structure to include language as part of the data structure. You place a data dictionary control (LNGP) on the application as a filter field, which should then be loaded with the system value for language. When you design your application this way, you attach two separate media objects, based on the language, to the same record.

If language (LNGP) is a database column, then you include LNGP (database) as a filter field, but you must add a separate record to the database table along with its media object attachment. The media object data structure still contains language as part of the key to retrieve the media object attachment. In both cases, the language filter fields (LNGP) must be loaded with the system value for language. LNGP must be built into the key and not associated with the LNGP column in the Media Objects Storage table (F00165).

For example, the following Address Book application includes Address Number and Language as filter fields. The language control is not from the database; it is a dictionary control.

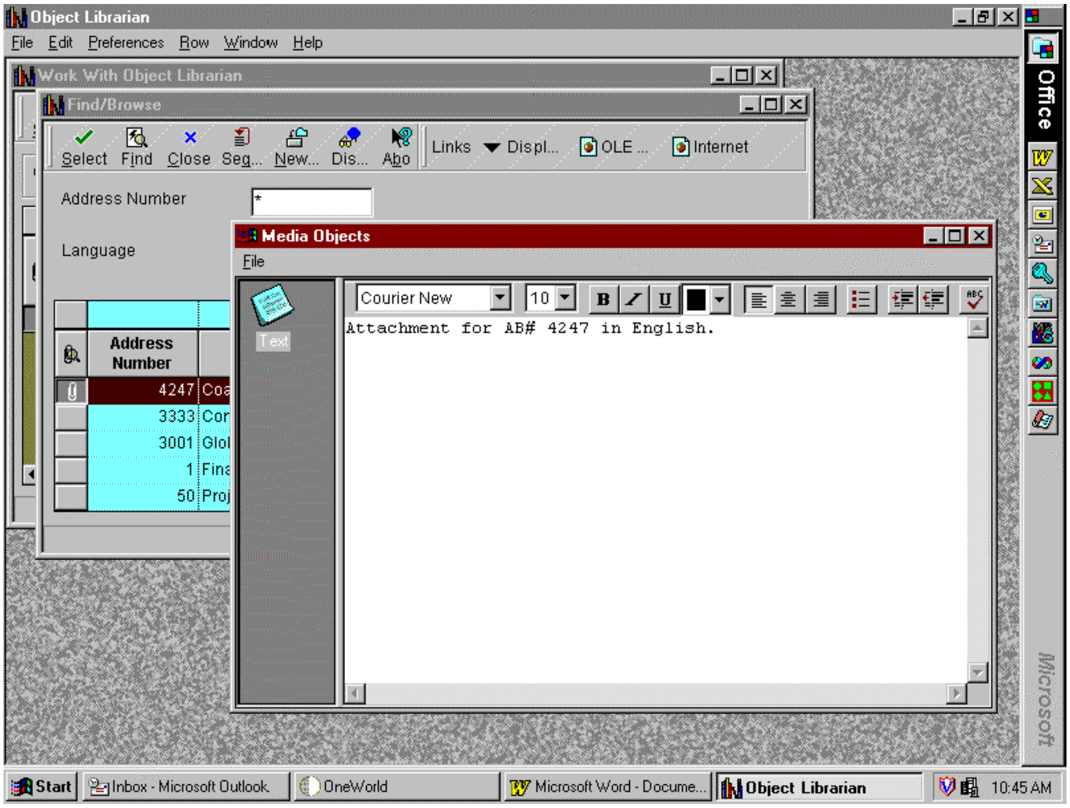

Object Librarian - [Find/Browse]
File Edit Preferences Row Window Help
Select Find Close Seq... New... Dis... Abo
Links Displ.. OLE ... Internet

Address Number *

Language

Address Number	Alpha Name	Sch Typ
4247	Coastal Services	C
3333	Continental Incorporated	C
3001	Global Enterprises	C
1	Financial/Distribution Company	O
50	Project Management Company	O
60	Financial Reporting Company	O
70	French Company	O
77	Canadian Company	O
80	Columbian Company	O
200	Manufacturing/Distribution Company	O
4242	Capital System	C

For a blank language (English), Address Book record number 4247 has a media object attachment in English.


Object Librarian
File Edit Preferences Row Window Help
Work With Object Librarian
Find/Browse
Select Find Close Seq... New... Dis... Abo
Links Displ... OLE ... Internet
Media Objects
File
Text
Courier New 10
Office
Microsoft
Start
Inbox - Microsoft Outlook
OneWorld
Microsoft Word - Docume...
Object Librarian
10:45 AM

Address Number *

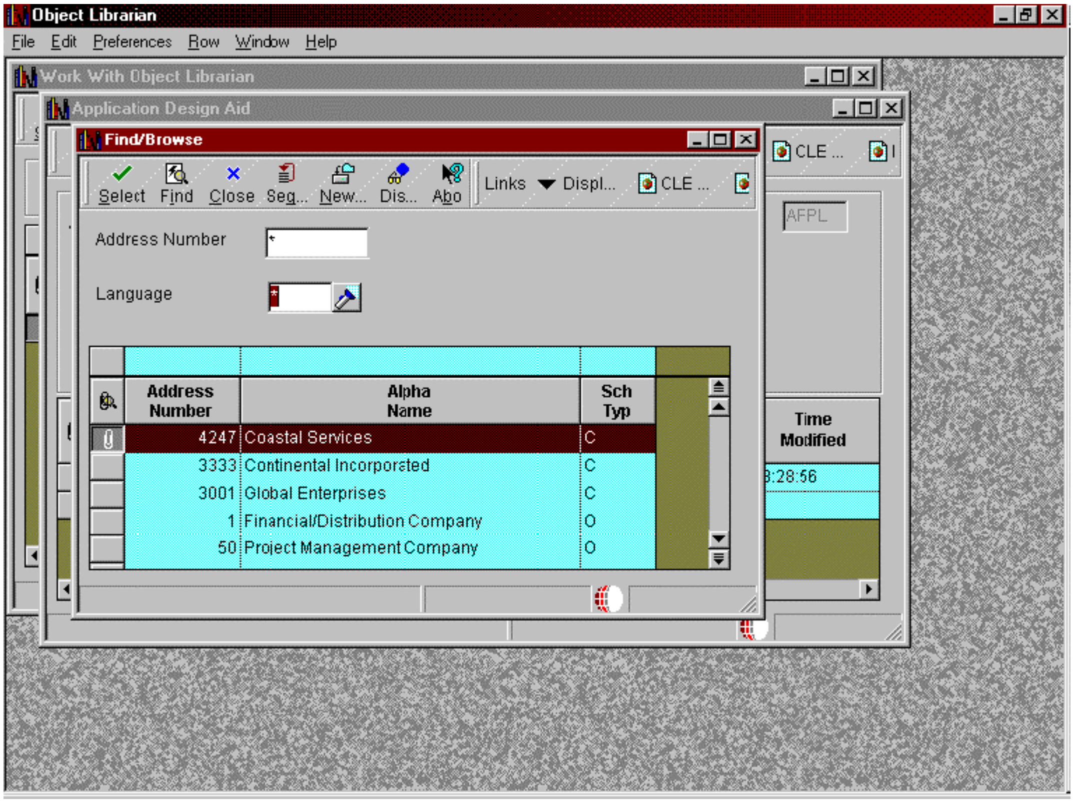
Language

Address Number	
4247	Coa
3333	Cor
3001	Glol
1	Fina
50	Proj

```
Attachment for AB# 4247 in English.
```

When a user whose language preference is set to French (F) accesses the same record, the filter field does not load with F, and the user sees a different media object attachment for the record. The Address Book does not have two separate records based on the language, it has only one record in English. For the French media object attachment, a separate record exists in the Media Objects Storage table (F00165) for the same Address Book record. It can be accessed using the language filter control field.

In this example, the ABLNGP field and column from the Address Book table is the filter field on the form. The Language field on the form is from the database, not from a data dictionary item.

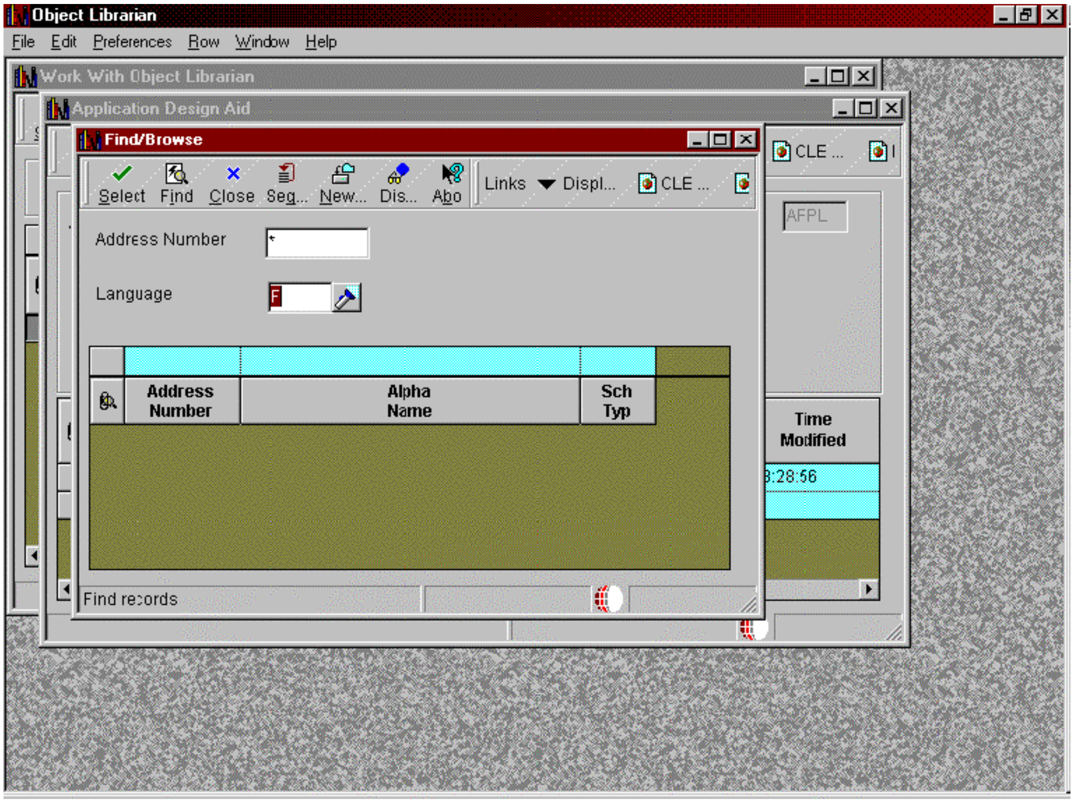

Object Librarian
File Edit Preferences Row Window Help
Work With Object Librarian
Application Design Aid
Find/Browse
Select Find Close Seg... New... Dis... Abo
Links Displ... CLE ...
CLE ...
AFPL
Time Modified
3:28:56

Address Number *

Language *

Address Number	Alpha Name	Sch Typ
4247	Coastal Services	C
3333	Continental Incorporated	C
3001	Global Enterprises	C
1	Financial/Distribution Company	O
50	Project Management Company	O

When you search for records using blank or * (English or ALL) as the base language, you see all records in the Address Book with their corresponding media object attachments. You use the same media object data structure as you did with the previous case, which contains the Address Number and Language data items. You still load the user's language preference in the language filter field, but, when the application fetches records from the database, it uses the language as part of the key to select records from the Address Book. In this example, if no records exist in the Address Book with the language preference for French (F), you do not see any records; therefore, you must then add records using French as the language.



Address Number *

Language F

Because the Address Book is keyed using only address number, not the language, you cannot add a record for address book number 4247 in French. Instead, if you want to see only the French records, you can add new address book records with new address book numbers that have the language preference of French.

For any database table that contains language as part of its key, you can attach media object functionality for records with different languages. For example, you can create one record for English and a copy of the record for French with unique media object attachments. For tables that do not include language as part of the key to that table, you can have media object languages. To see an example, go to the Item Master Print Message program (P40162) from menu GH4112.

Imaging

Imaging allows you to use images from third-party software. For performance reasons, imaging occurs on a form-by-form basis. When you attach an image to a document, the media object form includes the vendor's software as a choice.

When you use a third-party vendor, the Media Object Storage table (F00165) stores the reference to image attachments, but the third-party software controls the search and retrieval of images.

Menu/Toolbar Exits

Menu/toolbar exits are items that can appear only in the menu or that appear both in the menu and on the toolbar. An item must be placed in the menu to appear on the toolbar. The toolbar is a visual shortcut to menu items. You can display a bitmap next to an item in a form or row menu, or on the toolbar. All form types except Message have at least one menu: File. Message forms do not have menus. Furthermore, the File menu has one exit, which cannot be deleted: Close.

J.D. Edwards provides several standard base menus (all of which are empty except for File) and several standard row exits. You can create custom menus and exits for a form, or you can use any of the standard exits that J.D. Edwards provides. You cannot place custom row exits on the File menu. You can create cascading menus—that is, menus that contain other menus, which, in turn, provide other menus or exits. J.D. Edwards recommends that you limit submenus to three levels.

You can attach event rules to an exit. For example, you can attach a form interconnection so that, when you choose a menu option, another form is called.

You can see the menu/toolbar exits currently assigned to a form, along with their hierarchical structure, in the Application Tree View pane. Choose Application Tree View from the View menu to see the pane, and then expand the Menus node to see the menu/toolbar exit items.

To create a menu/toolbar exit, complete the following tasks:

1. Create the base menu item and populate it with exits or other menus.
2. Select a bitmap for the exit if you are going to show it on the toolbar.
3. Attach Event Rule logic, if necessary.

► To create a menu/toolbar

1. In Form Design, on the form with which you are working, choose Menu/Toolbar Exits from the Form menu.

   All form types (except for Message) have at least one menu, File, and it cannot be deleted. Depending on the form type with which you are working, any of the following standard exits might appear under the File class:

   OK Accepts the data in the form, clears fields, and remains in the form. In update mode, OK closes the form. In add mode, OK might or might not close the form.

   Cancel Closes the form and returns to the previous form. Any additions, revisions, or deletions the user made in this form are ignored.

2. On Menu Exits, if you do not want one of the listed exits to appear on your form, choose the menu exit and click Delete.

   Alternatively, if you want an exit to be unavailable by default but want to include logic that might enable the exit, set the Disabled property to Yes. You also can use property settings to hide an exit if hiding the object is more appropriate than disabling it.

3. To insert a new root menu on Menu Exits, perform the following steps:
    a. Click the Exit/Menu Properties tab.
    b. Click File.
    c. Click Insert.
    d. On Menu Exit Properties, choose an option from the Class field and click OK.

    Note
    The system places the Add, Copy, Delete, Find, and Select exits under File.

4. On Menu Exits, to insert an exit or to nest another menu in a menu, perform the following steps:
    a. Click the Exit/Menu Properties tab.
    b. Click the menu to which you want to add an item.
    c. Click Insert.
    d. On Menu Exit Properties, choose an option from the Class field and click OK.

    On Menu Exits, you can drag and drop objects to change the hierarchical structure of the exits, although the system will not allow you to create an illegal parent/child relationship. For example, the system will not allow you to drop a user defined exit on the File menu.

    Form, Row, and View are standard exits. All exits that are subordinate to these categories are user defined. The exits that you create appear on a drop-down menu on the toolbar. If you select a Row category, all exits that are subordinate to it need the Grid option turned on.

5. On Menu Exits, to attach a bitmap to an exit (thereby putting it on the toolbar for the form), perform the following steps:
    a. Click the Bitmap Strip Reference Properties tab.

        If you do not see this tab, then you cannot assign a bitmap to the item.

    b. Enter the name of the bitmap you want to use in the File Name field.

        Bitmap strips are stored in the `\package name\res` directory, such as the `\B9\PY9\res` directory. The name of each bitmap strip begins with `HC_`.

6. On Menu Exits, to attach event rules to an exit, click Event Rules to launch the Event Rules Design interface.
7. When finished, click OK on Menu Exits.

Attaching Event Rules to a Menu/Toolbar Exit

You can attach event rules to an exit. For example, you can attach a form interconnection so that, when you choose a menu option, another form is called. You cannot attach event rules to Begin Categories.

► To attach event rules to a menu/toolbar exit

1. In Form Design, on the form with which you are working, choose Menu/Toolbar Exits from the Form menu.
2. On the Menu/Toolbar Exits form, choose the exit that you want to use and click the Event Rules button.
3. On Event Rules Design, choose the Button Clicked event.
4. Add the event rule logic that you want to attach.
5. Save your work and exit from Event Rules Design.

Menu/Toolbar Exit Properties

The property settings for a form control its appearance and how it behaves. When you first create a menu/toolbar exit, the system prompts you to configure its properties. You can change the properties later in the design if necessary.

Menu/toolbar exits have the following property values:

- Class

  The type of item. All of the J.D. Edwards standard menus and toolbar exits are listed in this field, as well as User Defined, which is the type that you must choose to create your own menu or exit.

- Short Text

  Text that the user sees for the item as it is displayed in the application. The ampersand (&) followed by the letter indicates that it is the keyboard shortcut. For example, the user can activate $File from the keyboard with Alt+F.

- Long Text

  Text that the user sees for the item when they hover over it with their mouse.

- Display On

  An exit that can be displayed in the following ways:

  - Toolbar

    The exit is available from the toolbar under the parent menu, which is represented on the toolbar by its associated bitmap.

  - Grid

    The exit is available in the grid control, assuming one is included on the form.

  - Tree

    The exit is available in the tree control, assuming one is included on the form.

  - Parent/Child

    The exit is available in the parent/child control. Only parent/child forms can contain a parent/child control.

- State

  The initial state (enabled or disabled) for the object. If you set an object's initial state to disabled, then you should provide logic that enables it under certain conditions. If you do not need the object, then you should delete it. You also can hide an object.

- Style

  A designation that allows an exit to function as a separator (a line surrounded by extra space) on a menu or a toolbar.

Note

These properties are the ones that appear on the Menu Exit Properties form. You might see other property values for a menu/form exit in the Properties Browser. Click the + symbol at the bottom of the Property Browser to see a description for each property type, or right-click and choose What's This from the pop-up menu.

Form Interconnectivity

You can call a form from a form. This kind of form interconnect falls into two categories: modal and modeless.

- Model interconnects allow the user to view only one form in the connection at a time. Additionally, the data connection between the parent and child is not dynamic.
- Modeless interconnects allow the user to view multiple forms in the connection. The data changes on a parent or child are immediately reflected in all other open forms in the connection, regardless of whether the user has committed a data change.

Modal is the default type for interconnects when you create a new form with Form Design Aid. Modal interconnects are appropriate when you want to lead a user through a particular process in which a number of values must be input in a specific order, such as with a Director. In this case, you want the user to completely fill out each form before moving on to the next one. Add and Copy functions also lend themselves to modal processing because you want the user to complete the function before going on to others.

Modeless interconnects are valuable when a child form repeats a number of values displayed by the parent. Child forms with the same business view as the parent or which have a number of columns in common with the parent are good candidates for modeless interconnects. Additionally, if you are creating forms for use on the Web, applications with modeless interconnects tend to scale better than those without.

You can create a modeless form interconnection between a Find/Browse form and one or more Fix/Inspect forms or transaction forms (Header and Headerless Detail forms). When you update the transaction form, you do not need to reinquire on the database before the update appears on the Find/Browse form. After the initial Find/Browse, you can access any form type, including another Find/Browse.

Each time the system calls a modeless form interconnection, it passes the values in the form data structure. When you click Cancel in the calling form, the system destroys the called form and does not pass any values. When you click OK in the called form, the system passes values back to the calling form through the form data structure, but does not close the form. When you click Find, the system breaks all parent/child connections and does not reestablish them then you click Select.

Modeless interconnects affect entire form controls only. If you connect Header or Headerless Detail forms modelessly, changes inside the grid control are not communicated between the parent and its siblings.

The *Dialog is Initialized* event occurs only the first time that the system calls it. If you want event rules to run each time a form appears, attach them to *Post Dialog is Initialized*.

If a form in update mode fails to fetch a record from the database, the form mode changes to Add mode. If the Close Form On Add option is turned on, the form closes when you click OK.

When you close a Find/Browse form, it closes all of the modeless interconnect records before it destroys itself.

Event rules that follow a modeless form interconnection execute immediately instead of waiting for the called form to return.

Before You Begin

- ❑ Define the data structure of the receiving form to include any field for which you are passing values.

► To create a modal form interconnection

1. On Event Rules Design, choose an event.
2. Click the Interconnect button.
3. On Work with Applications, choose the application to which you are connecting.

   Work with Forms displays available forms for the chosen application.
4. Choose the appropriate form to which you want to connect (the target).
5. Choose the appropriate version of the form to which you want to connect.

   The Data Item column displays data items in the data structure of the target form. The keys in the primary unique index for the primary table of the business view are automatically set up as the data structure.
6. In the Available Objects column, choose the object that you want to pass.
7. Use the > button to move it to the Data Structure-Value Column.

   Indicate the direction of data flow between Value and Data Items.

   If you do not want data to pass between forms, set all Direction values to null, and then click OK to save the Form Interconnection and exit.

   As you click the direction arrow, it toggles through the following five options:

   - Data flows from the source to the target
   - Data flows from the target to the source
   - Data flows from the source to the target and, upon exiting the target, data flows back to the source.
   - Upon exiting the target, data flows back to the source
   - No data flows either way
8. Click the Include in Transaction option to include this interconnection for transaction processing.

   This option appears only if you are calling from a fix/inspect, header detail, or headerless detail form.
9. Click one of the following buttons to add notes:
   - Structure Notes
   - Parameter Notes
10. After the data structure is defined, click OK.

    Event Rules Design displays the Form Interconnection with the following statement:

    Call (Application <*name*> Form <*name*>)

J.D. Edwards Design Standards

The following standards apply to all form types:

- Accept the default placement of primary unique key fields at the top of the data structure.
- Change the data item name and description to describe the item that is passed between forms.

► To create a modeless form interconnection

1. On Event Rules Design for the find/browse form that you want to use (the source), choose an event.
2. Click the Form Interconnection button.
3. On Work with Applications, choose the application to which you are connecting.

   Work with Forms displays available forms for the chosen application.
4. Choose the appropriate fix/inspect form to which you want to connect (the target).

   The Form Interconnect - Values to Pass window displays the data structure for the target form.
5. Choose the appropriate versions of the fix/inspect form to which you want to connect.

   The Data Item column displays data items in the data structure of the target form. The keys in the primary unique index, for the primary table of the business view, are automatically set up as the data structure.
6. Click the Modeless option.
7. In the Available Objects column, choose objects that you want to pass. Use the > button to move objects to the Data Structure-Value Column.
8. Indicate the direction of data flow between Value and Data Items.

   If you do not want data to pass between forms, set all values to ↵and click OK to save the Form Interconnection and exit.

   As you click the direction arrow, it toggles through the following four options:

   - Data flows from the source to the target
   - Data flows from the target to the source
   - Data flows from the source to the target and, upon exiting the target, data flows back to the source
   - No data flow
9. Click one of the following buttons to add notes:
   - Structure Notes
   - Parameter Notes
10. After you define the data structure, click OK.

    The Event Rules Design displays the Form Interconnection with the following statement:

    Call (Application <*name*> Form <*name*>).

Designing Forms Using Multiple Modes

You can use control modes to develop an application with multiple interfaces, which reduces the need to maintain several different versions of the same application. You can create one base application and use modes to modify the application for different interfaces. You can enable or hide controls on forms for each mode. Only visibility and enable/disable properties for controls, columns, and menu exits are different for different modes. If you show hidden fields, they appear only for the current mode. All other properties are the same and are common for all modes. All fields are enabled and appear all forms.

The Windows runtime engine does not recognize control modes; only the HTML runtime engine recognizes them. Mode 1 is the default mode. You attach an application to a menu to run. This menu allows you to run an application in different modes. When you run an application over the Web, the application runs in mode 1 by default and another mode if you specify one. If you attach an application to a Windows menu, the Windows runtime engine ignores any modes that you specified and runs the application in mode 1. Use modes consistently throughout your applications. To create Web-enabled versions of your forms, you generate them in Java and HTML using eGenerator. The generator allows you to generate forms simultaneously for one or more modes.

► To view forms in a particular mode

1. On Form Design, from the View menu, choose Control Modes.
2. From the mode drop-down menu that appears, choose one of the following mode options:
   - Mode 1
   - Mode 2
   - Mode 3

   The forms and controls that are particular to the mode that you chose appear.

You usually develop your applications in Mode 1 and customize them as needed for Mode 2 and Mode 3.

Testing a Form

You should test a form to verify the placement of controls on the form. When you test a form, it appears exactly as it will appear in the application. You can preview forms in foreign languages; to do this, you must have the base language installed on your machine.

Test mode is for viewing only. You cannot modify a form in test mode.

► To test a form

1. On Form Design, choose Test from the Form menu.

   The form appears as it will appear in the working application.

2. To return to design mode, click Close or Cancel in the test form.

► To preview a form in another language

1. On the form you want to preview, choose Language preview from the Form menu.
2. Choose the language in which to display the form and click OK.
3. To continue designing the form, close the language preview form.

Using FDA Compare

The FDA Compare tool in Form Design allows you to compare one version of an application to another. You can compare them on the application level to determine whether forms have been added, deleted, or rearranged and whether the properties have changed. You also can compare the forms in the application to each other to see whether controls have been added, deleted, or rearranged and whether the properties have changed.

While working with the target object, you can use all Form Design functions except creating new forms. While comparing, you can change the target object to match the source object. If an object exists in the source but not in the target, you can copy it to the target. If an object exists in both but is different in some way, you can merge the specifications from the source to the target to make them identical.

Important

You cannot change the source in any way.

As a software developer, you might use FDA Compare as the final step before checking in your changes to ensure that you made all of the changes you intended to make. In this way, you can also make sure that you did not move a control or make a property change unintentionally.

As an administrator, you might use FDA Compare to see the changes between a software update and your pristine or current implementation. If you have performed a number of customized modifications, you can more carefully implement the software changes without fear of ruining your customization.

Before You Begin

- ❑ Ensure that you are familiar with *Understanding the FDA Compare Color Scheme* in the *Form Design Aid Guide*.

► To merge specifications using FDA compare

1. In Form Design, open the target application.

   This is the application that you have been working on or that your company uses. It is the version of the application that you might want to change when you compare it to the source, which is a base version of the application.

2. From the File menu, choose Compare Mode.
3. On Select Source Object, complete the following fields:
   - Source Object Name

     Enter the name of the application that you want to use as the basis for your comparison. While in Compare Mode, you can make changes to your target application, but you can't change the source application.

   - Source Object Location

     The system loads the source object and modifies the Form Design display to show both applications simultaneously. The source object is identified by name and the target object is identified by Local. The browsers and the form desktop are split; the source appears on the right and the target appears on the left.

In compare mode, you can work with the target as if you were in the regular Form Design mode, with one exception: you cannot add new forms to the application.

4. Click OK.

   The system loads the specifications for the source object from the source location.

5. To change the view in the source or target pane, choose one of the following from the drop-down menu at the bottom of the pane:

   - All

     Displays all the properties and values in alphabetic order for the object including the translation properties.

   - Standard

     Displays only the standard properties in alphabetic order.

   - Translation

     Displays only the translation properties in alphabetic order.

6. Compare the Local or target application specifications (on the left) to the target specifications (on the right).
7. To merge specifications from the source object to the target object, right-click in the source pane and the choose Merge to Target.

   The value in the source pane will now appear in the target pane.

8. When you have completed all compares and merges, from the File menu, choose Compare Mode to disable FDA Compare and return to regular form design.

► To exit FDA Compare

In Form Design with FDA Compare mode enabled, choose Compare Mode from the File menu to disable FDA Compare mode and return to regular Form Design mode.

Understanding the FDA Compare Color Scheme

FDA Compare uses colors and letters to highlight the differences between the source and target objects. The default colors and their meanings are as follows:

Black	The object exists in both the source and target and is the same in both versions.
Red	The object exists in both the source and target but is not the same in both versions.
Green	The object exists only in the target.
Blue	The object exists only in the source.

In the form workspace, the controls that differ between the versions are marked with a symbol in a color indicating the type of change. When you click a control, the browsers highlight the changes in specific property values with the same color coding.

Note

Nodes in the Application Tree View browser appear in a color if one or more of their children are different. Therefore, while the node itself may not be different, you will find that an object is different when you drill down into the tree.

► To change the FDA Compare color scheme

1. In Form Design, choose User Options from the View menu.
2. On User Options, click the Compare tab and set the colors.

   To return the color scheme to the default values, click Reset Colors.

3. When finished, click OK.

► To copy or merge an object from source to target

1. Launch FDA Compare and load the source and target applications.
2. On the target side of one of the browsers, right-click the object that you want to copy or merge.

   An object can be a form, a control, a property, and so forth.

3. Choose Copy to Target or Merge to Target.

   The choice varies depending on the nature of the operation.

   The system updates the target and changes the browsers and the form. The object is now black and unlabeled because it is identical in both source and target.