PeopleSoft

# EnterpriseOne 8.93
# Form Design Aid
# PeopleBook

**May 2004**

EnterpriseOne 8.93
Form Design Aid PeopleBook
SKU TT893FDA0504

# Table of Contents

# About These EnterpriseOne PeopleBooks Preface

EnterpriseOne PeopleBooks provide you with the information that you need to implement and use PeopleSoft EnterpriseOne applications.

This preface discusses:

- EnterpriseOne application prerequisites
- Obtaining documentation updates
- Typographical elements and visual cues
- Comments and suggestions

**Note**

EnterpriseOne PeopleBooks document only fields that require additional explanation. If a field is not documented with the process or task in which it is used, then either it requires no additional explanation or it is documented with common elements for the section, chapter, PeopleBook, or product line.

## EnterpriseOne Application Prerequisites

To benefit fully from the information that is covered in these books, you should have a basic understanding of how to use EnterpriseOne applications.

See the *Foundation Guide*.

You might also want to complete at least one EnterpriseOne introductory training course.

You should be familiar with navigating the system and adding, updating, and deleting information by using EnterpriseOne menus and forms. You should also be comfortable using the World Wide Web and the Microsoft Windows or Windows NT graphical user interface.

These books do not review navigation and other basics. They present the information that you need to use the system and implement your EnterpriseOne applications most effectively.

## Obtaining Documentation Updates

You can find updates and additional documentation for this release, as well as previous releases, on the PeopleSoft Customer Connection Website. Through the Documentation section of PeopleSoft Customer Connection, you can download files to add to your PeopleBook Library. You can find a variety of useful and timely materials, including updates to the full PeopleSoft documentation that is delivered on your PeopleBooks CD-ROM.

**Note**

Before you upgrade, you must check PeopleSoft Customer Connection for updates to the upgrade instructions. PeopleSoft continually posts updates as the upgrade process is refined.

### See Also

PeopleSoft Customer Connection Website, *http://www.peoplesoft.com/corp/en/login.jsp*

# Typographical Conventions and Visual Cues

This section discusses:

- Typographical conventions
- Visual cues

## Typographical Conventions

The following table contains the typographical conventions that are used in EnterpriseOne PeopleBooks:

| Typographical Convention or Visual Cue | Description |
|---|---|
| *Italics* | Indicates emphasis, topic titles, and titles of PeopleSoft or other book-length publications. Also used in code to indicate variable values. |
| Key+Key | A plus sign (+) between keys means that you must hold down the first key while you press the second key. For example, Alt+W means hold down the Alt key while you press W. |
| `Monospace font` | Indicates a PeopleCode program or other code example. |
| “ ” (quotation marks) | Indicates an adjective that is used in a way that might not be readily understood without the quotation marks, for example "as of" date, “as if” currency, "from" date, and "thru" date. |
| Cross-references | EnterpriseOne PeopleBooks provide cross-references either below the heading “See Also” or preceded by the word See. Cross-references lead to other documentation that is pertinent to the immediately preceding documentation. |

## Visual Cues

EnterpriseOne PeopleBooks contain the following visual cues:

- Notes
- Cautions

### Notes

Notes indicate information that you should pay particular attention to as you work with the PeopleSoft system.

---

**Note**

Example of a note.

---

### Cautions

Text that is preceded by *Caution* is crucial and includes information that concerns what you must do for the system to function properly.

---

**Caution**

Example of a caution.

---

# Comments and Suggestions

Your comments are important to us. We encourage you to tell us what you like, or what you would like to see changed about PeopleBooks and other PeopleSoft reference and training materials. Please send your suggestions to:

PeopleSoft Product Documentation Manager, PeopleSoft Inc., 4460 Hacienda Drive, Pleasanton CA 94588

Or you can send e-mail comments to *doc@peoplesoft.com*.

While we cannot guarantee an answer to every e-mail message, we will pay careful attention to your comments and suggestions.

# Using the PeopleBooks Online Library

This section provides an overview of the PeopleSoft Online Library and discusses:

- Navigating Through the PeopleSoft Online Library
- Searching Through the PeopleBooks Online Library

The PeopleSoft Online Library (PSOL) is an HTML-based tool that contains comprehensive documentation for PeopleSoft EnterpriseOne application and tools. Use this documentation as an online research library. The PeopleSoft Online Library is organized hierarchically, like a library of books. It provides standard navigation and search capabilities, including an expandable table of contents, a keyword index, and a full-text search feature.

# Navigating Through the PeopleSoft Online Library

This section discusses:

- Accessing the PeopleBooks interface
- Enabling the Reference Pane
- Navigating Between Books and Chapters
- Navigating Within a Chapter
- Using the Table of Contents
- Using the Index

## Accessing the PeopleBooks Interface

When you open the PeopleSoft Online Library home page, the PeopleBooks Online Library appears in your browser. When you click the PeopleBooks link, the PeopleBook interface appears in your browser with a list of available PeopleBooks.

After you choose a PeopleBook, the PeopleBook interface displays information in the following three panes:

- Document pane

  The pane on the right side of the window; displays the document HTML file, which corresponds to a chapter in the PeopleBook.

- Navigation pane

  The pane above the Document pane; contains display options, navigation controls, and the current PeopleBook and chapter titles.

- Reference pane

  The pane on the left side of the window; contains three tabs: Contents, Index, and Search. Use these tabs to explore the PeopleBooks or to locate a specific topic. You can hide the Reference pane to maximize the size of the Document pane.

## Enabling the Reference Pane

If the Reference pane (with the Contents, Index, and Search tabs) does not appear, click the Show Reference Pane button in the Navigation pane at the top of the browser. To hide the Reference pane, click the Hide Reference Pane button.

## Navigating Between Books and Chapters

Use the Navigation pane to view information about the current PeopleBook, and to navigate between PeopleBooks or between chapters in the current PeopleBook.

Click the links at the bottom of the Navigation pane to access the PeopleSoft Online Library home page (Home), the PeopleBooks Library home page, and the first page of the current PeopleBook.

Use the following buttons in the Navigation pane to navigate through each PeopleBook:

| Button | Description |
|---|---|
| Previous (left directional arrow) | Click Previous to go to the previous chapter in the book. |
| Next (right directional arrow) | Click Next to proceed to the next chapter in the book. |
| First | Click First to go to the first chapter in a book. |
| Last | Click Last to go to the last chapter in a book. |

**Note**

These buttons move you to the previous or next file in the sequence in which the chapter files are organized in the book, not (as with a browser's Forward and Back buttons) in the sequence in which you opened the files.

## Navigating Within a Chapter

Use the navigation features in the Document pane to navigate within a chapter. Click a cross-reference link to go to a related topic. These links appear in See Also headings.

Use the following buttons in the Document pane to navigate within a chapter:

| Button | Description |
|---|---|
| Top (up arrow) | Click Top to go to the top of the current chapter. |
| Previous (double arrows) | Click Previous to go to the beginning of the parent section. |

## Using the Table of Contents

Click the Contents tab to display the table of contents for the current PeopleBook. Use the following functions in the Contents tab to navigate through the current PeopleBook:

| Feature | Description |
|---|---|
| Closed folder (with plus symbol) | Click a closed folder icon to expand a chapter. |
| Open folder | Click an open folder icon to collapse a chapter. |
| Section (document symbol) | Click a section icon to open to the section. |
| Synchronize | Click the Synchronize button to open the Table of Contents to the topic that is currently displayed in the Document pane. |
| Expand All | Click Expand All to open all of the folders in the Table of Contents. |
| Collapse All | Click Collapse All to close all of the folders in the Table of Contents. |
| Keep TOC synchronized with document | Click this option to automatically synchronize the Table of Contents as you navigate through the chapters. |

## Using the Index

Click the Index tab to search through a keyword index of the current PeopleBook. To display an index topic, enter a keyword in the text box, or scroll to the keyword and click it. The document appears at the associated topic unless multiple topics exist.

# Searching Through the PeopleBooks Online Library

This section discusses:

- Performing a Simple Search
- Performing an Advanced Search

## Performing a Simple Search

To perform a simple search, enter the text for which you want to search, and then press Enter or click the Search button. The Search list box displays all of the topics that contain the text that you entered, along with the PeopleBook in which each topic belongs.

**Note**

When you use this tab on the PeopleSoft Online Library home page, the system only searches through the first group of guides, or the guides associated with the first link. To search through a different group of guides, click the appropriate link to display the list of guides, and then perform a simple search. The system searches through all the guides in the list.

The simple search form uses an *accrue* logic when searching. That is, it finds results that contain any or all of the terms which you entered, with priority given to documents that contain all or most of the keywords. The results appear sorted by book title and then by score.

When you search on multiple words, the system displays topics that contain any of the words in the search criteria. However, if you want the search to return topics that contain all of the words in the search criteria, surround the words with quotation marks, for example “Accounts Receivable Features.”

## Performing an Advanced Search

Use the Advanced Search options to expand your search. You can define the type of search to perform and refine your search results. Click the Advanced Search link on the Search tab to access the following Advanced Search options:

- Full Text

  Choose this option to perform a full text search. The Search list box displays all of the chapters with text that matches your search criteria, along with the PeopleBook wherein the chapters belong.

- Chapter Title

  Choose this option to search for text within chapter titles only. The Search list box displays all of the chapter titles that contain the text that you entered, along with the PeopleBook in which the chapters belong.

- Search Within Results

  Choose this option in combination with the Full Text or Chapter Title option to refine your search results.

# 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 PeopleSoft 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 *EntepriseOne Web Client Guide* for more information about developing Web applications

# 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 PeopleSoft 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.

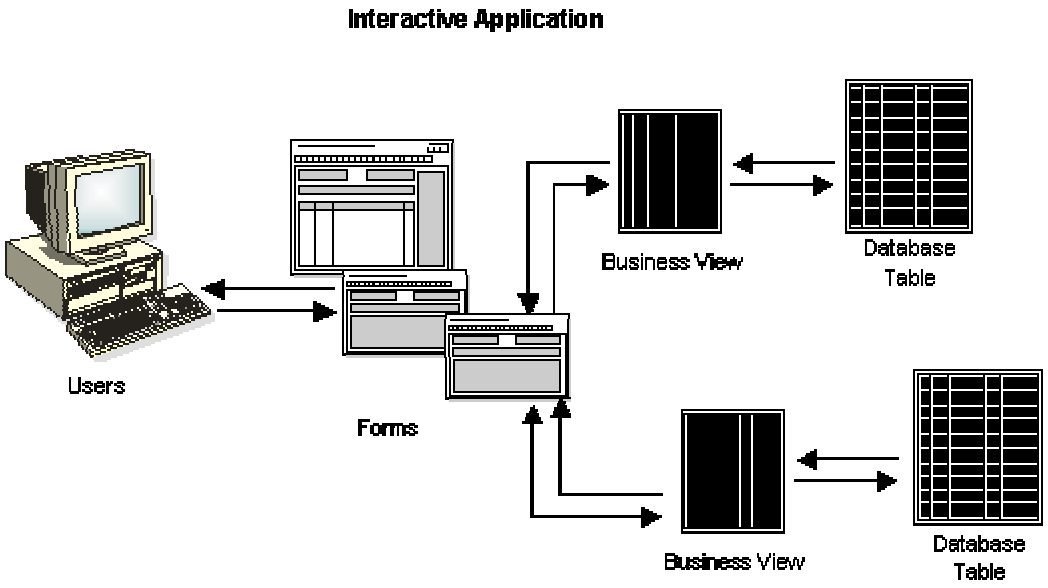
# Understanding Applications

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).

PeopleSoft 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.



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.

### ► To create an interactive application object

*In the PeopleSoft EnterpriseOne Menus task view, choose the PeopleSoft EnterpriseOne Tools node. Expand the PeopleSoft EnterpriseOne Tools node and then the Object Management node.*

1. Launch Object Management Workbench.
2. On Object Management Workbench, click Add.
3. On Add PeopleSoft EnterpriseOne 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 PeopleSoft EnterpriseOne Menus task view, choose the PeopleSoft EnterpriseOne Tools node. Expand the PeopleSoft EnterpriseOne Tools node and then the Object Management node.*

1. Launch Object Management Workbench.
2. On Object Management Workbench, click Add.
3. On Add PeopleSoft EnterpriseOne 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.

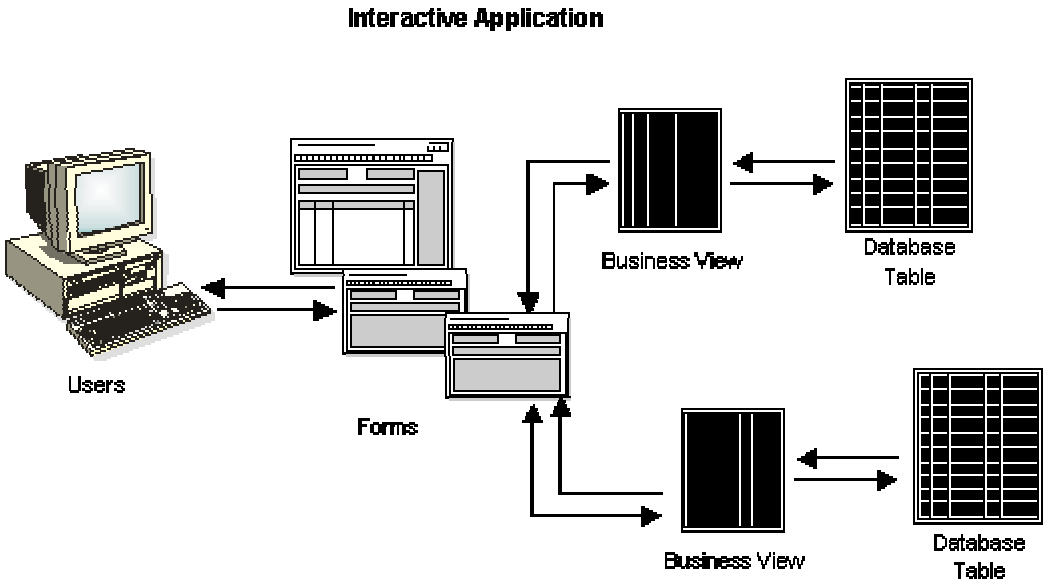
    - Version Title

      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

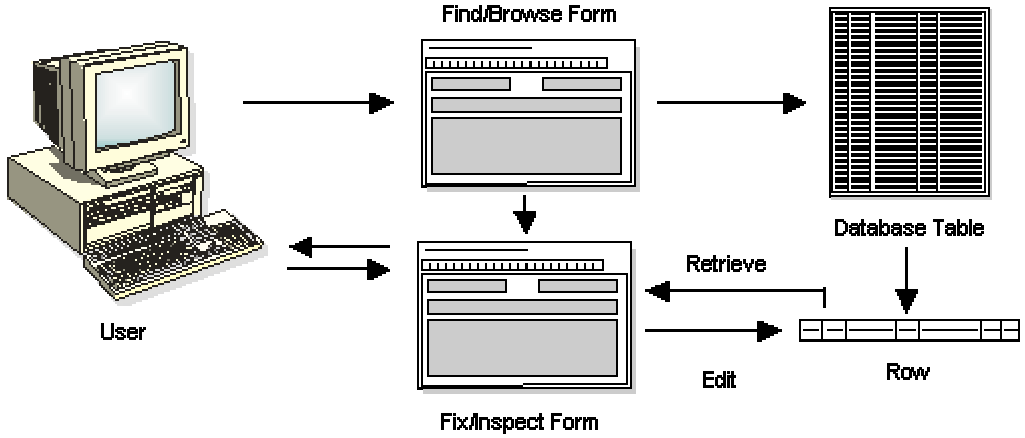
# Form Design

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.




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.

### Prerequisite

- Acquire a working knowledge of Windows and have an understanding of the PeopleSoft 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 in the following table. PeopleSoft's recommendations for using the objects follow the table.

| 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. The pane can be filtered for language. You can also configure the tree to display its hierarchy by control or by business unit. | 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 |
| Tab Sequence Toolbar | Allows you to manipulated the tab sequence functionality, including changing the object being sequenced on the form. | Choose View > Toolbars > Tab Sequence Toolbar |

### 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.
- Display the Tab Sequence Toolbar when working with Power Forms and Subforms.

### ► 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 and a discussion of why you might enable or disable some values
- *Menu/Toolbar Exits* in the *Form Design Aid Guide* for instructions about how to add menus and toolbar exits to a form
- *To map the parent form's variables to a child Subform* in the *Form Design Aid Guide* how to map variables

### ▶ 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 the properties of a form later in the design process.

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, the user can activate $File 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.

  The following additional transaction settings exist if you are using Subforms:

  - Transaction Disabled

    The Subform is not part of any transaction of the parent form.

  - Include in Parent Transaction

    The Subform is included in the parent transactions.

  - Subform Only in Transaction

    The system commits all transactional changes local to the Subform only and is not part of the parent's transaction.

- Size
- Position
- Update on form/subform business view

  When enabled, the system updates the tables underlying the form (except 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/Subform business view

  When enabled, the system updates the form based on information in the tables underlying the form (except 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.

- Subform Parent

  The parent form of the Subform. This can be at the ROOT level but does not have to be.

- Show Subform Header

  An option that shows or hides the header of the Subform.

- Visible

  An option that makes the tab or Subform visible to the end user.

- Disabled

  An option that disables the Subform.

- Mapping Link

  Data that defines the variable mappings between each form and its child Subforms in the business view hierarchy. The following mapping link directions are allowed:

  - From parent to child
  - From child to parent

---

**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 of 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. Tabs allow you to use multiple controls on 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.

Additionally, you can embed or reuse Subforms on a tab page, or you can specify a Subform to act as the tab page itself.

### ► 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.

### Prerequisite

- *To create a Subform as a tab page* in the *Form Design Aid Guide* for instructions about how to insert Subforms on tab controls.

# 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.

### Prerequisite

- 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.

# Power Form

A Power Form is a Web application form that allows users to view multiple, interrelated views of data, grids, and tab pages on one form and to pass logic between them. The tab pages can have their own business views, and these business views can communicate with each other and can update based on data selection and changes that occur in other business views on the form. Power Forms are designed for advanced end users. Power Forms might be useful for the following verticals: Wine Industry Modules (WIM), Engineer to Order (ETO), Procurement, and Customer Relationship Management (CRM).

You can create the following two types of Power Forms using Form Design Aid:

- Power Edit

  Use a Power Edit form to browse a form, change the data on a form, or both. This form is similar to a Headerless Detail form.

- Power Browse

  Use a Power Browse form to browse data on a form. You cannot use a Power Browse form to change data on the form. This form is similar to a Find/Browse form.

Power Forms have the following general properties:

- All regular controls except a parent/child control can be placed on a Power Form.
- Multiple tab controls are allowed.
- The maximize grid feature is supported for all grids.
- All Power Form (and Subform) errors and warnings are shown from the error button on the Power Form.
- Power Forms contain vertical and horizontal scroll bars.

Using Power Forms to create applications has many advantages. For example: Applications are usually characterized by a sequence of forms. Typically, a user launches an entry form and then, to update information or find related data, must navigate through a series of interconnected forms, which might not be the most efficient way of performing simple and repetitive tasks. Power Forms eliminate these navigations by allowing developers to place Subforms on Power Forms.

## Subforms

Subforms allow users to see multiple data views. For example, a user selecting a purchase order in a grid could see related shipping information in one Subform and fulfillment information in another Subform on the same Power Form. When the user selects a row in the grid, all of the data is updated and, most importantly, the user doesn't have to open a new form to see the updated data.

Subforms have two *main* properties:

- To its parent, it is a control.
- To the controls that it contains, it is a form.

Subforms also have the following *general* properties:

- All regular controls can be placed on a Subform.
- Multiple tab controls are allowed unless the Subform is inside a tab page. If the Subform is inside a tab page, it cannot have tab controls.
- Toolbar and Form/Row exits are not allowed on a Subform.
- Users must add action buttons to a Subform. No default action buttons are defined by the Form Design Aid tool.
- Default actions can be placed anywhere on the Subform.

  For example, the default action Save can be placed below the grid.

- All Subform errors and warnings include the Subform name to help the user identify which form is causing errors and warnings.
- Subforms do not contain scroll bars; you must display all controls within the form.
- Subforms have their own business views (only one per Subform), grids, and filters that pass logic between other Subforms.
- Subforms can be used as tab pages, or they can be placed on a tab page.
- Subforms support the following two form options:
  - Update on Business View
  - Fetch on Business View
- Subforms support the following two event rules created especially for them:
  - Notified by Parent

    Use the Notified by Parent event rule to run any business logic that you want the application to run after the parent of the Subform calls the Notify Child system function. This logic is usually based on the information passed to this Subform from its parent.

    The Notified by Parent event rule is executed by runtime only when its parent calls the Notify Child system function.

  - Notified by Child

    Use the Notified by Child event rule to run any business logic that you want the application to run after its child calls the Notify Parent system function. This logic is usually based on the information returned from its child.

    The Notified by Child event rule is executed by runtime only when its child calls the Notify Parent system function.

    **Note**

    The Notified by Child event rule available on parent Subforms as well as Power Forms.

- Multiple tab controls are allowed unless the Subform is inside a tab page. If the Subform is inside a tab page, it cannot have tab controls. However, this restriction cannot be enforced programmatically when you reuse Subforms. Therefore, when developing reusable Subforms, carefully consider the context in which you expect to use them to prevent poor user interface design.
- Default actions can be placed anywhere on the Subform. For example, the default action, Save, can be placed below the grid. Use the dotted lines at the top and bottom of a Subform during design time as guides for where to place more common default actions. For example, buttons such as "Select" or "Find" belong at the top of the form while buttons such as "Save" or "Cancel" belong at the bottom.

## Reusing Subforms

Subforms can exist as discrete objects in the system. You can place the Subform itself on a Power Form; this action is referred to as embedding. You can also reference an existing Subform on a Power Form; this action is referred to as reusing. Typically, developers reuse Subforms that are designed to appear on numerous forms, such as the display of Address Book information. In this way, you can more easily standardize the interface.

A reused Subform is actually just a pointer; therefore, it is unaware of its parent and children in any given context. If you want a reused Subform to communicate with its parent or children, you must do so through event rules. While you cannot embed a Subform within another Subform, you can reuse a Subform within another Subform. In other words, you can embed or reuse a Subform within a reusable Subform, but you cannot embed or reuse a Subform that contains an embedded Subform.

Inserting reusable Subforms in a form is different from inserting an embedded Subform. They exist as two different control types in the user interface. Therefore, the FDA menu and toolbars contain two different insert actions. If you insert an Alias (that is, an embedded Subform), you are prompted for the application. If you insert Reusable Subform, you are prompted for the Subform. You cannot insert a Subform onto a form until the application the Subform has been defined in has been saved.

To make reusing Subforms effective, you must plan carefully and be cautious when altering Subforms. For example, if both a parent and a child are reused, then you must set up their data mappings in a way that facilitates their passing information to each other. You should use variables as often as possible, especially for BCs, GCs, and FCs. Establish a naming convention for your variables so that you can determine which ones were created for the purpose of mapping. That way, if another developer wants to reuse the Subform and sees several “extra” variables, that developer will understand that the variables are not extraneous.

Knowing that any Subform can be reused, you must be careful when changing a Subform—even just resizing it. Anyone who is currently pointing to your Subform will see the changes you make. It is possible that if you make your Subform “a little larger” that it will no longer fit on someone else’s form. Even worse, if you change the data structure by removing elements from it, you might break someone else’s application.

---

**Important**

Always determine the full affect of changing a Subform. If you reuse Subforms in your applications, check them occasionally to ensure that the Subforms have not been altered in such a way as to negatively affect your applications.

---

# Hierarchy and Power Forms

Hierarchical relationships govern the processing between containers in your application. Before the Power Form enhancement, the Form Design Aid tool used only a control hierarchy; the entire hierarchy was contained in other controls. The Power Form enhancement allows you to create hierarchies based on business views using Power Forms and Subforms. A Power Form is at the top of the hierarchy. It is a parent form, and can have multiple children, or Subforms. Subforms also can function as parent forms, and they can have multiple children. All Subforms must have at least one parent Power Form or Subform. At most, you can have three levels, including the Power Form, in a Power Form hierarchy.

---

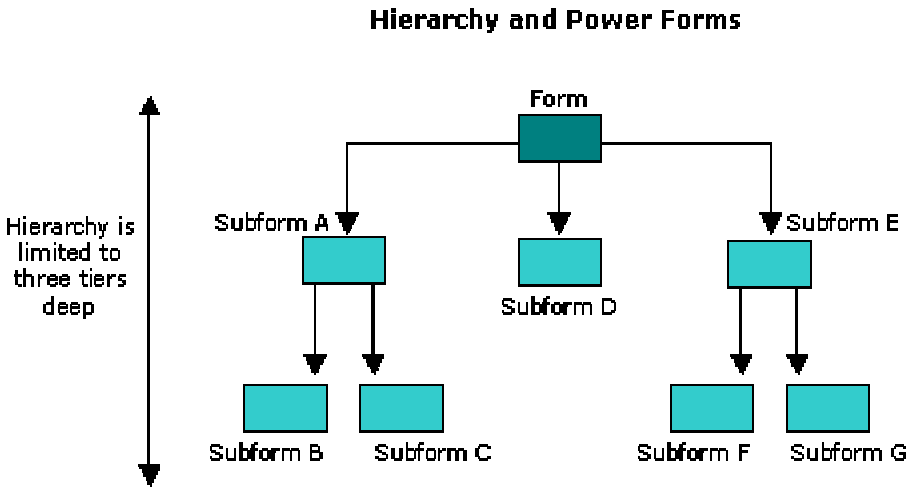
**Note**

Data flow follows the parenting structure in the business view hierarchy.

A Subform can communicate only with its parent or its children. Logic from the event rules (ER) on the Subform, or its subcontrols, does not flow to or from the other Subforms. Related information does not appear in the available objects.

---

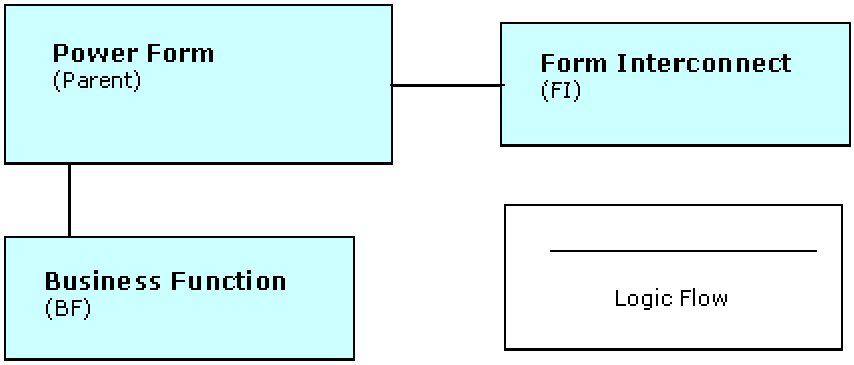
The following graphic illustrates the general hierarchy scheme of Power Forms:



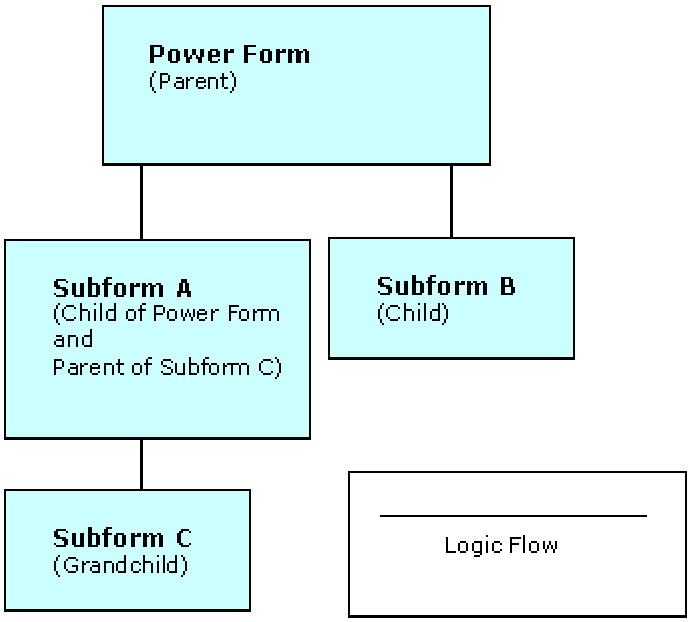

| Form | Passes Logic To and From |
|---|---|
| Power Form | Subforms A, D, and E |
| Subform A | Power Form, Subforms B and C |
| Subforms B and C | Subform A, Power Form |
| Subform D | Power Form |
| Subform E | Power Form, Subforms F and G |
| Subforms F and G | Subform E, Power Form |

## Examples of the Logic Flow of Power Forms

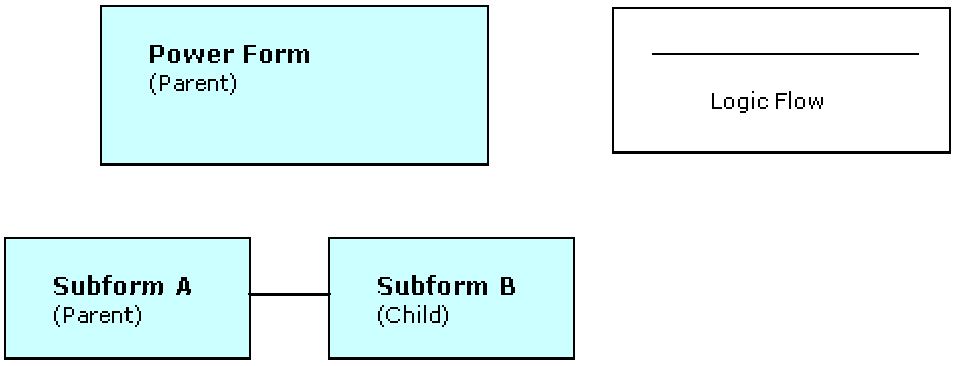
The following flowchart illustrates the flow of logic between a Power Form and a business function, and the Power Form and a form interconnect:




The following flowchart illustrates the flow of logic between a Power Form as a parent, two Subforms as children (Subforms A and B), and a Subform as a parent and a child (Subform A):



The following flowchart illustrates the flow of logic between Subform A and Subform B. The flow of logic occurs only between Subform A (Parent) and Subform B (Child). The Subforms do not communicate with the Power Form:



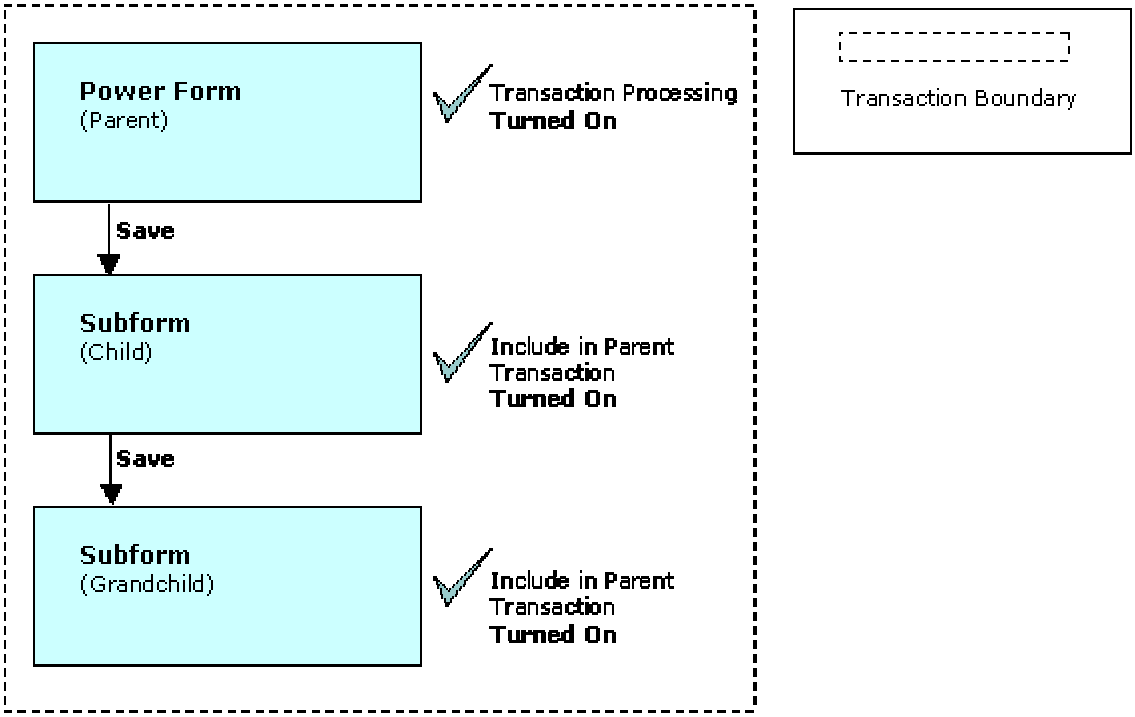

# Transaction Processing for Power Forms and Subforms

Power Forms include the Transaction feature, which functions the same as the other form types. By default, it is scoped to OK processing; however, the scope can be extended to business functions, table I/O, and form interconnects.

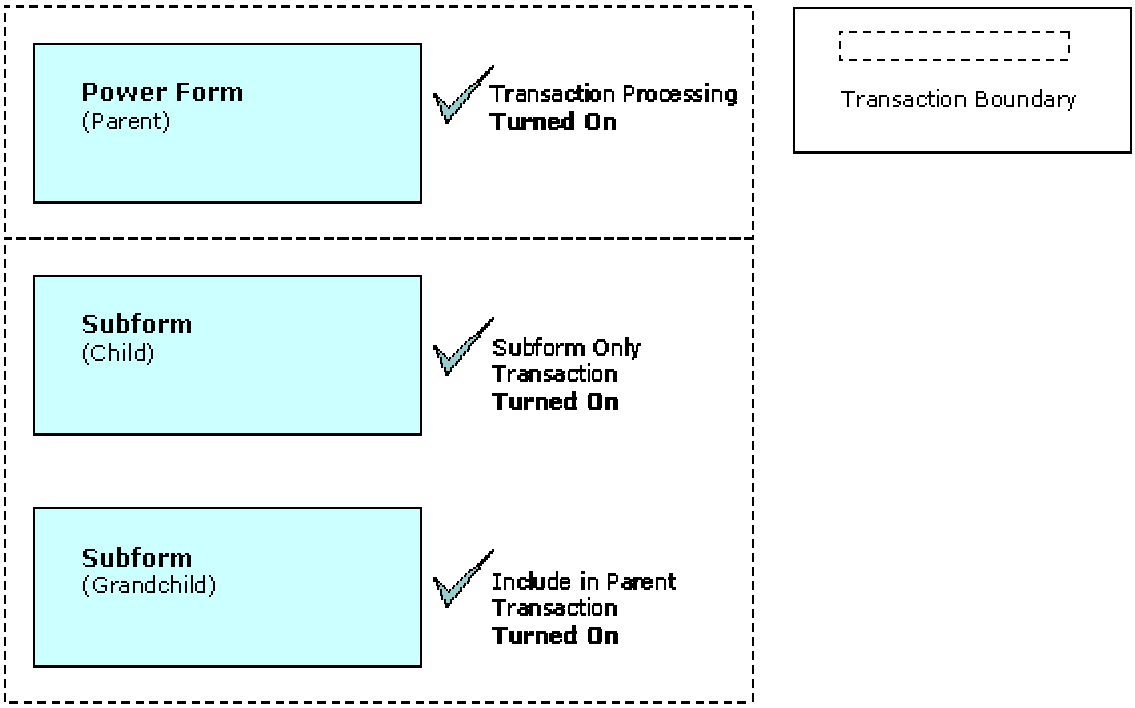
Subforms are scoped to Save processing; however, the scope can be extended to business functions, table I/O, and form interconnects. You can use the following transaction processing settings for Subforms:

| | |
|---|---|
| **Transaction Disabled (default)** | No transaction processing occurs for this form. |
| **Include in Parent Transaction** | If the form is called within the parent's transaction boundary (OK or Save processing), the transaction will be included in the parent's transaction boundary. |
| **Subform Only Transaction** | Save processing for the Subform has its own transaction boundary, which is independent of all boundaries. |

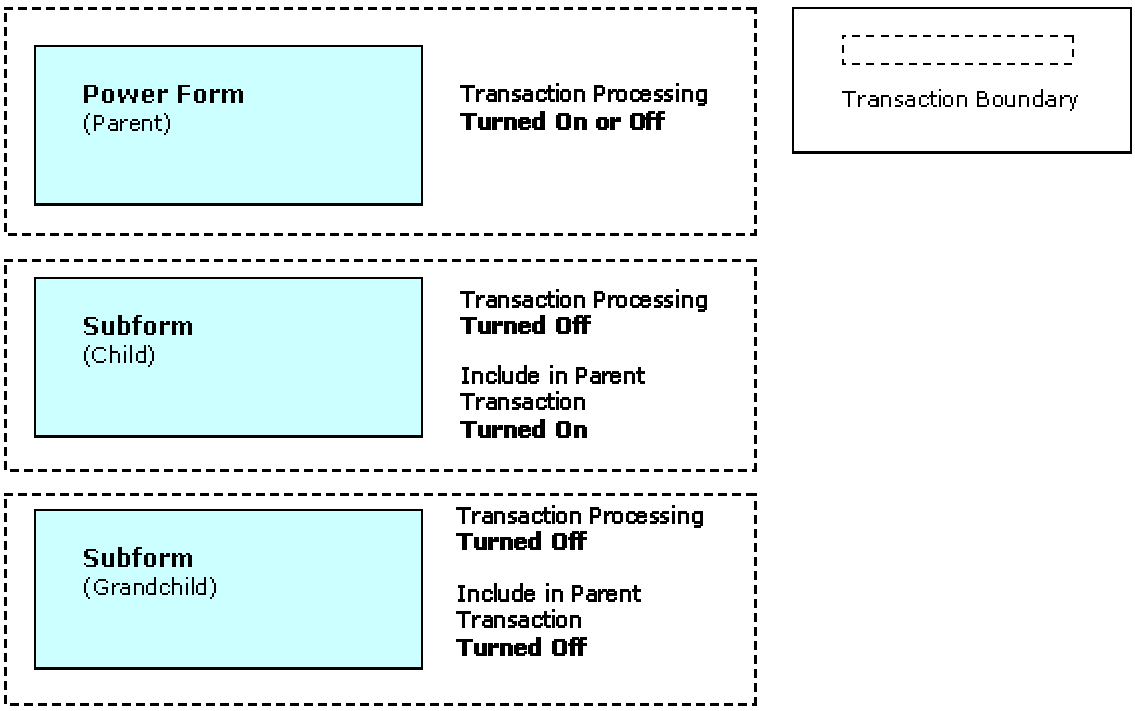
The following graphic shows how to include all of the subforms in the form save transaction. If you rollback data on any one form, the system rolls back the changes on all of them.



Below, the child is not included in the form's transaction, which means that a rollback on the parent level will not affect any of the Subforms. However, all of the Subforms are included in the same transaction, so any rollback on the Subforms will affect all of the Subforms.




In all these scenarios, transactions are local and will not affect appear on any other form.



### Prerequisite

- Ensure that you understand the following tasks in the *Form Design Aid Guide*:
  - *To create a form*
  - *To set properties for a form*

### ► To create a Power Form

*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 Aid, choose either Power Edit or Power Browse from the Form menu.
2. On Form Properties, complete the properties for your form and then click OK.
3. Choose Subform from the Insert menu, and then position your Subform on the Power Form.
4. Click the Subform, right click, and then choose Properties from the menu.
5. Complete the properties for the Subform and then click OK.
6. Complete steps 1 through 5 for as many Subforms as you need.

**Note**

If your development task requires the use of many Subforms, you might consider using tab form Subforms.

### See Also

- *Form Properties* in the *Form Design Aid Guide* for a description of the purpose of the different property values and a discussion of why you might enable or disable some values
- *Menu/Toolbar Exits* in the *Form Design Aid Guide* for instructions about how to add menus and toolbar exits to a form
- *To map the parent form's variables to a child Subform* in the *Form Design Aid Guide* how to map variables

### ► To create a Subform as a standard control

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

1. Create a Power Form, if necessary.
2. Choose Subform from the Insert menu.
3. Position Subform on the Power Form and right-click to place it.
4. From the right-click menu, choose Properties.

5. On Subform Properties, assign all applicable properties. Ensure that you complete the following properties on the General and the Mapping Link tabs:

   On the General tab:

   - Parent

   On the Mapping Link tab:

   - Map the parent form's variables to the child Subform.

   **Note**

   See *To map the parent form's variables to a child Subform* in the *Form Design Aid Guide* for information on how to map variables.

6. Click OK.

### ► To create a Subform as a tab page

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

1. Click the tab control in which you want to insert the Subform.
2. From the Insert menu, choose Subform.

## See Also

- *Putting Tabs on a Form* in the *Form Design Aid Guide* for more information on tab controls.

### ► To convert from a standard Subform to a Subform tab page

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

1. Locate the Subform on the form.
2. Drag the Subform and drop it onto a tab control.

   You can drag a subform by its border.

### ► To convert from a Subform tab page to a standard Subform

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

1. Click the Subform tab page header in which you want to switch from a Subform tab page to a standard Subform.
2. Drag the Subform tab page by its tab and position the Subform on the form.

## See Also

- *Form Properties* in the *Form Design Aid Guide* for a description of the purpose of the different property values and a discussion of why you might enable or disable some values
- *Menu/Toolbar Exits* in the *Form Design Aid Guide* for instructions about how to add menus and toolbar exits to a form
- *To map the parent form's variables to a child Subform* in the *Form Design Aid Guide* how to map variables

## ► To map the parent form variables to a child Subform

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

1. Click the parent form that you want to map, right-click, and then choose Properties from the right-click menu.
2. Click the General tab.
3. Using the drop-down menu in the following field, choose the parent form.
   - Parent

   **Warning**

   Changing the parent assignment after mapping links have been set or event rules have been developed could cause your application to fail.

4. Click the Mapping Link tab.
5. Using the drop-down menu in the following field, choose the child Subform.
   - Link to:

     A list of child Subforms
6. Choose an element of the child's interface in the right pane of the Mapping Link tab.

   The right pane contains a list of data structure elements.
7. Choose a variable of the child's interface in the left pane of the Mapping Link tab and set it.

   The left pane contains a list of all the available objects for mapping.
8. Choose one of the following mapping directions:
   - From parent to child
   - From child to parent
   - Bidirectional
9. Repeat steps 5 through 8 until all mappings are defined.

### Subform System Functions

The following table describes the Subform system functions for an event:

**Enable Subform**

Use this system function to enable all controls on the Subform to the end user.

**Parameter:** Subform

- **I/O:** Input
- **Required:** Yes

**Return Values:** None

**Special Handling Instructions and Prerequisites:** None

**Available Events:** All Power Browse forms, Power Edit forms, and Subforms events

**Related Tables:** None

**Related System Function(s):** Disable Subform

**Related Topics:** Disable Subform

**Disable Subform**

Use this system function to disable all controls (except action buttons) on the Subform. Disabling the Subform makes it read-only.

**Parameter:** Subform

- **I/O:** Input
- **Required:** Yes

**Return Values:** None

**Special Handling Instructions and Prerequisites:** None

**Available Events:** All Power Browse forms, Power Edit forms, and Subforms events

**Related Tables:** None

**Related System Function(s):** Enable Subform

**Related Topics:** Enable Subform

**Hide Subform**

Use this system function to hide Subforms.

**Parameter:** Subform

- **I/O:** Input
- **Required:** Yes

**Return Values:** None

**Special Handling Instructions and Prerequisites:** None

**Available Events:** All Power Browse forms, Power Edit forms, and Subforms events

**Related Tables:** None

**Related System Function(s):** Show Subform

**Related Topics:** Show Subform

**Show Subform**

Use this system function to show Subforms.

**Parameter:** Subform

- **I/O:** Input
- **Required:** Yes

**Return Values:** None

**Special Handling Instructions and Prerequisites:** None

**Available Events:** All Power Browse forms, Power Edit forms, and Subforms events

**Related Tables:** None

**Related System Function(s):** Hide Subform

**Related Topics:** Hide Subform

**Update Parent**

Use this system function to update the mapping link values between the calling Subform and its parent. Items that are mapped both bi-directionally (child-to-parent and parent-to-child) and in a single direction (from child to parent) are updated because of this system function call.

**Parameter:** None

- **I/O:** None
- **Required:** No

**Return Values:** None

**Special Handling Instructions and Prerequisites:** This system function updates the mapping link values between the form that calls the system function and its parent form. The parent form can be a Power Browse form, Power Edit form, or another Subform. Both items that are mapped bi-directionally (child-to-parent and parent-to-child) and in a single direction (from child to parent) are updated because of this system function call. The latest Subform Interconnect (SI) values on the Subform are copied to the designated form controls on the parent form.

**Available Events:** All Power Browse forms, Power Edit forms, and Subforms events

**Related Tables:** None

**Related System Function(s):** None

**Related Topics:** Notified Child and Notify Parent

**Notify Parent**

Use this system function to trigger the Notified By Child event to run on the parent form of the calling Subform.

**Parameter:** None

- **I/O:** None
- **Required:** No

**Return Values:** None

**Special Handling Instructions and Prerequisites:** Mapping link values are not updated automatically. To update mapping links, use system function Update Parent. You should only use this system function on a Subform.

**Available Events:** All Subform events

**Related Tables:** None

**Related System Function(s):** Update Parent updates the mapping link values from the calling Subform to its parent.

**Related Topics:** Update Parent

**Get Error Count**

Use this system function to return the number of errors on the Subform.

**Parameter:** Subform

- **Description:** The Subform on which the errors occur.
- **I/O:** Input
- **Required:** Yes

**Parameter:** Number

- **Description:** The number of errors on the Subform on which the warnings occur.
- **I/O:** Output
- **Required:** Yes
- **Note:** This output parameter must have either an integer or a math numeric data dictionary type.

**Return Values:** Number; the number of errors on the Subform.

**Special Handling Instructions and Prerequisites:** None

**Available Events:** All Power Browse forms, Power Edit forms, and Subforms events

**Related Tables:** None

**Related System Function(s):** Get Warning Count

**Related Topics:** Get Warning Count

**Get Warning Count**

Use this system function to return the number of warnings on the Subform.

**Parameter:** Subform

- **Description:** The Subform on which the warnings occur.
- **I/O:** Input
- **Required:** Yes

**Parameter:** Number

- **Description:** The number of warnings on the Subform on which the warnings occur.
- **I/O:** Output
- **Required:** Yes
- **Note:** This output parameter must have either an integer or a math numeric data dictionary type.

**Return Values:** Number; the number of warnings on the Subform.

**Available Events:** All Power Browse forms, Power Edit forms, and Subforms events

**Related Tables:** None

**Related System Function(s):** Get Error Count

**Related Topics:** Get Error Count

**Get Subform ID**

Use this system function to return the unique control ID of the calling Subform. You should only use this system function on a Subform.

**Parameter:** Subform ID—holds the ID of the calling Subform

- **I/O:** Output
- **Required:** Yes
- **Notes:** The output parameter must have a numeric data type, including integer, handle, and math numeric.

**Return Values:** Returns the current Subform control ID.

**Special Handling Instructions and Prerequisites:** You should only use this system function on a Subform.

**Available Events:** All Subform events

**Related Tables:** None

**Related System Function(s):** None

**Related Topics:** None

**Notify Child**

Use this system function to notify a child Subform and trigger the Notified By Parent event to run on the child Subform.

**Parameter:** Subform

- **I/O:** Input
- **Required:** Yes

**Return Values:** None

**Special Handling Instructions and Prerequisites:** You should only use this system function on a Power Form or a Subform that is a parent to another Subform.

This system function first updates the mapping link. All items that are mapped from parent to child are updated with the latest parent control values.

Second, the Notified By Parent Event Rule executes for the child Subform.

**Available Events:** All Power Browse forms, Power Edit forms, and Subforms events

**Related Tables:** None

**Related System Function(s):** None

**Related Topics:** None

**Trigger Default Child Action**

Use this system function to call an action button on the child form. As a result, the default runtime processing for that button and the event rules for the button are executed.

**Parameter:** Subform

- **Description:** The child Subform whose action button the system calls.
- **I/O:** Input
- **Required:** Yes

**Parameter:** Default Action

- **Description:** The default action button that the system calls.
- **I/O:** Input
- **Required:** Yes
- **Note:** For Subforms on Power Edit forms, valid values are:
  - Save
  - Delete
  - Find
  - Clear
  - Select

  For Subforms on Power Browse forms, valid values are:
  - Find
  - Select

**Return Values:** None

**Special Handling Instructions and Prerequisites:** If the action button does not exist on the child form, no action is taken.

This system function is called from the parent form, which can be a Power

Browse form, a Power Edit form, or a Subform.

The end result depends on the designated action button. Before any of the default action button Event Rules (ER) are executed, the Subform Interconnect (SI) values update with the latest parent control values.

**Available Events:** All Power Browse forms, Power Edit forms, and Subforms events

**Related Tables:** None

**Related System Function(s):** None

**Related Topics:** None

# 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 PeopleSoft 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.

**Note**

You cannot place Parent/Child controls on either Power Forms or Subforms.

- 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. Power Forms can have any number of tab controls, but all other forms are restricted to one.

**Note**

You cannot use Tab Control on a Subform that is a tab page.

- 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.

- Subform

  Use Subforms to allow users multiple data views.

**Note**

Although a Subform has some properties of a control (for example, size, position, visible, enabled, and so on), its primary function is a logical container with all the essential functionality of a form (for example, engine behaviors to find, delete, save, and so on). See *Power Forms* and *To create a power form* in the *Form Design Aid Guide* for more information about using Subforms.

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 Form Design Aid, 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**

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 Layout menu.

   Form Design Aid 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.

   **Note**

   Subforms contain their own sequence orders for objects on the Subform. Therefore, you set the tab sequence on each Subform within the Subform itself. The Subform, then, becomes a single object in the tab sequence for the Power Form.

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

## Data Dictionary Triggers

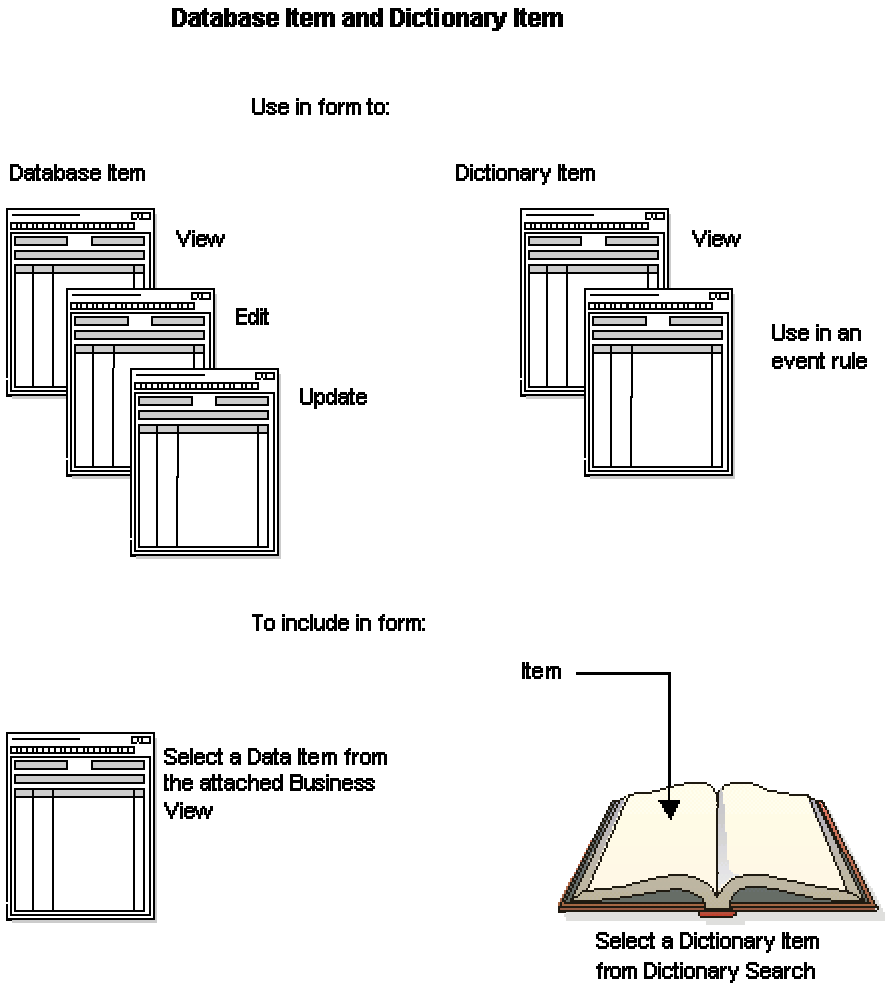
You use triggers to initiate display and edit routines that are associated with data items at application runtime. When you create a data dictionary item, you define its default triggers at that time.

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.




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.

Subforms and message forms do not have a toolbar; instead, users must use push buttons. The push buttons behave the same as the standard push buttons for other form types.

# 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.

### See Also

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

### ► 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.

# 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 using the following options:
  - Disable Page-at-a-Time Processing

    Allows you to either 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.

  - Automatically Find On Entry

    The fetch logic initializes as the form opens.

- Control how a user enters data into a grid using the following options:
  - Clear Grid After Add

    On Header and Headerless Detail forms, allow the user to continue adding lines to the grid after clicking OK.

  - No Adds On Update Grid

    Prevent users from adding rows to a grid, although the user can change existing rows.

- Control how the runtime engine processes the rows in the grid.
  - Process All Rows in Grid

    Cause the runtime engine to process all of the rows in the grid, not just the rows that have been added or changed.

- Control how the HTML grid processes:
  - Multi-line Edit

    Allows the user to edit multiple input fields simultaneously.

  - Hide HTML Row Selector

    Hides the default radio button or check box that appears to the left of the row.

  - Use Alternate Grid Format

    Specifies that the system should display the user-defined grid format defined in the HTML Alternate Grid Format setting.

  - Grid Row Count

    Defines how many grid rows display on a page.

  - HTML Alternate Grid Format

    If the Use Alternate Grid Format option is turned on, the system displays the grid format defined here. You must supply a valid HTML string.

  - Support Multiple Currencies

    If the Support Multiple Currencies option is turned on, you can configure different rows in the same grid to contain data of multiple currencies. Modifying the currency in one row does not affect any other rows.

### ► 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.

## Showing Multiple Currencies per Column

**Note**

The Support Multiple Currencies option is supported for HTML applications only.

When a column has the Support Multiple Currencies option enabled, the runtime engine assumes that each cell contains its own currency setting, and it formats each cell based on that cell's currency decimal setting. The runtime engine will not apply the currency settings to other grid rows, however. Therefore, the application needs to apply currency to each grid row individually. For example, the Amount column in row 1 might have a JPY currency type and be formatted with no decimals, while the Amount column in row 2 might have a USD currency type and be formatted with two decimals.

When a column has the Support Multiple Currencies option disabled, the runtime engine assumes that all of the cells in that column share the same currency setting, and so it applies that currency setting to other grid rows. Therefore, if you specify the currency setting in one row, the system overwrites the currency setting for all the other rows in the grid to match. This feature offers a performance benefit for those grids that contain only one currency because the application needs to specify a currency setting to one grid row only to affect the entire grid.

The following currency rules apply:

- When assigning values using conventions such as target = source, if the source object does not have any currency information (currency code = null or empty string), then the target object keeps its own currency.
- When a GB object is cleared, the currency code and currency decimal information for that column is not cleared.

# Parent/Child Controls

The Parent/Child control is a composite control with a tree on the left and a grid on the right. The control is used to provide a hierarchical view of the business data. Users can resize the tree part of the control during runtime. The tree and grid portions use the same business view.

If you place the parent/child control on a Power Edit form, an embedded Subform on a Power Edit form, or an editable reusable Subform, then the parent/child will be editable within the following parameters:

- Both tree column and grid columns are editable.
- Columns that define the parent child key relationship are not editable.
- Runtime provides format and validation for grid cells and tree nodes, similar to the one provided for regular grid.
- The column events column is exited, column is exited and changed inline, column is exited and changed – asynch events are supported for the tree column and all grid columns.
- The row events row is exited, row is exited and changed inline, row is exited and changed – asynch are supported for parent/child control.
- The event sequence is the same as the grid event sequence.
- Form default buttons such as OK and Delete work for the parent/child control as they do for the grid control.

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:

- 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
- Enable or disable location indicator
- Select a column to be used as ID column

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

## 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.

For each node, the parent/child control automatically generates and updates hierarchical numbers called location indicators. If you enable the feature at design time, then users will see the location indicators in the tree. To enable the feature, click Location Indicator Feature on the Tree Control Options tab in the parent/child control properties.

An application can specify a column as the ID column. The value in this column will be used as a unique identifier for that row. The node ID column can be any data type, as long as it is unique. Use the system functions, Insert Grid Buffer Row By Node ID and Get Related Node ID, to specify a column. Node ID column and these system functions work together: Node ID column needs to exist for these system functions to work. If none of these system functions is called, you do not need to specify a node ID column.

## 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 GCs are copied to GBs on the Begin Drag Operation event.

- 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.

### Node ID Column

When application logic is used to load the tree instead of the runtime logic, the application must often insert a tree node at a specific location based on data column values. To achieve this, at design time, select a column to be the node ID column. The value in this column will be used as a unique identifier for that row. The node ID column can be any data type, but must be unique.

After you specify a node ID column, use the "Insert Grid Buffer Row Under Node ID" system function to insert a row under any other row, identified by its node ID value. Use this method to allow an application to insert tree nodes at any position of the tree.

You can also traverse the entire tree using the "Get Related Node ID" system function. This system function returns the node ID value of a parent or sibling node of a given node.

You must specify a node ID column for both of these system functions to work properly.

### Location Indicators

For each node, the parent/child control can generate a hierarchical number – the location indicator. If you enable this feature at design time, then users will see the location indicators in the tree. To enable the feature, click Location Indicator Feature on the Tree Control Options tab in the parent/child control properties. When a tree node is moved, deleted, or inserted into the tree, the system updates the impacted tree nodes' location indicators.

# 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.

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.

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

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.

# Form and Control Processing

Processing for the following form types is discussed in this section:

- Find/browse
- Parent/child
- Fix/inspect
- Header detail
- Headerless detail
- Search/select
- Power Browse and Power Edit
- Message form

This section also provides information about edit and grid control processing.

# 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 using the following options:
  - Disable Page-at-a-Time Processing

    Allows you to either 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.

  - Automatically Find On Entry

    The fetch logic initializes as the form opens.

- Control how a user enters data into a grid using the following options:
  - Clear Grid After Add

    On Header and Headerless Detail forms, allow the user to continue adding lines to the grid after clicking OK.

  - No Adds On Update Grid

    Prevent users from adding rows to a grid, although the user can change existing rows.

- Control how the runtime engine processes the rows in the grid.
  - Process All Rows in Grid

    Cause the runtime engine to process all of the rows in the grid, not just the rows that have been added or changed.

- Control how the HTML grid processes:
  - Multi-line Edit

    Allows the user to edit multiple input fields simultaneously.

  - Hide HTML Row Selector

    Hides the default radio button or check box that appears to the left of the row.

  - Use Alternate Grid Format

    Specifies that the system should display the user-defined grid format defined in the HTML Alternate Grid Format setting.

  - Grid Row Count

    Defines how many grid rows display on a page.

  - HTML Alternate Grid Format

    If the Use Alternate Grid Format option is turned on, the system displays the grid format defined here. You must supply a valid HTML string.

  - Support Multiple Currencies

    If the Support Multiple Currencies option is turned on, you can configure different rows in the same grid to contain data of multiple currencies. Modifying the currency in one row does not affect any other rows.

**► 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.

# Process Flow for Find/Browse Forms

Find/Browse forms are used to query business views and select records from business views for operations.

## Default Flags

The system does not check by default any form option flags on Find/Browse forms. The only form option flag that can affect this form type is the "Fetch On Grid Business View" flag. Choosing any of the other form option flags does not have any effect on the form processing.

Find/Browse forms and parent/child browse forms are the only forms that have the "Entry Point" property activated by default. Find/Browse forms are typically the entry point into applications. The application developer can deactivate the entry point property.

## Dialog Initialization

The system performs the following tasks in the order shown:

- Initialize Thread Handling.
- Initialize Error Handling Process.
- Initialize Business View Columns.
- Initialize Form Controls.
- Initialize Grid Fields.
- Initialize Static Text.
- Initialize Helps.
- Initialize Event Rules Structures.
- Create Toolbar.
- Load Form Interconnection data into corresponding business view columns and filter fields, if any.
- Perform Event Rules: Dialog is Initialized
- Perform Event Rules: Post Dialog is Initialized
- Begin Detail Data Selection and Sequencing (if the grid option "Automatically Find On Entry" is activated).

## Header Data Retrieval

Find/Browse forms do not have header records.

## Detail Data Selection and Sequencing

The system creates an internal structure that represents the data selection and data sequencing requirements specified by the user. The system then passes this to the database engine to perform the actual database select and sequencing. The data used for selection is based on values from filter fields and QBE columns. The system holds the data until the data is retrieved.

If the form option flag called "Fetch On Grid Business View" is not checked, the following two actions occur:

- Select and Sequence
- Begin Data Retrieval

If the form option flag "Fetch On Grid Business View" is checked, the system does not retrieve data

## Data Retrieval

The system issues a request to the JDEKRNL, which actual fetches the data from the database. The system reads one record at a time; it performs the following processing for each record:

- Attempts to fetch a record from the database.
- If it succeeds, the following processing occurs:
  - Copy the data into the business view data structures.
  - Perform Event Rules: Grid Record is Fetched
- If the application developer has not chosen to suppress the writing of this grid record, the following occurs:
  - Copy the business view data into the grid data structures.
  - Perform Event Rules: Write Grid Line - Before
  - Add the row to the grid. The row now exists in the grid control.
  - Perform Event Rules: Write Grid Line - After
  - Clear the grid data structures for reading the next record
  - Remove the suppress grid line flag

The previous steps occur for each record read from the database. After all records have been read, the following processing occurs:

- Perform Event Rules: Last Grid Record Has Been Read

## Closing Form

The system performs the following tasks in the order shown:

- Perform Event Rules: End Dialog
- Load Form Interconnection data from corresponding business view columns, if any.
- Terminate Error Handling.
- Terminate Thread Handling.
- Terminate Helps.
- Free all structures for form, including structures for business view columns, form controls, grid columns, and event rules.
- Destroy the window.

# Menu/Toolbar Items

This section discusses the following menu and toolbar items:

- Select
- Close
- Delete
- Find
- Copy
- Add
- User Defined Items

## Select

Select is a standard item that is automatically placed on Find/Browse forms. No default processing exists for Select on Find/Browse forms. Select acts as a user defined item.

## Close

Close is a standard item that automatically appears on Find/Browse forms. It closes the form.

- Perform Event Rules: Button Clicked
- Perform Event Rules: Post Button Clicked
- Begin Closing Form

## Find

Find is a standard item that automatically appears on Find/Browse forms. When the user clicks it, it is the signal to the runtime engine to call the database and reload the grid based on the information in the filter fields.

- Perform Event Rules: Button Clicked
- If no errors exist in any filter fields, do the following:
    - Begin Data Selection and Sequencing.
    - Perform Event Rules: Post Button Clicked

## Delete

Delete is a standard item that can be added to Find/Browse forms. It deletes a record in the grid from the database.

- Perform Event Rules: Button Clicked
- Do the following for each selected grid row that you can delete:
  - Copy the grid row data into the Business View structures.
  - Remove Suppress Delete flag.
  - Perform Event Rules: Delete Grid Rec Verify – Before.
- If the Suppress Delete flag is not set, do the following:
  - Display Delete Confirmation dialog. If the user clicks NO or CANCEL the system skips therest of this processing.
  - Perform Event Rules: Delete Grid Rec Verify – After.
  - Perform Event Rules: Delete Grid Rec from DB – Before.
  - Delete the record in the Business View from the database.
  - Delete the grid row from the grid control.
  - Perform Event Rules: Delete Grid Rec from DB – After.
  - Perform Event Rules: All Grid Recs Deleted.
  - Perform Event Rules: Post Button Clicked.

## Copy

Copy is a standard item that can be added to Find/Browse forms. The copy function allows users to duplicate and rename an object. No default processing exists for the copy function on Find/Browse forms. They act as user-defined items. If a form interconnection is placed on the Button Clicked event, then the called form opens in copy mode.

## Add

Add is a standard item that can be added to Find/Browse forms. No default processing exists for add on Find/Browse forms. They act as user-defined items. If a form interconnection is placed on the Button Clicked event, then the called form will open in add mode.

## User Defined Items

User-defined items are nonstandard items that you can add to Find/Browse forms to perform specialized processing that is not handled by the standard items.

- Perform Event Rules: Button Clicked
- Perform Event Rules: Post Button Clicked

# Process Flow for Parent/Child Browse Forms

A parent/child browse form is used to query a business view and represent the data in a hierarchical manner. Records can also be selected from that business view for an operation. The following sections describe the processing flow of a parent/child browse form type.

## Default Flags

No form options flags are checked by default on parent/child browse forms. The only form option flag that can have an effect for this form type is the "Fetch On Grid Business View" flag. Checking any of the other form option flags does not have any effect on the form processing.

The parent/child browse forms (along with find/browse forms) are the only forms that have the "Entry Point" property checked by default. Parent/child browse forms are typically the entry points into applications. The entry point property can be unchecked by the application developer.

## Dialog Initialization

These are the actions, in order, that the runtime engine executes to initialize a dialog box.

- Initialize Thread Handling.
- Initialize Error Handling Process.
- Initialize Business View Columns.
- Initialize Form Controls.
- Initialize Grid/Tree Fields.
- Initialize Static Text.
- Initialize Helps.
- Initialize Event Rules Structures.
- Create Toolbar.
- Load Form Interconnection data into corresponding business view columns and filter fields, if any.
- Perform Event Rules: Dialog is Initialized
- Perform Event Rules: Post Dialog is Initialized
- Begin Detail Data Selection and Sequencing if the grid option flag "Automatically Find On Entry" is checked.

## Header Data Retrieval

Parent/child browse forms have no header records.

# Detail Data Selection and Sequencing

The system creates an internal structure that represents the data selection and data sequencing requirements specified by the user. This structure is then passed to the database engine to perform the actual database selection and sequencing. The data is then held until the data is retrieved in the next step.

The data used for selection is pulled from filter fields and QBE columns.

If the form option flag "Fetch On Grid Business View" is not checked, the following occur:

- Select and Sequence
- Begin Data Retrieval

# Data Retrieval

A request is issued to the JDEKRNL, which performs the actual fetch of the data from the database. It reads one record at a time; and for each record, it performs the required processing. The data retrieval is performed in two different ways in the parent/child browse form. The fetch performed to get the first level nodes in the tree is similar to the one performed in the find/browse form.

## Data Retrieval for the First Level Node in the Tree

- Attempt to fetch a record from the database
- If a record is fetched, do the following:
  - Copy the data into the business view data structures.
  - Perform Event Rules: Grid Record is Fetched
- If the application developer has not chosen to suppress the writing of this record, do the following:
  - Copy the business view data into the grid data structures.
  - Perform Event Rules: Write Grid Line – Before
  - Add the row to the grid and the corresponding column in Tree. The row and the corresponding tree node now exist in the control.
  - Perform Event Rules: Write Grid Line - After
  - Clear the data structures to read the next record

The previous actions occur for each record read from the database. After all records have been read, the runtime engine performs the following actions:

- If the parent/child browse form has the grid permanently hidden, then expand the header node and move the selection to the first child node under the header. This initiates the events “Tree - Node Level Changed” and “Tree - Node Selection Changed.”
- Perform Event Rules: Last Grid Record Has Been Read

The parent/child browse form also performs data retrieval whenever a particular node on the tree is expanded. However, this fetch is performed only once for each node that is expanded. If a particular node is collapsed and expanded again, then the tree and the grid are replenished from internal structures. The processing performed for node expand event is given below:

### Data Retrieval for Tree Node Expand

- Perform Event Rules: Tree - Node is expanding
- Check for internal flags to see whether the fetch is suppressed. If the fetch is suppressed through the system function, Suppress Fetch on Node Expand, the processing stops here.
- Perform the key substitution (copying child key into parent key) that was set up in the design of the form.
- Attempt to fetch a record from the database.
- If a record is fetched, do the following:
  - Copy the data into the business view data structures.
  - Perform Event Rules: Grid Record is Fetched

  If the application developer has not chosen to suppress the writing of this record, do the following:

  - Copy the business view data into the grid data structures.
  - Perform Event Rules: Write Grid Line - Before
  - Add the row to the grid and the corresponding column in the tree. The row and the corresponding tree node now exist in the control.
  - Perform Event Rules: Write Grid Line - After
  - Clear the data structures to read the next record.
  - Perform Event Rules: Last Grid Record Has Been Read
  - Move the tree selection to the first child node under the expanding node. This initiates the Events: Tree - Node Level Changed and Tree - Node Selection Changed.

## Closing the Form

These are the actions, in order, that the runtime engine executes to close a form.

- Perform Event Rules: End Dialog
- Load Form Interconnection data from corresponding business view columns, if any.
- Terminate Error Handling.
- Terminate Thread Handling.
- Terminate Helps.
- Free all structures for form, including structures for business view columns, form controls, grid columns, and event rules.
- Destroy the window.

# Menu/Toolbar Items

The following is the process flow for the menu/toolbar items on a parent/child form.

### Select

Select is a standard item that is automatically placed on parent/child browse forms. No default processing exists for Select on parent/child browse forms. It acts as a user-defined item.

### Close

Close is a standard item that automatically appears on parent/child browse forms. It closes the form.

- Perform Event Rules: Button Clicked
- Perform Event Rules: Post Button Clicked
- Begin Closing Form

### Delete

Delete is a standard item that can be added to parent/child browse forms. Similar to the find/browse form, the parent/child browse form does the following.

- Perform Event Rules: Button Clicked
- For the selected tree node:
  - Copy the grid row data into the Business View structures.
  - Remove Suppress Delete flag.
  - Perform Event Rules: Delete Grid Rec Verify - Before
- If the Suppress Delete flag is not set, do the following:
  - Display Delete Confirmation dialog. If the user clicks No or Cancel, the rest of this processing is skipped.
  - Perform Event Rules: Delete Grid Rec Verify - After
  - Perform Event Rules: Delete Grid Rec from DB - Before
- If the Suppress Delete flag is not set, do the following:
  - Delete the record in the Business View from the database.
  - This will not delete the child records of the node, if any, from the database. It is your responsibility to delete them from the database.
  - Delete the grid row from the grid control.
  - Perform Event Rules: Delete Grid Rec from DB - After
- Perform Event Rules: All Grid Recs Deleted
- Perform Event Rules: Post Button Clicked

### Find

Find is a standard item that automatically appears on parent/child browse forms. When the user clicks Find, the system signals the runtime engine to call the database and reload the grid based on the information in the filter fields and the QBE columns.

- Perform Event Rules: Button Clicked
- If no errors are in any filter fields, do the following:
  - Begin Data Selection and Sequencing.
  - Perform Event Rules: Post Button Clicked

### Copy

Copy is a standard item that can be added to parent/child browse forms. No default processing exists for copy on parent/child browse forms. The forms work as user defined items. If a form interconnection is placed on the Button Clicked event then the called form will open in copy mode.

### Add

Add is a standard item that can be added to parent/child browse forms. No default processing exists for add on parent/child browse forms. The forms work as user defined items. If a form interconnection is placed on the Button Clicked event, then the called form will open in add mode.

### User-Defined Items

User-defined items are nonstandard items that you can add to parent/child browse forms to perform specialized processing that is not handled by the standard items. These are the actions, in order, that the runtime engine executes to resolve user-defined items.

- Perform Event Rules: Button Clicked
- Perform Event Rules: Post Button Clicked

# Process Flow for Fix/Inspect Forms

A Fix/Inspect form is used to update and insert single data base records. It is also referred to as a Single Record Maintenance form. This form type can also be used to display static information to the user, as well as prompt the user for information. Sign-on screens, for example, prompt the user to enter sign-on information. The following sections describe the processing flow of a Fix/Inspect form.

## Default Flags

None of the form option flags are set by default.

# Dialog Initialization

- Initialize Thread Handling.
- Initialize Error Handling Process.
- Initialize Business View Columns.
- Initialize Form Controls.
- Initialize Static Text.
- Initialize Helps.
- Initialize Event Rules Structures.
- Create Toolbar.
- Load Form Interconnection data into corresponding business view columns.

---

**Note**

Form controls and business view columns are sharing internal memory, so copying into the business view is effectively copying into the internal form control memory locations.

---

- Perform Event Rules: Dialog is Initialized
- If the No Fetch Form Business View is checked, do the following:
  - Perform Event Rules: Post Dialog is Initialized
- If the form is in Add Mode, do the following:
  - Begin Clearing Dialog.
- If the form is in update mode and No Fetch Form Business View is checked, do the following:
  - Change mode to add mode.
  - Display the internal form control values to the screen.
  - Begin Clearing Dialog.
- If the form is in update mode and No Fetch Form Business View is not checked, do the following:
  - Begin Data Retrieval.

## Data Retrieval

A request is issued to the JDEKRNL, which performs the actual fetch of the data from the database. The fetch is based on the information passed to the form through its form data structure.

- Attempt to fetch a record from the database.
- If a record is fetched, do the following:
  - Copy the data into the business view data structures.

---

**Note**

Remember that copying into the business view is effectively copying into the internal form control memory locations.

---

- If a record is found and the form is in Copy Mode, do the following:
  - Switch to Add Mode.
  - Display the internal form control values to the screen.
  - Begin Clearing Dialog.
- If a record is found and not in Copy mode, do the following:.
  - Perform Event Rules: Post Dialog is Initialized
- If no records were fetched, do the following:
  - Switch to Add Mode.
  - Display the internal form control values to the screen.
  - Begin Clearing Dialog.
- Set the default focus based on the resulting data base mode.

## Clearing Dialog

- If the form was called in Copy Mode, do the following:
  - Clear the key (primary index) controls that have the "Do Not Clear After Add" flag unchecked.
- If the form was not called in Copy Mode, do the following:
  - Clear all form controls that have the "Do Not Clear After Add" flag unchecked.
- Perform Event Rules: Clear Screen Before Add
- Perform Event Rules: Post Dialog is Initialized

# Closing Form

- Perform Event Rules: End Dialog
- Load Form Interconnection data from corresponding business view columns, if any.
- Terminate Error Handling.
- Terminate Thread Handling .
- Terminate Helps.
- Free all structures for form, including structures for Business View Columns, Form Controls, and Event Rules.
- Destroy the window.

# Menu/Toolbar Items

The following is the process flow for the menu/toolbar items on a Fix/Inspect Form.

### OK

OK is a standard item that is automatically placed on Fix/Inspect forms. It validates the information on the form and updates or adds to the database through JDEKRNL function calls.

- If any errors or warnings appear on the form, stop OK processing
- Perform Event Rules: Button Clicked
- For each control on the form, do the following:
  - If the current control is a form control and it has not passed validation, do the following:

    Perform Event Rules: Control is Exited

    Perform Event Rules: Control is Exited and Changed - Inline

    Perform Event Rules: Control is Exited and Changed - Async

    Perform Data Dictionary validation.
- If any errors appear on the form, stop OK processing
- If the form is in Add mode, do the following:
  - Perform Event Rules: Add Record to Database - Before
- If the form is in Update mode, do the following:
  - Perform Event Rules: Update Record to Database - Before
- Perform Event Rules: Post Button Clicked

- If form is in Add Mode, do the following:
  - If form was called in Copy mode or the flag “End Form On Add” is checked, do the following:

    Begin Closing Form
  - Else

    Begin Clearing Dialog
- If the form is in Update mode, do the following:
  - If no errors occurred while attempting to update or add to the database, do the following:

    Begin Closing Form

### Cancel

Cancel is a standard item that is automatically placed on Fix/Inspect forms.

- Perform Event Rules: Button Clicked
- Perform Event Rules: Post Button Clicked
- Begin Closing Form

### User-Defined Items

User-defined items are nonstandard items that a developer can add to Fix/Inspect forms to perform specialized processing that is not handled by the standard items.

- Perform Event Rules: Button Clicked
- Perform Event Rules: Post Button Clicked

# Process Flow for Header Detail Forms

A header detail form is used when a relationship exists between the information in the header and the information in the detail area. The Header portion uses one business view, and the detail portion of the form uses another business view. This form type (as well as the headerless detail form type) is referred to as a transaction form. The following sections describe the processing flow of a header detail form type.

## Default Flags

No form option flags are automatically set for this form type. All form option flags can be checked or unchecked by the application developer.

# Dialog Initialization

- Initialize Thread Handling.
- Initialize Error Handling Process.
- Initialize Business View Columns.
- Initialize Form Controls.
- Initialize Grid Fields.
- Initialize Static Text.
- Initialize Helps.
- Initialize Event Rules Structures.
- Create Toolbar.
- Load Form Interconnection data into corresponding business view columns and filter fields, if any.
- Perform Event Rules: Dialog is Initialized
- If the form option flag "Fetch On Form Business View" is checked, do the following:
  - If the form is not in Add mode and was not entered with a Copy button, do the following:

    Perform Event Rules: Post Dialog is Initialized
  - Copy the Form Control values to the screen.
- If the form is in Update mode (includes forms entered with a Copy button), do the following:
  - If the form option flag "Fetch On Form Business View" is checked, do the following:

    Begin Detail Data Selection and Sequencing
  - If the form option flag "Fetch On Form Business View" is unchecked, do the following:

    Begin Header Data Retrieval
- If the form is in Add mode, do the following:
  - Begin Clearing Dialog

## Header Data Retrieval

A key structure is built for the business view of the header based on the header business view columns, or from the filter fields, if any exist. Then a call is made to the database attempting to retrieve the record specified.

- If the database fetch is successful, do the following:
  - Copy the fetched database record to the header Business View Columns
  - If the form was not opened with a Copy button, do the following:

    Perform Event Rules: Post Dialog is Initialized (note that where form controls and business view columns share memory, the form controls will have the values from the database during this event, which may mean blanks in the case of strings and characters)
  - Copy the Business View Columns to the Form Controls.
  - If the grid option "Automatically Find on Entry" is checked, do the following:

    Begin Detail Data Selection and Sequencing
  - If the grid option "Automatically Find on Entry" is unchecked, do the following:

    Begin Add Entry to Row to Grid
- If the database fetch failed, do the following:
  - Begin Clearing Dialog

## Detail Data Selection and Sequencing

An internal structure is now created representing the data selection and data sequencing requirements specified by the user for the detail portion. This structure is then passed to the database engine to do the actual database selection and sequencing. The data is retrieved in the next step.

The data used for selection comes from filter fields, if any exist, or from the key business view columns, if no filter fields exist.

- If the form option flag "Fetch On Grid Business View" is unchecked, do the following:
  - Select and Sequence
  - Begin Data Retrieval
- If the form option flag `Fetch On Grid Business View' is checked, do the following:
  - Begin Add Entry Row To Grid

## Data Retrieval

A request is issued to the JDEKRNL, which performs the actual fetch of the data from the database. The system reads one record at a time; and for each record, the system performs the following processing:

- Attempt to fetch a detail record from the database
- If a record is fetched, do the following:
  - Copy the data into the detail business view data structures.
  - Perform Event Rules: Grid Record is Fetched
  - If the application developer has not chosen to suppress the writing of this grid record, do the following:

    Copy the business view data into the grid data structures.

    Perform Event Rules: Write Everest Grid Line - Before

    Add the row to the grid. The row now exists in the grid control.

    Perform Event Rules: Write Everest Grid Line - After
  - Clear the grid data structures for reading the next record.
  - Remove the suppress grid line flag.

The previous steps occur for each record read from the database. After all records have been read, the runtime engine will then, do the following:

- If the form was opened with a Copy button, do the following:
  - Switch to Add mode.
  - Begin Clearing Dialog once Detail Data Retrieval processing is complete.
- If no records were fetched, do the following:
  - Switch to Add mode.
- Perform Event Rules: Last Grid Record Has Been Read

  (This event runs regardless if records were actually fetched.)
- Begin Add Entry Row to Grid.

## Clearing Dialog

- If the form was called in Copy mode, do the following:
  - Clear the key controls that have the "Do Not Clear After Add" flag unchecked.
- If the form was not called in Copy mode, do the following:
  - Clear all form controls that have the "Do Not Clear After Add" flag unchecked.
- Perform Event Rules: Clear Screen Before Add
- Delete any existing rows.
- Perform Event Rules: Post Dialog is Initialized
- Begin Add Entry Row to Grid.

## Add Entry Row to Grid

- Clear grid data structures.
- Perform Event Rules: Add Last Entry Row to Grid
- Add the row to the grid control.

## Closing Form

- Perform Event Rules: End Dialog
- Load form interconnection data from corresponding business view columns, if any.
- Terminate error handling.
- Terminate thread handling.
- Terminate Helps.
- Free all structures for form, including structures for business view columns, form controls, grid columns, and event rules.
- Destroy the window.

## Menu/Toolbar Items

The following is the process flow for the menu/toolbar items on a header detail form:

### OK

OK is a standard item that is automatically placed on header detail forms. It validates the information on the form and updates or adds to the database through JDEKRNL function calls.

- If any errors or warnings appear on the form, stop OK processing.
- Perform Event Rules: Button Clicked

- For each control on the form, do the following:
  - If the current control is a form control and it has not passed validation, do the following:

    Perform Event Rules: Control is Exited

    Perform Event Rules: Control is Exited and Changed - Inline

    Perform Event Rules: Control is Exited and Changed - Asynch

    Perform Data Dictionary validation.
  - If the current control is a grid control, do the following:

    For each grid row, do the following:

    If the current grid row needs to have Leave Row processing run and the row is updatable, do the following:

    Perform Event Rules: Row is Exited and Changed - Inline for current row

    Perform Event Rules: Row is Exited and Changed - Asynch for current row
- If any errors appear on the form, stop OK processing.
- Delete from the database any grid rows that are in the delete stack. For each grid row in the delete stack, do the following:
  - Copy the information from the delete stack into the grid data structures.
  - Copy the grid data structures into the Business View structures.
  - Perform Event Rules: Delete Grid Record from DB – Before.
  - If the database delete has not been suppressed, do the following:

    Delete the record in the Business View from the database.

    Delete the grid row from the grid control.

    Perform Event Rules: Delete Grid Record from DB - After
- Perform Event Rules: All Grid Recs Deleted from DB
- If the form option flag `Update On Form Business View' is unchecked, do the following:
  - If the form is in Add mode, do the following:

    Perform Event Rules: Add Record to Database - Before

    If the database add has not been suppressed, do the following:

    Add the header business view record to the database.
  - Perform Event Rules: Add Record to Database - After
  - If the form is in Update mode, do the following:

    Perform Event Rules: Update Record to Database - Before

    If the database Update has not been suppressed, do the following:

    Update the record to the database.

    Perform Event Rules: Update Record to Database - After

- If the form option flag “Update On Grid Business View” not unchecked, do the following:
  - For each grid row that was changed or added, do the following:

    Clear the business view data structures.

    Reset the original key values for this row in the business view data structures.

    Copy grid data structures to the business view data structures.

    Copy all non-filter database form controls to the business view data structures.

    Copy all equal filters to the business view data structures.

    If form is in Add mode, do the following:

    Perform Event Rules: Add Grid Rec to DB - Before

    Add the record in the business view data structure to the database.

    Perform Event Rules: Add Grid Rec to DB - After

    If form is in Update mode, do the following:

    Perform Event Rules: Update Grid Rec to DB - Before

    Update the record in the business view data structure to the database.

    Perform Event Rules: Update Grid Rec to DB - After
  - If form is in Add mode, do the following:

    Perform Event Rules: All Grid Recs Added to DB
  - If form is in Update mode, do the following:

    Perform Event Rules: All Grid Recs Updated to DB
- Perform Event Rules: Post Button Clicked
- If form is in Add mode, do the following:
  - If form was called in Copy mode or if the flag `End Form On Add' is checked, do the following:

    Begin Closing Form

    Else

    Begin Clearing Dialog
- If the form is in Update mode and if no errors were encountered while attempting to update or add to the database, do the following:
  - Begin Closing Form.

## Cancel

Cancel is a standard item that is automatically placed on header detail forms.

- Perform Event Rules: Button Clicked
- Perform Event Rules: Post Button Clicked
- Begin Closing Form.

## Delete

Delete is a standard item that can be added to header detail forms. The delete applies to the grid record(s) selected. The actual delete from the database does not happen at this point, so that if the user clicks cancel, the records are not deleted from the database. The delete verification happens when the delete is pressed, and the actual database delete happens when OK is pressed.

- Perform Event Rules: Button Clicked
- For each selected grid row that is deletable, do the following:
  - Remove Suppress Delete flag.
  - Perform Event Rules: Delete Grid Rec Verify - Before
  - If the Suppress Delete flag is not set, do the following:

    Display Delete Confirmation dialog.

    If the user clicks NO or CANCEL the rest of this processing is skipped.

    Perform Event Rules: Delete Grid Rec Verify - After

    If the Suppress Delete flag is not set, do the following:

    If the record was read from the database, do the following:

    Add this record to the delete stack (records to be deleted when the user presses OK)

    Delete the grid row from the grid control.

## Find

Find is a standard item that can be added to header detail forms. When clicked by the user, it is the signal to the runtime engine to call the database and reload the grid based on the selections in the filter fields.

- Perform Event Rules: Button Clicked
- If there no errors in any filter fields, do the following:
  - Switch to Update mode.
  - Begin Detail Data Selection and Sequencing.
  - Perform Event Rules: Post Button Clicked

## User-Defined Items

User-defined items are nonstandard items that a developer can add to header detail forms to perform specialized processing not handled by the standard items.

- Perform Event Rules: Button Clicked
- Perform Event Rules: Post Button Clicked

# Process Flow for Headerless Detail Forms

A Headerless Detail form is used to update and enter records that have information that is common to all records in a selected group. This form type and the header detail form type are referred to as Transaction forms. The following sections describe the processing flow of a Headerless Detail form type.

## Default Flags

The following form option flags are automatically checked for Headerless Detail forms and should not be unchecked:

- Update On Form Business View
- Fetch On Form Business View

Other form option flags can be checked or unchecked by the application developer.

## Dialog Initialization

- Initialize Thread Handling.
- Initialize Error Handling Process.
- Initialize Business View Columns.
- Initialize Form Controls.
- Initialize Grid Fields.
- Initialize Static Text.
- Initialize Helps.
- Initialize Event Rules Structures.
- Create Toolbar.
- Load Form Interconnection data into corresponding business view columns and filter fields, if any.
- Perform Event Rules: Dialog is Initialized
- If the form is not in Add mode, do the following:
  - If the form is not in Copy mode

    Perform Event Rules: Post Dialog is Initialized (note that this event is run immediately after Dialog is Initialized when in Update mode, so the FC values are still at their NULL or zero value).
  - If the grid option "Automatically Find on Entry" is checked, do the following:

    Begin Data Selection and Sequencing.

- If the grid option "Automatically Find on Entry" is unchecked, do the following:

  Begin Add Entry Row to Grid.

- If the form is in Add mode, do the following:

  Begin Clearing Dialog.

## Data Selection and Sequencing

An internal structure is now created representing the data selection and data sequencing requirements specified by the user. This is then passed to the database engine to do the actual database select and sequencing. The data is then held until the data is retrieved in the next step.

The data used for selection is pulled from filter fields if any exist, or from the key business view columns if there are no filter fields.

- If the form option flag "Fetch On Grid Business View" is not checked, do the following:
  - Select and Sequence
  - Begin Data Retrieval
- If the form option flag "Fetch On Grid Business View" is checked, do the following:
  - Begin Add Entry Row To Grid.

## Data Retrieval

A request is issued to the JDEKRNL, which performs the actual fetch of the data from the database. It will read one record at a time and for each record, perform the following processing:

- Attempt to fetch a record from the database.
- If a record is fetched.
  - Copy the data into the business view data structures.
  - Perform Event Rules: Grid Record is Fetched
  - If the application developer has not chosen to suppress the writing of this grid record, do the following:

    Copy the business view data into the grid data structures.

    Perform Event Rules: Write Grid Line - Before

    Add the row to the grid. The row now exists in the grid control.

    Perform Event Rules: Write Grid Line - After
  - Clear the grid data structures for reading the next record.
  - Remove the suppress grid line flag.

The previous steps occur for each record read from the database. After all records have been read, the runtime engine does the following:

- If the form is in Copy mode, do the following:
  - Switch to Add mode.
  - Begin Clearing Dialog once Data Retrieval processing is complete.
- If no records were fetched, do the following:
  - Switch to Add mode.
- Perform Event Rules: Last Grid Record Has Been Read

  **Note**

  This event is run, regardless of whether records were actually fetched.

- Begin Add Entry Row to Grid.

## Clearing Dialog

- If the form was called in Copy mode, do the following:
  - Clear the key controls that do not have the "Do Not Clear After Add" flag checked.
- If the form was not called in Copy mode, do the following:
  - Clear all form controls that do not have the "Do Not Clear After Add" flag checked.
- Perform Event Rules: Clear Screen Before Add
  - Delete any existing grid rows.
- Perform Event Rules: Post Dialog is Initialized
  - Begin Add Entry Row to Grid.

## Add Entry Row to Grid

- Clear grid data structures.
- Perform Event Rules: Add Last Entry Row to Grid
- Add the row to the grid control.

# Closing Form

- Perform Event Rules: End Dialog
- Load Form Interconnection data from corresponding business view columns, if any.
- Terminate Error Handling.
- Terminate Thread Handling.
- Terminate Helps.
- Free all structures for form, including structures for Business View Columns, Form Controls, Grid Columns, and Event Rules.
- Destroy the window.

# Menu/Toolbar Items

The following is the process flow for the menu/toolbar items on a Headerless Detail Form.

### OK

OK is a standard item that is automatically placed on Headerless Detail forms. It validates the information on the form and updates or adds to the database through JDEKRNL function calls.

- If any errors/warnings exist on the form, stop OK processing.
- Perform Event Rules: Button Clicked
- For each control on the form:
  - If the current control is a form control and it has not passed validation, do the following:

    Perform Event Rules: Control is Exited

    Perform Event Rules: Control is Exited and Changed - Inline

    Perform Event Rules: Control is Exited and Changed - Asynch

    Perform Data Dictionary validation.
  - If the current control is a grid control, perform the following for each grid row:

    Perform Event Rules: Row is Exited and Changed - Inline for current row

    Perform Event Rules: Row is Exited and Changed - Asynch for current row
- If errors are on the form, stop OK processing.
- Delete from the database any grid rows that are in the delete stack. See Delete for details. For each grid row in the delete stack:
  - Copy the grid row data into the Business View structures.
  - Copy the grid data structures into the business view data structures.

- Perform Event Rules: Delete Grid Record from DB - Before
- If the database delete has not been suppressed, do the following:

    Delete the record in the Business View from the database.

    Delete the grid row from the grid control.

    Perform Event Rules: Delete Grid Record from DB - After

- Perform Event Rules: All Grid Recs Deleted from DB
- If the form option flag "Update On Grid Business View" is not checked, do the following:
    - For each grid row that was changed or added, do the following:

        Clear the business view data structures.

        Reset the original key values for this row in the business view data structures.

        Copy grid data structures to the business view data structures.

        Copy all non-filter database form controls to the business view data structures.

        Copy all equal filters to the business view data structures.

        If form is in Add mode, do the following:

            Perform Event Rules: Add Grid Rec to DB - Before

            Add the record in the business view data structure to the database.

            Perform Event Rules: Add Grid Rec to DB - After

        If form is in Update mode, do the following:

            Perform Event Rules: Update Grid Rec to DB - Before

            Update the record in the business view data structure to the database.

            Perform Event Rules: Update Grid Rec to DB - After

    - If form is in Add mode, do the following:

        Perform Event Rules: All Grid Recs Added to DB

    - If form is in Update mode, do the following:

        Perform Event Rules: All Grid Recs Updated to DB

- Perform Event Rules: Post Button Clicked
- If form is in Add mode, do the following:
    - If form was called in Copy mode or if the flag "End Form On Add" is checked, do the following:

        Begin Closing Form

    - Else

        Begin Clearing Dialog

- If the form is in Update mode, do the following:
    - If no errors exist for attempting to update/add to the database, do the following:

        Begin Closing Form

## Cancel

Cancel is a standard item that is automatically placed on Headerless Detail forms.

- Perform Event Rules: Button Clicked
- Perform Event Rules: Post Button Clicked
- Begin Closing Form.

## Delete

Delete is a standard item that can be added to Headerless Detail forms. The actual delete from the database does not happen at this point, so that if the user clicks cancel, the records are not deleted from the database. The delete verification happens when delete is pressed, and the actual database delete happens when OK is pressed.

- Perform Event Rules: Button Clicked
- For each selected grid row that is deletable, do the following:
  - Remove Suppress Delete flag.
  - Perform Event Rules: Delete Grid Rec Verify - Before
  - If the Suppress Delete flag is not set, do the following:

    Display Delete Confirmation dialog.

    If the user clicks No or Cancel, the rest of this processing is skipped.

    Perform Event Rules: Delete Grid Rec Verify - After

    If the Suppress Delete flag is not set, and if the record was read from the database, do the following:

    Add this record to the delete stack (records to be deleted if the user presses OK)
  - Delete the grid row from the grid control.

## Find

Find is a standard item that can be added to Headerless Detail forms. When clicked by the user, it is the signal to the runtime engine to call the database and reload the grid based on the selections in the Filter Fields.

- Perform Event Rules: Button Clicked
- If no errors exist in any filter fields, do the following:
  - Switch to Update mode.
  - Begin Data Selection and Sequencing.
  - Perform Event Rules: Post Button Clicked

### User-Defined Items

User-defined items are nonstandard items that a developer can add to Headerless Detail forms to perform specialized processing not handled by the standard items.

- Perform Event Rules: Button Clicked
- Perform Event Rules: Post Button Clicked

# Process Flow for Search/Select Forms

A Search/Select form is used to select a single predetermined field from a record in a predetermined file.

**Important**

Search/Select forms return only one value to the calling form, based on the dictionary alias. If no data dictionary alias matches the value, the first value from the data structure will be returned.

The following sections describe the processing flow of a Search/Select form type. A Search/Select form should be attached as a visual assist only to data items that have the same data type as the form data structure element.

## Default Flags

No form option flags are automatically set for this form type, but application developers often check the Grid option flag called Automatically Find on Entry. The only form option flag that this form type affects is the Fetch On Grid Business View flag. Checking any of the other form option flags does not affect form processing.

## Dialog Initialization

- Initialize Thread Handling.
- Initialize Error Handling Process.
- Initialize Business View Columns.
- Initialize Form Controls.
- Initialize Grid Fields.
- Initialize Static Text.
- Initialize Helps.
- Initialize Event Rules Structures.
- Create Toolbar.
- Load Form Interconnection data into corresponding business view columns and filter fields, if any.

- Perform Event Rules: `Dialog is Initialized`
- Perform Event Rules: `Post Dialog is Initialized`
- If the grid option "Automatically Find On Entry' is checked, do the following:
  - Begin Detail Data Selection and Sequencing

## Header Data Retrieval

Header records do not exist on Search/Select forms.

## Detail Data Selection and Sequencing

The system creates an internal structure that represents the data selection and data sequencing requirements specified by the user. This structure is then passed to the database engine to do the actual database selection and sequencing. The data is then held until the data is retrieved in the next step.

The data used for selection is determined from filter fields.

If the form option flag "Fetch On Grid Business View" is not checked, do the following:

- Select and Sequence
- Begin Data Retrieval

## Data Retrieval

A request is issued to the JDEKRNL, which performs the actual fetch of the data from the database. It reads one record at a time, and, for each record, performs the following processing:

- Attempt to fetch a record from the database
- If a record is fetched, do the following:
  - Copy the data into the business view data structures.
  - Perform Event Rules: `Grid Record is Fetched`
  - If the application developer has not chosen to suppress the writing of this grid record, do the following:
- Copy the business view data into the grid data structures.
- Perform Event Rules: `Write Grid Line - Before`
- Add the row to the grid. The row now exists in the grid control.
- Perform Event Rules: `Write Grid Line - After`
  - Clear the grid data structures for reading the next record.
  - Remove the suppress grid line flag.

The previous steps occur for each record read from the database. The following occurs only once.

- Perform Event Rules: `Last Grid Record Has Been Read`

# Clearing Dialog

Clearing Dialog does not apply to this form type.

# Closing Form

- Perform Event Rules: `End Dialog`
- Load Form Interconnection data from corresponding business view columns, if any.
- Terminate Error Handling.
- Terminate Helps.
- Free all structures for form, including structures for business view columns, form controls, grid columns, and event rules.
- Destroy the window.

# Menu/Toolbar Items

The following is the process flow for the menu/toolbar items on a Search/Select Form.

## Select

Select is a standard item that is automatically placed on Search/Select forms. It returns a value to the form interconnection and closes the form.

- Copy the selected grid row into the business view column.
- Perform Event Rules: `Button Clicked`
- Perform Event Rules: `Post Button Clicked`
- Begin Closing Form.

## Close

Close is a standard item that is automatically placed on Search/Select forms.

- Perform Event Rules: `Button Clicked`
- Perform Event Rules: `Post Button Clicked`
- Begin Closing Form.

### Find

Find is a standard item that can be added to Search/Select forms. When clicked by the user, it is the signal to the runtime engine to call the database and reload the grid based on the selections in the filter fields.

- Perform Event Rules: `Button Clicked`
- Begin Data Selection and Sequencing.
- Perform Event Rules: `Post Button Clicked`

### User-Defined Items

User-defined items are nonstandard items that a developer can add to Search/Select forms to perform specialized processing not otherwise handled by the standard buttons.

- Perform Event Rules: `Button Clicked`
- Perform Event Rules: `Post Button Clicked`

# Process Flow for Power Forms and Subforms

Power Forms provide users with multiple data views. For example, while defining a wine cellar operation, wine makers might be required to search for blend IDs that they want to process, which involves searching for and selecting the blend ID and its associated attributes; these attributes come from several tables. Power Forms allow users to view all of the attributes on one form, rather than on several forms.

You can create the following types of Power Forms:

- Power Browse

  Power Browse forms are used to browse data, and resemble the process flow for Find/Browse forms. You can place Subforms with business views that are specific to them on these forms to allow the user to modify data.

- Power Edit

  Power Edit forms are used to browse data, and resemble the process flow for Fix/Inspect forms. You can place Subforms with business views that are specific to them on these forms to allow the user to display data, modify data, or both.

## Process Flow for Power Browse Form

You use a Power Browse form with a grid to query business views and to select records from business views for system operations. A Power Browse form with a business view should always have a grid with at least one column. You can hide the grid.

## Default Flags

The system does not check any default form option flags on Power Browse forms until you add a business view and a grid to the form. When you add a business view and a grid, the system checks the following options by default:

- Update on grid business view
- Fetch on grid business view

Similar to a Find/Browse form, a Power Form has only one business view. Place this business view on the grid.

---

**Note**

Remember that each Subform on Power Forms can have its own business view.

---

## Dialog Initialization

- Initialize Thread Handling.
- Initialize Error Handling Process.
- Initialize Business View Columns.
- Initialize Form Controls.
- Initialize Grid Fields.
- Initialize Static Text.
- Initialize Helps.
- Initialize Event Rules Structures.
- Create Toolbar.
- Load Form Interconnection data into corresponding business view columns and filter fields, if any.
- Perform Event Rules: Dialog is Initialized.
- Perform Event Rules: Post Dialog is Initialized.
- Begin Detail Data Selection and Sequencing (if the grid option Automatically Find On Entry is turned on).

## Header Data Retrieval

Power Browse forms do not have header records. Similar to a Find/Browse form, a Power Form has only one business view on the grid.

## Detail Data Selection and Sequencing

The system creates an internal structure that represents the data selection and data sequencing requirements specified by the user. The system then passes this structure to the database engine for actual database selection and sequencing. The data used for selection is based on values from filter fields and query-by-example (QBE) columns. The system holds the data until the data is retrieved.

If the Fetch on grid business view form option is turned on, the following two actions occur:

- Select and Sequence
- Begin Data Retrieval

If the Fetch on grid business view form option is not turned on, the system does not retrieve data.

## Data Retrieval

The system fetches data from the database. The system reads one record at a time and performs the following processing for each record:

- Attempts to fetch a record from the database.

  If successful, the following processing occurs:

  - Copy the data into the business view data structures.
  - Perform Event Rules: Grid Record is Fetched.
- If the application developer has not chosen to suppress the writing of this grid record, the following occurs:
  - Copy the business view data into the grid data structures.
  - Perform Event Rules: Write Grid Line - Before.
  - Add the row to the grid. The row now exists in the grid control.
  - Perform Event Rules: Write Grid Line - After.
  - Clear the grid data structures for reading the next record.
  - Remove the suppress grid line flag.

The previous steps occur for each record read from the database.

- After all records are read, the following processing occurs:
  - Perform Event Rules: Last Grid Record Has Been Read.

## Closing Form

- Perform Event Rules: End Dialog.
- Load Form Interconnection data from corresponding business view columns, if any.
- Terminate Error Handling.
- Terminate Thread Handling.
- Terminate Helps.

- Free all structures for form, including structures for business view columns, form controls, grid columns, and event rules.
- Destroy the window.

## Menu/Toolbar Items

The following menu and toolbar items are discussed in this section:

- Select
- Close
- Find
- User Defined Items

### Select

Select is a standard item that is automatically placed on Power Browse forms with grids. No default processing exists for Select on Power Browse forms with grids. Select acts as a user defined item.

### Close

Close is a standard item that automatically appears on Power Browse forms with grids. It closes the form.

- Perform Event Rules: Button Clicked.
- Perform Event Rules: Post Button Clicked.
- Begin Closing Form.

### Find

Find is a standard item that automatically appears on Power Browse forms with grids. When the user clicks Find, the runtime engine is signaled to call the database and reload the grid based on information in the filter fields.

- Perform Event Rules: Button Clicked.

If no errors exist in any filter fields, do the following:

- Begin Data Selection and Sequencing.
- Perform Event Rules: Post Button Clicked.

### User Defined Items

User-defined items are nonstandard items that you can add to Power Browse forms with grids to perform specialized processing that is not handled by standard items.

- Perform Event Rules: Button Clicked.
- Perform Event Rules: Post Button Clicked.

# Process Flow for Power Edit Form

A Power Edit form with a grid allows users to update and enter multiple records simultaneously. Similar to a Headerless/Detail form, a Power Edit form has only one business view. Place this business view on the grid. A Power Browse form with a business view should always have a grid with at least one column. You can hide the grid.

## Default Flags

The system does not check any default form option flags on Power Edit forms until you add a business view and a grid to the form. When you add a business view and a grid, the system checks the following options by default:

- Update on grid business view
- Fetch on grid business view

---

**Note**

Remember that each Subform on a Power Form can have its own business view.

---

## Dialog Initialization

- Initialize Thread Handling.
- Initialize Error Handling Process.
- Initialize Business View Columns.
- Initialize Form Controls.
- Initialize Grid Fields.
- Initialize Static Text.
- Initialize Helps.
- Initialize Event Rules Structures.
- Create Toolbar.
- Load Form Interconnection data into corresponding business view columns and filter fields, if any.
- Perform Event Rules: Dialog is Initialized.
- If the form is not in Add mode, continue as follows:

  If the form is not in Copy mode, do the following:

  - Perform Event Rules: Post Dialog is Initialized. (Note that this event is run immediately after Dialog is Initialized when in Update mode, so the FC values are still at their NULL or zero value.)

If the grid option Automatically Find on Entry is checked, do the following:

- Begin Data Selection and Sequencing.

If the grid option Automatically Find on Entry is not checked, do the following:

- Begin Add Entry Row to Grid.

- If the form is in Add mode, do the following:
  - Begin Clearing Dialog.

## Data Selection and Sequencing

An internal structure is now created representing the data selection and data sequencing requirements specified by the user. This structure is then passed to the database engine for actual database selection and sequencing. The data is then held until the data is retrieved in the next step.

The data used for selection is pulled from filter fields or, if no filter fields exist, from the key business view columns.

If the Fetch on grid business view option is checked, do the following:

- Select and Sequence
- Begin Data Retrieval

If the Fetch on grid business view option is not checked, do the following:

- Begin Add Entry Row To Grid

## Data Retrieval

A request is issued to the JDEKRNL, which performs the actual fetch of the data from the database. It will read one record at a time and for each record, perform the following processing:

- Attempt to fetch a record from the database.
- If a record is fetched.
  - Copy the data into the business view data structures.
  - Perform Event Rules: Grid Record is Fetched.
- If the application developer has not chosen to suppress the writing of this grid record, do the following:
  - Copy the business view data into the grid data structures.
  - Perform Event Rules: Write Grid Line - Before.
  - Add the row to the grid. The row now exists in the grid control.
  - Perform Event Rules: Write Grid Line - After.
  - Clear the grid data structures for reading the next record.
  - Remove the suppress grid line flag.

- The previous steps occur for each record read from the database. After all records have been read, the runtime engine does the following:

  If the form is in Copy mode, do the following:

  - Switch to Add mode.
  - Begin Clearing Dialog. once Data Retrieval processing is complete.

  If no records were fetched, do the following:

  - Switch to Add mode.

  Perform Event Rules: Last Grid Record Has Been Read.

  ---

  **Note**

  This event is run, regardless of whether records were actually fetched.

  ---

  - Begin Add Entry Row to Grid.

## Clearing Dialog

If the form was called in Copy mode, do the following:

- Clear the key controls that do not have the "Do Not Clear After Add" flag checked.

If the form was not called in Copy mode, do the following:

- Clear all form controls that do not have the "Do Not Clear After Add" flag checked.

Perform Event Rules: Clear Screen Before Add.

- Delete any existing grid rows.

Perform Event Rules: Post Dialog is Initialized.

- Begin Add Entry Row to Grid.

## Add Entry Row to Grid

- Clear grid data structures.
- Perform Event Rules: Add Last Entry Row to Grid
- Add the row to the grid control.

## Closing Form

- Perform Event Rules: End Dialog.
- Load Form Interconnection data from corresponding business view columns, if any.
- Terminate Error Handling.
- Terminate Thread Handling.
- Terminate Helps.
- Free all structures for form, including structures for Business View Columns, Form Controls, Grid Columns, and Event Rules.
- Destroy the window.

## Menu/Toolbar Items

The following is the process flow for the menu/toolbar items on a Power Edit form with a grid.

### OK

OK is a standard item that is automatically placed on Power Edit form with grids. It validates the information on the form and updates or adds to the database through JDEKRNL function calls.

- If any errors/warnings exist on the form, stop OK processing.
- Perform Event Rules: Button Clicked.
- For each control on the form:

  If the current control is a form control and it has not passed validation, do the following:

  - Perform Event Rules: Control is Exited.
  - Perform Event Rules: Control is Exited. and Changed - Inline.
  - Perform Event Rules: Control is Exited. and Changed - Asynch.
  - Perform Data Dictionary validation.

  If the current control is a grid control, perform the following for each grid row:

  - Perform Event Rules: Row is Exited and Changed - Inline for current row.
  - Perform Event Rules: Row is Exited and Changed - Asynch for current row.
- If errors are on the form, stop OK processing.
- Delete from the database any grid rows that are in the delete stack.

---

**Note**

See Delete for more information about deleting grid rows that are in the delete stack.

---

For each grid row in the delete stack:

- Copy the grid row data into the Business View structures.
- Copy the grid data structures into the business view data structures.
- Perform Event Rules: Delete Grid Record from DB - Before.
- If the database delete has not been suppressed, do the following:
  - Delete the record in the Business View from the database.
  - Delete the grid row from the grid control.
  - Perform Event Rules: Delete Grid Record from DB - After.
- Perform Event Rules: All Grid Recs Deleted from DB
- If the form option flag "Update on grid business view." is checked, do the following:

  For each grid row that was changed or added, do the following:

  - Clear the business view data structures.
  - Reset the original key values for this row in the business view data structures.
  - Copy grid data structures to the business view data structures.
  - Copy all non-filter database form controls to the business view data structures.
  - Copy all equal filters to the business view data structures.
- If form is in Add mode, do the following:
  - Perform Event Rules: Add Grid Rec to DB - Before.
  - Add the record in the business view data structure to the database.
  - Perform Event Rules: Add Grid Rec to DB - After.
- If form is in Update mode, do the following:
  - Perform Event Rules: Update Grid Rec to DB - Before.
  - Update the record in the business view data structure to the database.
  - Perform Event Rules: Update Grid Rec to DB - After.
- If form is in Add mode, do the following:
  - Perform Event Rules: All Grid Recs Added to DB.
- If form is in Update mode, do the following:
  - Perform Event Rules: All Grid Recs Updated to DB.
  - Perform Event Rules: Post Button Clicked.
- If form is in Add mode, do the following:
  - If form was called in Copy mode or if the flag "End Form On Add" is checked, do the following:
    - Begin Closing Form.
  - Else

        - Begin Clearing Dialog.
- If the form is in Update mode, do the following:

  If no errors exist for attempting to update/add to the database, do the following:

    - Begin Closing Form.

### Cancel

Cancel is a standard item that is automatically placed on Power Edit form with grids.

- Perform Event Rules: Button Clicked.
- Perform Event Rules: Post Button Clicked.
- Begin Closing Form.

### Delete

Delete is a standard item that can be added to Power Edit form with grids. The actual delete from the database does not happen at this point, so that if the user clicks cancel, the records are not deleted from the database. The delete verification happens when delete is pressed, and the actual database delete happens when OK is pressed.

- Perform Event Rules: Button Clicked.
- For each selected grid row that is deletable, do the following:
    - Remove Suppress Delete flag.
    - Perform Event Rules: Delete Grid Rec Verify - Before.
    - If the Suppress Delete flag is not set, do the following:
        - Display Delete Confirmation dialog.

          If the user clicks No or Cancel, the rest of this processing is skipped.

        - Perform Event Rules: Delete Grid Rec Verify - After.
    - If the Suppress Delete flag is not set, and if the record was read from the database, do the following:
        - Add this record to the delete stack (records to be deleted if the user presses OK).
    - Delete the grid row from the grid control.

### Find

Find is a standard item that can be added to Power Edit form with grids. When clicked by the user, it is the signal to the runtime engine to call the database and reload the grid based on the selections in the Filter Fields.

- Perform Event Rules: Button Clicked.
- If no errors exist in any filter fields, do the following:
  - Switch to Update mode.
  - Begin Data Selection and Sequencing.
  - Perform Event Rules: Post Button Clicked.

### User-Defined Items

User-defined items are nonstandard items that a developer can add to Power Edit form with grids to perform specialized processing not handled by the standard items.

- Perform Event Rules: Button Clicked.
- Perform Event Rules: Post Button Clicked.

## Process Flow for Subform

You use Subforms to browse or update records using one business view; you cannot place more than one business view on a Subform. Subforms can contain all Form Design Aid controls with the exception of the Parent/Child control and may or may not have a grid.

### Default Flags

The system does not check any default form option flags on Subforms until you add a business view and a grid to the form. When you add a business view and a grid, the system checks the following options by default:

- Update on form/Subform business view.

  You may disable this option.

- Fetch on form/Subform business view.

  You may disable this option.

### Initialization

Each Subform resides on a parent Power Form. When the Power Form initializes, all of its Subforms are initialized. Subform initialization consists of the following steps.

- Initialize Business View Columns.
- Initialize Form Controls.
- Initialize Grid Fields.

- Initialize Static Text.
- Initialize Helps.
- Initialize Event Rules Structures.
- Begin Detail Data Selection and Sequencing (if the grid option Automatically Find On Entry is turned on).

## Header Data Retrieval

Header data retrieval does not apply to Subfoms.

## Detail Data Selection and Sequencing

The system creates an internal structure that represents the data selection and data sequencing requirements specified by the user. The system then passes this structure to the database engine for actual database selection and sequencing. The data used for selection is based on values from filter fields and QBE columns. The system holds the data until the data is retrieved.

If the Fetch on grid business view form option is turned on, the following two actions occur:

- Select and Sequence
- Begin Data Retrieval

If the Fetch on grid business view form option is not turned on, the system does not retrieve data.

## Data Retrieval

The system issues a request to fetch the data from the database. The system reads one record at a time, and performs the following processing for each record:

- Attempts to fetch a record from the database.

  If successful, the following processing occurs:

  - Copy the data into the business view data structures.
  - Perform Event Rules: Grid Record is Fetched.
- If the application developer has not chosen to suppress the writing of this grid record, the following occurs:
  - Copy the business view data into the grid data structures.
  - Perform Event Rules: Write Grid Line - Before.
  - Add the row to the grid. The row now exists in the grid control.
  - Perform Event Rules: Write Grid Line - After.
  - Clear the grid data structures for reading the next record.
  - Remove the suppress grid line flag.

The previous processing occurs for each record that is read from the database. After all records are read, the following processing occurs:

- Perform Event Rules: Last Grid Record Has Been Read.

## Closing Form

- Perform Event Rules: End Dialog.
- Terminate Error Handling.
- Terminate Thread Handling.
- Terminate Helps.
- Free all structures for form, including structures for business view columns, form controls, grid columns, and event rules.
- Destroy the window.

## Push Buttons

Subforms do not have a toolbar; instead, users must add push buttons. The push buttons behave the same as the standard push buttons for other form types. The following section discusses the push buttons available on the different types of Subforms. The system does not automatically place any of these push buttons on the forms. The user must place any or all of the buttons that they need for the application.

If a Subform is on a Power Browse form, the following push buttons are available to that form:

- Select
- Find

If a Subform is on a Power Edit form, the following push buttons are available to that form:

- Clear
- Find
- Delete
- Save

### Select

Select is a standard push button that is available on Subforms placed on Power Edit forms.

- Load the mapping link value from the Subform parent.
- Perform Event Rules: Button Clicked.
- Perform Event Rules: Post Button Clicked.

### Clear

Clear is a standard push button that that is available on Subforms placed on Power Edit forms. It clears the form.

- Perform Event Rules: Button Clicked.
- Clear all controls on the Subform.
- Clear errors on the Subform.
- Perform Event Rules: Post Button Clicked.

### Find

Find is a standard push button that is available on all Subforms. When the user clicks it, it is the signal to the runtime engine to call the database and reload the form based on the information in the filter fields.

- If there is no grid on the Subform, the Find button fetches one database record and writes the values to the form controls.
    - Perform Event Rules: Button Clicked.
    - Fetch record form the Subform business view and update the business controls.
    - If the record is fetched, set the form mode to Update, and Update the business view controls.
    - Perform Event Rules: Post Button Clicked.
- If there is a grid on the Subform, the Find button fetches all database records and writes the values to the grid.
    - Perform Event Rules: Button Clicked.
    - Begin Data Selection and Sequencing
    - If record is fetched, set the form mode to update; otherwise, set the form module to Add mode.
    - If the Subform is on a Power Edit form, add an entry row.
    - Perform Event Rules: Post Button Clicked.

### Delete

Delete is a standard push button that is available on all Subforms placed on Power Edit forms. It deletes a record in the grid from the database.

- Perform Event Rules: Button Clicked.
- Do the following for each selected grid row that you can delete:
    - Copy the grid row data into the Business View structures.
    - Remove Suppress Delete flag.
    - Perform Event Rules: Delete Grid Rec Verify – Before.

- If the Suppress Delete flag is not set, do the following:
  - Display Delete Confirmation dialog. If the user clicks NO or CANCEL the system skips the rest of this processing.
  - Perform Event Rules: Delete Grid Rec Verify – After.
  - Perform Event Rules: Delete Grid Rec from DB – Before.
  - Delete the record in the Business View from the database.
  - Delete the grid row from the grid control.
  - Perform Event Rules: Delete Grid Rec from DB – After.
  - Perform Event Rules: All Grid Recs Deleted.
  - Perform Event Rules: Post Button Clicked.

### Save

Save is a standard push button that is available on all Subforms placed on Power Edit forms. It validates the information on the Subform and updates or adds to the database.

---

**Note**

If the Subform contains a grid control, the Save button processing behaves similar to the OK button on a Headerless/Detail form. If the Subform does not contain a grid, the Save button processing behaves similar to the OK button on a Fix/Inspect form.

---

#### Save Processing for a Subform with a Grid

- If any errors or warnings appear on the form, stop OK processing
- Perform Event Rules: Button Clicked.
- For each control on the form, do the following:

  If the current control is a form control and it has not passed validation, do the following:

  - Perform Event Rules: Control is Exited.
  - Perform Event Rules: Control is Exited. and Changed - Inline.
  - Perform Event Rules: Control is Exited. and Changed - Async
  - Perform Data Dictionary validation.
- If any errors appear on the form, stop OK processing
- If the Subform does not have a business view, do the following:
  - Perform Event Rules: Post Button Clicked.
  - Stop Processing.
- If the form is in Add mode, do the following:
  - Perform Event Rules: Add Record to Database - Before.

- If the form is in Update mode, do the following:
  - Perform Event Rules: Update Record to Database - Before

  Perform Event Rules: Post Button Clicked.

- If form is in Add Mode, do the following:

  If form was called in Copy mode or the flag “End Form On Add” is checked, do the following:

  - Begin Closing Form.

  Else

  - Begin Clearing Dialog.
- If the form is in Update mode, do the following:

  If no errors occurred while attempting to update or add to the database, do the following:

  - Begin Closing Form.
- If the Subform is in Add mode, Begin Clearing Dialog.

### Save Processing for a Subform without a Grid

- If any errors or warnings appear on the form, stop OK processing
- Perform Event Rules: Button Clicked.
- For each control on the form, do the following:

  If the current control is a form control and it has not passed validation, do the following:

  - Perform Event Rules: Control is Exited.
  - Perform Event Rules: Control is Exited. and Changed - Inline.
  - Perform Event Rules: Control is Exited. and Changed - Async
  - Perform Data Dictionary validation.
- If any errors appear on the form, stop OK processing, if the “No Update on Subform Business View” option is turned on.
- If the form is in Add mode, do the following:
  - Perform Event Rules: Add Record to Database - Before.
  - Add the record to the database.
  - Perform Event Rules: Add Record to Database - After.
- If the form is in Update mode, do the following:
  - Perform Event Rules: Update Record to Database – Before.
  - Update the record in the database.
  - Perform Event Rules: Add Record to Database - After.

- Perform Event Rules: Post Button Clicked.
- If form is in Add Mode, clear the form.

### User Defined Items

User-defined items are nonstandard items that you can add to Subforms to perform specialized processing that is not handled by the standard items.

- If the Update Mapping Link option is turned on, the system updates the mapping link items with the current values on the parent form.
- Perform Event Rules: Button Clicked.
- Perform Event Rules: Post Button Clicked.

# Process Flow for Message Forms

The message form type is a form that appears as a secondary window to inform the user of something or to ask a question. It parallels the behavior of a Windows message box. The form does not have a toolbar or a status bar and can only contain static text and buttons. It is the only form type for which standard buttons can appear on the form instead of the toolbar.

Message forms allow only limited use of PO values and business functions. Therefore, do not use this form type for complex logic.

## Dialog Initialization

- Initialize Form Controls.
- Initialize Static Text.
- Initialize Helps.
- Initialize Event Rules Structures.
- Perform Event Rules: `Dialog is Initialized`

## Closing Form

- Terminate Helps.
- Free all structures for Controls and Event Rules.
- Destroy the window.

## Buttons

The form includes an OK button by default. Other options include Cancel, Yes, and No. All of these buttons cause the dialog to close, and result in no other default processing.

## PeopleSoft Standards

Message forms can have only static text and buttons. Event rules are limited to `Dialog is Initialized` and any button events. This event rule cannot contain any Form Interconnection calls.

# Process Flow for Edit Controls

An edit control is a text field on a form. All form types can contain edit controls except message forms. Two types of edit controls are available. The first type is commonly referred to as a database field. It is associated with an item in the business view and through that connection to a specific data dictionary item. Database fields represent a field in a database record. The second type of edit control is commonly referred to as a data dictionary field, and it also has a connection to a specific data dictionary item.

Within the realm of database fields an additional distinction between filter fields and nonfilter fields. Database fields are nonfilter fields by default. Filter fields are used to alter the selection criteria of a database fetch. A filter can have a comparison type of equal, not equal, less than, less than or equal to, greater than, or greater than or equal to. A filter can be marked such that a wildcard (*) displays when the filter is not included in the selection.

The storage for the value of the edit control is based on the type of its associated data dictionary item (such as numeric, string, character). An edit control is affected by the properties of the associated data dictionary item. For example, if an edit control is associated with a data dictionary item of type string with a length of thirty, the edit control will not allow more than thirty characters to be typed into the field.

## Properties and Options

The following properties and options are available for edit controls. These properties are available on all form types and for both database fields and data dictionary fields. Only properties that affect processing are discussed here. The properties that are exclusively interface-related are either not discussed, or are only discussed as they relate to control processing.

| | |
|---|---|
| **Disable** | Controls with this property appear gray and cannot be changed by the user. You can use event rules to change the value of disabled fields. When a control is disabled, the associated text (static text to the left) is also disabled. Controls do not have this property by default. |
| **Visible** | Controls with this property can be seen. Without this property, the control cannot be seen. You can use event rules to change the value of hidden controls. |
| **Do not clear after add** | This option applies only when a form is in add mode and is being cleared. Controls with this option retain their value. Controls do not have this option by default. |
| **Required entry field** | This option requires that a value be entered in the control before the record is saved (OK processing). This option does not accept null or blank. Controls do not have this option by default. |

| | |
|---|---|
| **Default cursor on add/update mode** | This option specifies where the cursor is placed in add/update mode. You can assign this option to only one control on a form. Otherwise, the system cannot determine where to place the cursor. Controls do not have this option by default. |
| **No display if currency is Off** | This option causes the control to be hidden when Multicurrency Conversion is off. Controls do not have this option by default. |

# Control is Entered

- Perform event rules: `Control is Entered`

On a Windows client, when a user presses the tab key or otherwise causes focus to move from one control to another, the control that receives focus actually receives the Control is Entered message before the control that loses focus receives the Control is Exited message. This is due to the sequence of Windows messages and cannot be altered.

# Control is Exited

- If the focus is going to another window, stop processing.

  The focus returns to this control when this window again receives focus.

- Copy the text from the screen to the internal storage.

  Conversion from string to the appropriate type happens during this step.

- If this control is marked as a required entry field and does not contain a value, then set the field in error.
- Perform event rules: `Control is Exited`.
- If this is the first time this control has been exited or if this control has changed in value since the last time the control was exited, then do the following:
  - `Begin Control is Exited and Changed (Inline)`

## Control is Exited and Changed (Inline portion)

- Perform event rules: `Control is Exited and Changed Inline`
- If the form is not closing, do the following:
  - Attempt to execute `Control is Exited and Changed (Asynchronous)` on a thread.
- If the form is closing or if the attempt to execute asynchronously was unsuccessful, do the following:
  - Begin `Control is Excited and Changed (Asynchronous)` as an inline function.

### Control is Exited and Changed (Asynchronous portion)

- Perform event rules: `Control is Exited and Changed Asynch`
- If no errors were set during event rules processing, do the following:
  - If this is not a filter field, or if it is an equal filter field (validation takes place only on filter fields with a comparison type of equal), perform Data Dictionary Validation.
  - If an associated description field for this control exists, populate it (if errors were present, this clears the associated description).
  - If no errors occurred during validation, then copy the value to the control.

## PeopleSoft Standards

All filter fields are marked with Display Wildcard. Only index fields are marked as filter fields (for performance considerations). If reasonable, an event rule should be placed on `Control is Exited and Changed Asynch` instead of on one of the inline events (for performance considerations).

# Process Flow for Grid Controls

A grid control is similar to a spreadsheet on a form. Form types that can contain grid controls are as follows:

- Find/browse
- Search/select
- Header detail
- Headerless detail

Grid controls contain columns. The columns are specified at design time and are either of the following two types:

- Database columns
- Data dictionary columns

A database column is associated with an item in the business view and through that connection to a dictionary item. Database columns represent a field in a database record.

Although only one type of column is referred to as a data dictionary column, both types have a connection to a specific data dictionary item. The difference is that a database column has the additional connection to a business view field.

A grid can either be a browse grid or an update grid. You can use a browse grid for viewing only, and you cannot select individual cells. The find/browse form and the search/select form have browse grids.

You can use an update grid to add or update records. Cells in an update grid can be selected individually. The header detail form and the headerless detail form have update grids.

Grid controls can also have a query by example (QBE) line. The QBE columns have a one-to-one correspondence with the grid columns. You use a QBE value to change the selection criteria of a database fetch. Only database columns allow entry in the QBE columns because the purpose of the QBE is to affect the selection and only database columns are in the business view. A QBE column can have one of the following comparison types:

- Equal
- Not equal
- Less than
- Less than or equal to
- Greater than
- Greater than or equal to

The comparison type is *equal* unless you specify otherwise. You can specify the comparison type in the QBE column or by using system functions. You can use wildcards (* or %) for an inexact search on a string field.

The values contained on a grid row act as a logical unit. You must validate a grid row prior to accepting a record, but validating the individual columns is not required. Edit control validation and grid row validation are parallel, but edit control validation and grid column validation are not parallel. The *Column is Exited* events are executed only if the user physically exits the cell. The Data Dictionary Validation for a cell executes when the cell is exited after a change or the first time that the row is exited after it has been added. If you change a cell programmatically, then the *Row is Exited* events execute prior to accepting a record, but the column events and validation do not execute unless focus is physically set on the column.

The vendor spreadsheet stores the grid cell values as a tab delimited string (one per row). The values can be retrieved on a cell-by-cell basis or on a row-by-row basis. Internal storage for the grid columns also exists in the interactive engine. The actual storage for the grid column value is based on the type of the associated dictionary item (for example, math numeric, string, character), and it is distinct from the screen representation of the value. Only one row of data can be acted upon at any given time. Each event executes in the context of a specific row.

A grid column is affected by the properties of the associated dictionary item. For example, if a grid column is associated with a dictionary item of type string with a length of 30, that grid column will not allow more than 30 characters to be typed into the cell.

# Properties and Options

The following properties and options are available for grid controls. Except where noted, these properties are available on all grids. Only those properties that affect processing are listed here. Those properties that are exclusively related to the interface are either not discussed, or are discussed only as they relate to grid processing.

**Disable** Grids with this property appear gray and cannot be changed by the user. The application developer can change the value of disabled fields through event rules (ER). Grids do not have this property by default.

**Visible** Grids with this property can be seen. Without this property, the grid cannot be seen. The application developer can cause hidden grids to change value through ER.

| | |
|---|---|
| **Hide Query By Example** | Grids with this property do not have a QBE line. QBE is neither available to the user nor the ER. Browse grids do not have this property by default. Update grids have this property by default. |
| **Update Mode** | This property is only available on update grids, but does not have any effect on the grid during runtime. |
| **Multiple Select** | This property allows for multiple grid rows to be selected at once. This can affect ER execution and deleting. Grids do not have this option by default. |
| **Automatically Find On Entry** | This option determines whether grid records will be fetched when the form is opened. Grids without this option will open with no grid rows. Grids do not have this property by default. |
| **Auto Find On Changes** | This option determines whether grid records will be fetched after a child form that has changed records closes. This option should be used only on forms that have no modeless form interconnects. Grids do not have this property by default. |
| **No Adds on Update Grid** | This property applies to updates only. It determines whether an entry row appears in the grid. Without an entry row, records cannot be added. With this property turned on, only existing records can be altered. (Records can be updated only). Grids do not have this option by default. |
| **Disable Page-At-A-Time Processing** | This option causes all available grid records to be fetched when Find is pressed. Without this option, only the first page of grid records is fetched until the user scrolls down to see additional records. Each time that more records are requested, a page of data is returned. This provides a substantial performance benefit for large files and it is not recommended that this option be checked without careful consideration. Grids do not have this option by default. |
| **Clear Grid After Add** | There is currently no reference to this field during runtime. The presence or absence of this flag has no effect on the grid. |
| **Refresh Grid After Update** | Currently, no reference to this field exists during runtime. The presence or absence of this flag has no effect on the grid. |
| **Process All Rows In Grid** | This option causes each row in the grid to perform the three Row is Exited events (*Row is Exited*, *Row is Exited and Changed* (Asynchronous portion), and *Row is Exited Validation*) on all of the grid rows at least once before updating or adding the database records. |

# Set Focus on Grid

Perform event rules: *Set Focus on Grid*

# Kill Focus on Grid

Perform event rules: *Kill Focus on Grid*

## Row is Entered

- Update the status bar with the current row number.
- If this row is not the entry row (last row on update grids), do the following:
  - Perform event rules: *Row is entered*

## Row is Exited

- If the form is not losing focus:
  - Perform event rules: *Row is exited*
  - If this is an update grid and the row has been changed since the last time the row was exited, do the following:

    Perform event rules: *Row is Exited and changed - Inline*

    If you are not leaving the grid (exiting one row, entering another), do the following:

    Attempt to execute *Row is Exited and Changed* (Asynchronous portion) on a thread.
  - If you are leaving the grid or the attempt to execute asynchronously was unsuccessful, do the following:

    Begin *Row is Exited and Changed* (Asynchronous portion) as an inline function.

## Row is Exited and Changed (Asynchronous portion)

- Set a bitmap on the grid row header to indicate that this row is being processed.
- Perform event rules: *Row is Exited and Changed - Asynch*
- If this is the first time that this row has been exited since it was added, do the following:
  - Begin *Row is Exited Validation*

## Row is Exited Validation

For each grid cell in this row that has not already performed data dictionary validation, do the following:

- If this database item is also in the header portion of the form, the grid column is populated from that value, so skip this validation and go to the next grid cell.

  **Note**

  Filter fields are populated in the grid column only if they are equal filters.

- If the text contained in the cell can be stored (for example, no alphas in numerics and no out-of-range data parameters):

- Begin *Data Dictionary Validation* for this cell.

  If an associated description column for this cell exists, populate it with the information returned from *Data Dictionary Validation*.

## Cell is Exited after a Change

On update grids when the contents of a cell have changed (via user keying or visual assist) and the cell has been exited, do the following:

- Clear any errors set on this cell before calls to *Row is Exited Validation* and *Column is Exited* events.
- If the text contained in the cell can be stored (no alphas in numerics, no our of range date parameters), do the following:
  - Perform *Data Dictionary Validation*
  - If an associated description column exists for this cell, populate it with the information returned from *Data Dictionary Validation*.

## Double-Click a Grid Row

On browse grids, double-clicking on the grid row causes the Select button to be pressed.

## Key Pressed

On update grids, when a key is pressed on the entry row (last row in the grid), a new row is added to the grid.

## PeopleSoft Standards

The event *Row is Exited and Changed - Asynch* is equivalent to validating the contents of the row. It is often used for the Edit Line master business function. For performance reasons, ER should be placed on *Row is Exited and Changed - Asynch* instead of on one of the inline events.

# 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

### Prerequisite

- 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.

# 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.

PeopleSoft 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 PeopleSoft 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. PeopleSoft 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:

- Create the base menu item and populate it with exits or other menus.
- Select a bitmap for the exit if you are going to show it on the toolbar.
- 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 PeopleSoft 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.

### Prerequisite

- 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>)

# EnterpriseOne Design Standards for Form Types

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.

### Prerequisite

- 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.

# EnterpriseOne PeopleBooks Glossary

| | |
|---|---|
| **“as of” processing** | A process that is run at a specific point in time to summarize item transactions. |
| **52 period accounting** | A method of accounting that uses each week as a separate accounting period. |
| **account site** | In the invoice process, the address to which invoices are mailed. Invoices can go to a different location or account site from the statement. |
| **active window** | The window that contains the document or display that will be affected by current cursor movements, commands, and data entry in environments that are capable of displaying multiple on-screen windows. |
| **ActiveX** | A technology and set of programming tools developed by Microsoft Corporation that enable software components written in different languages to interact with each another in a network environment or on a web page. The technology, based on object linking and embedding, enables Java applet-style functionality for Web browsers as well as other applications (Java is limited to Web browsers at this time). The ActiveX equivalent of a Java applet is an ActiveX control. These controls bring computational, communications, and data manipulation power to programs that can “contain” them—for example, certain Web browsers, Microsoft Office programs, and anything developed with Visual Basic or Visual C++. |
| **activity** | In Advanced Cost Accounting, an aggregation of actions performed within an organization that is used in activity-based costing. |
| **activity driver** | A measure of the frequency and intensity of the demands that are placed on activities by cost objects. An activity driver is used to assign costs to cost objects. It represents a line item on the bill of activities for a product or customer. An example is the number of part numbers, which is used to measure the consumption of material-related activities by each product, material type, or component. The number of customer orders measures the consumption of order-entry activities by each customer. Sometimes an activity driver is used as an indicator of the output of an activity, such as the number of purchase orders that are prepared by the purchasing activity. See also cost object. |
| **activity rule** | The criteria by which an object progresses from a given point to the next in a flow. |
| **actual cost** | Actual costing uses predetermined cost components, but the costs are accumulated at the time that they occur throughout the production process. |
| **adapter** | A component that connects two devices or systems, physically or electronically, and enables them to work together. |
| **add mode** | The condition of a form where a user can enter data into it. |
| **advanced interactive executive** | An open IBM operating system that is based on UNIX. |
| **agent** | A program that searches through archives or other repositories of information on a topic that is specified by the user. |

| | |
|---|---|
| **aging** | A classification of accounts by the time elapsed since the billing date or due date. Aging is divided into schedules or accounting periods, such as 0-30 days, 31-60 days, and so on. |
| **aging schedule** | A schedule that is used to determine whether a payment is delinquent and the number of days which the payment is delinquent. |
| **allegato IVA clienti** | In Italy, the term for the A/R Annual VAT report. |
| **allegato IVA fornitori** | In Italy, the term for the A/P Annual VAT report. |
| **application layer** | The seventh layer of the Open Systems Interconnection Reference Model, which defines standards for interaction at the user or application program level. |
| **application programming interface (API)** | A set of routines that is used by an application program to direct the performance of procedures by the computer's operating system. |
| **AS/400 Common** | A data source that resides on an AS/400 and holds data that is common to the co-existent library, allowing PeopleSoft EnterpriseOne to share information with PeopleSoft World. |
| **assembly inclusion rule** | A logic statement that specifies the conditions for using a part, adjusting the price or cost, performing a calculation, or using a routing operation for configured items. |
| **audit trail** | The detailed, verifiable history of a processed transaction. The history consists of the original documents, transaction entries, and posting of records and usually concludes with a report. |
| **automatic return** | A feature that allows a user to move to the next entry line in a detail area or to the first cell in the next row in several applications. |
| **availability** | The expression of the inventory amount that can be used for sales orders or manufacturing orders. |
| **available inventory** | The quantity of product that can be promised for sale or transfer at a particular time, considering current on-hand quantities, replenishments in process, and anticipated demand. |
| **back office** | The set of enterprise software applications that supports the internal business functions of a company. |
| **backhaul** | The return trip of a vehicle after delivering a load to a specified destination. The vehicle can be empty or the backhaul can produce less revenue than the original trip. For example, the state of Florida is considered a backhaul for many other states—that is, many trucking companies ship products into the state of Florida, but most of them cannot fill a load coming out of Florida or they charge less. Hence, trucks coming out of Florida are either empty or produce less revenue than the original trip. |
| **balance forward** | The cumulative total of inventory transactions that is used in the Running Balance program. The system does not store this total. You must run this program each time that you want to review the cumulative inventory transactions total. |
| **balance forward receipt application method** | A receipt application method in which the receipt is applied to the oldest or newest invoices in chronological order according to the net due date. |

| | |
|---|---|
| **bank tape (lock box) processing** | The receipt of payments directly from a customer's bank via customer tapes for automatic receipt application. |
| **base location** | [In package management] The topmost location that is displayed when a user launches the Machine Identification application. |
| **basket discount** | A reduction in price that applies to a group or "basket" of products within a sales order. |
| **basket repricing** | A rule that specifies how to calculate and display discounts for a group of products on a sales order. The system can calculate and display the discount as a separate sales order detail line, or it can discount the price of each item on a line-by-line basis within the sales order. |
| **batch job** | A job submitted to a system and processed as a single unit with no user interaction. |
| **batch override** | An instruction that causes a batch process to produce output other than what it normally would produce for the current execution only. |
| **batch process** | A type of process that runs to completion without user intervention after it has been started. |
| **batch program** | A program that executes without interacting with the user. |
| **batch version** | A version of a report or application that includes a set of user-defined specifications, which control how a batch process runs. |
| **batch/lot tracking** | The act of identifying where a component from a specific lot is used in the production of goods. |
| **batch/mix** | A manufacturing process that primarily schedules short production runs of products. |
| **batch-of-one processing** | A transaction method that allows a client application to perform work on a client workstation, and then submit the work all at once to a server application for further processing. As a batch process is running on the server, the client application can continue performing other tasks. See also direct connect, store-and-forward. |
| **binary large object (BLOB)** | A collection of binary data stored as a single entity in a [file]. |
| **binder clip** | See paper clip. |
| **black products** | Products that are derived from the low or heavy end of the distillation process—for example, diesel oils and fuel oils. See also white products. |
| **blend note** | Document that authorizes a blending activity, and describes both the ingredients for the blend and the blending steps that occur. |
| **blend off** | Reworking off-specification material by introducing a small percentage back into another run of the same product. |
| **blind execution** | The mode of execution of a program that does not require the user to review or change the processing options set for the program, and does not require user intervention after the program has been launched. |

| | |
|---|---|
| **boleto** | In Brazil, the document requesting payment by a supplier or a bank on behalf of a supplier. |
| **bolla doganale** | VAT-Only Vouchers for Customs. In Italy, a document issued by the customs authority to charge VAT and duties on extra-EU purchasing. |
| **bookmark** | A shortcut to a location in a document or a specific place in an application or application suite. |
| **bordero & cheque** | In Brazil, bank payment reports. |
| **broker** | A program that acts as an intermediary between clients and servers to coordinate and manage requests. |
| **BTL91** | In the Netherlands, the ABN/AMRO electronic banking file format that enables batches with foreign automatic payment instructions to be delivered. |
| **budgeted volume** | A statement of planned volumes (capacity utilization) upon which budgets for the period have been set. |
| **bunkering** | A rate per ton or a sum of money that is charged for placing fuel on board; can also mean the operation itself. |
| **business function** | An encapsulated set of business rules and logic that can normally be re-used by multiple applications. Business functions can execute a transaction or a subset of a transaction (check inventory, issue work orders, and so on). Business functions also contain the APIs that allow them to be called from a form, a database trigger, or a non-EnterpriseOne application. Business functions can be combined with other business functions, forms, event rules, and other components to make up an application. Business functions can be created through event rules or third-generation languages, such as C. Examples of business functions include Credit Check and Item Availability. |
| **business function event rule** | Encapsulated, reusable business logic that is created by using through event rules rather than C programming. Contrast with embedded event rule. See also event rule. |
| **business object library** | [In interoperability] The repository that stores EnterpriseOne business objects, which consist of Java or CORBA objects. |
| **business unit** | A financial entity that is used to track the costs, revenue, or both, of an organization. A business unit can also be defined as a branch/plant in which distribution and manufacturing activities occur. Additionally, in manufacturing setup, work centers and production lines must be defined as business units; but these business unit types do not have profit/loss capability. |
| **business view** | Used by EnterpriseOne applications to access data from database tables. A business view is a means for selecting specific columns from one or more tables with data that will be used in an application or report. It does not select specific rows and does not contain any physical data. It is strictly a view through which data can be handled. |
| **business view design aid (BDA)** | An EnterpriseOne GUI tool for creating, modifying, copying, and printing business views. The tool uses a graphical user interface. |

| | |
|---|---|
| **buy-back crude** | In foreign producing oil countries, that portion of the host government's share of "participation crude" which it permits the company holding a concession to "buy back." |
| **CAB** | In Italy, the bank branch code or branch ID. A five-digit number that identifies any agency of a specific bank company in Italy. |
| **cadastro de pessoas físicas** | Cadastro de pessoas físicas. In Brazil, the federal tax ID for a person. |
| **category code** | A code that identifies a collection of objects sharing at least one common attribute. |
| **central object** | A software component that resides on a central server. |
| **central objects merge** | A process that blends a customer's modifications with the objects in a current release with objects in a new release. |
| **central server** | A computer that has been designated to contain the originally installed version of the software (central objects) for deployment to client computers. |
| **certificate input** | See direct input. |
| **certificate of analysis (COA)** | A document that is a record of all of the testing which has been performed against an item, lot, or both, plus the test results for that item and lot. |
| **change management** | [In software development] A process that aids in controlling and tracking the evolution of software components. |
| **change order** | In PeopleSoft, an addendum to the original purchase order that reflects changes in quantities, dates, or specifications in subcontract-based purchasing. A change order is typically accompanied by a formal notification. |
| **chargeback** | A receipt application method that generates an invoice for a disputed amount or for the difference of an unpaid receipt. |
| **chart** | EnterpriseOne term for tables of information that appear on forms in the software. See forms. |
| **check-in location** | The directory structure location for the package and its set of replicated objects. This location is usually \\deploymentserver\release\path_code\ package\packagename. The subdirectories under this path are where the central C components (source, include, object, library, and DLL file) for business functions are stored. |
| **checksum value** | A computed value that depends on the contents of a block of data, and that is transmitted or stored with the data to detect whether errors have occurred in the transmission or storage. |
| **class** | [In object-oriented programming] A category of objects that share the same characteristics. |
| **clean cargo** | Term that refers to cargoes of gasoline and other refined products. See also dirty cargo. |
| **client access** | The ability to access data on a server from a client machine. |
| **client machine** | Any machine that is connected to a network and that exchanges data with a server. |

| | |
|---|---|
| **client workstation** | A network computer that runs user application software and is able to request data from a server. |
| **ClieOp03** | In the Netherlands, the euro-compliant uniform electronic banking file format that enables batches with domestic automatic direct debit instructions and batches with domestic payment instructions to be delivered. |
| **ClieOp2** | In the Netherlands, the uniform electronic banking file format that enables batches with domestic automatic direct debit instructions and batches with domestic payment instructions to be delivered. |
| **cluster** | Two or more computers that are grouped together in such a way that they behave like a single computer. |
| **co-existence** | A condition where two or more applications or application suites access one or more of the same database tables within the same enterprise. |
| **cold test** | The temperature at which oil becomes solid. Generally considered to be 5 degrees F lower than the pour point. |
| **commitment** | The number of items that are reserved to fill demand. |
| **common object request broker architecture** | An object request broker standard that is endorsed by the Object Management Group. |
| **compa-ratio** | An employee's salary divided by the midpoint amount for the employee's pay grade. |
| **component changeout** | See component swap. |
| **component object model (COM)** | A specification developed by Microsoft for building software components that can be assembled into programs or add functionality to existing programs running on Microsoft Windows platforms. COM components can be written in a variety of languages, although most are written in C++, and can be unplugged from a program at runtime without having to recompile the program. |
| **component swap** | In Equipment/Plant Management, the substitution of an operable component for one that requires maintenance. Typically, you swap components to minimize equipment downtime while servicing one of the components. A component swap can also mean the substitution of one parent or component item for another in its associated bill of material. |
| **conference room pilot environment** | An EnterpriseOne environment that is used as a staging environment for production data, which includes constants and masters tables such as company constants, fiscal date patterns, and item master. Use this environment along with the test environment to verify that your configuration works before you release changes to end-users. |
| **configurable network computing (CNC)** | An application architecture that allows interactive and batch applications that are composed of a single code base to run across a TCP/IP network of multiple server platforms and SQL databases. The applications consist of re-usable business functions and associated data that can be configured across the network dynamically. The overall objective for businesses is to provide a future-proof environment that enables them to change organizational structures, business processes, and technologies independently of each other. |

| | |
|---|---|
| **configurable processing engine** | Handles all "batch" processes, including reporting, Electronic Data Exchange (EDIt) transactions, and data duplication and transformation (for data warehousing). This ability does not mean that it exists only on the server; it can be configured to run on desktop machines (Windows 95 and NT Workstation) as well. |
| **configuration management** | A rules-based method of ordering assemble-to-order or make-to-order products in which characteristics of the product are defined as part of the Sales Order Entry process. Characteristics are edited by using Boolean logic, and then translated into the components and routing steps that are required to produce the product. The resulting configuration is also priced and costed, based on the defined characteristics. |
| **configured item segment** | A characteristic of a configured item that is defined during sales order entry. For example, a customer might specify a type of computer hard drive by stating the number of megabytes of the hard drive, rather than a part number. |
| **consuming location** | The point in the manufacturing routing where a component or subassembly is used in the production process. In kanban processing, the location where the kanban container materials are used in the manufacturing process and the kanban is checked out for replenishment. |
| **contra/clearing account** | A G/L account used by the system to offset (balance) journal entries. For example, you can use a contra/clearing account to balance the entries created by allocations. |
| **contribution to profit** | Selling price of an item minus its variable costs. |
| **control table** | A table that controls the program flow or plays a major part in program control. |
| **control table workbench** | During the Installation Workbench process, Control Table Workbench runs the batch applications for the planned merges that update the data dictionary, user defined codes, menus, and user overrides tables. |
| **control tables merge** | A process that blends a customer's modifications to the control tables with the data that accompanies a new release. |
| **corrective work order** | A work order that is used to formally request unscheduled maintenance and communicate all of the details pertaining to the requested maintenance task. |
| **corrective work order** | A work order that is used to formally request unscheduled maintenance and communicate all of the details pertaining to the requested maintenance task. |
| **cost assignment** | Allocating resources to activities or cost objects. |
| **cost component** | An element of an item's cost—for example, material, labor, or overhead. |
| **cost object** | Any customer, product, service, contract, project, or other work unit for which you need a separate cost measurement. |
| **cost rollup** | A simulated scenario in which work center rates, material costs, and labor costs are used to determine the total cost of an item. |
| **costing elements** | The individual classes of added value or conversion costs. These elements are typically materials, such as raw and packaging; labor and machine costs; and overhead, such as fixed and variable. Each corporation defines the necessary detail of product costs by defining and tracking cost categories and subcategories. |

| | |
|---|---|
| **credit memo** | A negative amount that is used to correct a customer's statement when he or she is overcharged. |
| **credit notice** | The physical document that is used to communicate the circumstances and value of a credit order. |
| **credit order** | A credit order is used to reflect products or equipment that is received or returned so that it can be viewed as a sales order with negative amounts. Credit orders usually add the product back into inventory. This process is linked with delivery confirmation. |
| **cross segment edit** | A logic statement that establishes the relationship between configured item segments. Cross segment edits are used to prevent ordering of configurations that cannot be produced. |
| **crude oil assay** | A procedure for determining the distillation curve and quality characteristics of a crude oil. |
| **cumulative update** | A version of software that includes fixes and enhancements that have been made since the last release or update. |
| **currency relationships** | When converting amounts from one currency to another, the currency relationship defines the from currency and the to currency in PeopleSoft software. For example, to convert amounts from German marks to the euro, you first define a currency relationship between those two currencies. |
| **currency restatement** | The process of converting amounts from one currency into another currency, generally for reporting purposes. It can be used, for example, when many currencies must be restated into a single currency for consolidated reporting. |
| **current cost** | The cost that is associated with an item at the time a parts list and routing are attached to a work order or rate schedule. Current cost is based on the latest bill of material and routing for the item. |
| **customer pricing rules** | In Procurement, the inventory pricing rules that are assigned to a supplier.<br>In Sales, inventory pricing rules that are assigned to a customer. |
| **D.A.S. 2 Reporting (DAS 2 or DADS 1)** | In France, the name of the official form on which a business must declare fees and other forms of remuneration that were paid during the fiscal year. |
| **data dictionary** | A dynamic repository that is used for storing and managing a specific set of data item definitions and specifications. |
| **data source workbench** | During the Installation Workbench process, Data Source Workbench copies all of the data sources that are defined in the installation plan from the Data Source Master and Table and Data Source Sizing tables in the Planner data source to the System - release number data source. It also updates the Data Source Plan detail record to reflect completion. |
| **data structure** | A description of the format of records in a database such as the number of fields, valid data types, and so on. |
| **data types** | Supplemental information that is attached to a company or business unit. Narrative type contains free-form text. Code type contains dates, amounts, and so on. |

| | |
|---|---|
| **datagram** | A self-contained packet of information that is forwarded by routers, based on their address and the routing table information. |
| **date pattern** | A period of time that is set for each period in standard and 52-period accounting and forecasting. |
| **DCE** | See distributed computing environment. |
| **DEB** | See déclaration d'echange de biens. |
| **debit memo** | In Accounts Payable, a voucher that is entered with a negative amount. Enter this type of voucher when a supplier sends you a credit so that you can apply the amount to open vouchers when you issue payment to the supplier. |
| **debit memo** | A form that is issued by a customer, requesting an adjustment of the amount, which is owed to the supplier. |
| **debit statement** | A list of debit balances. |
| **de-blend** | When blend off does not result in a product that is acceptable to customers. The further processing of product to adjust specific physical and chemical properties to within specification ranges. See also blend off. |
| **déclaration d'echange de biens (DEB)** | The French term that is used for the Intrastat report. |
| **delayed billing** | The invoicing process is delayed until the end of a designated period. |
| **delta load** | A batch process that is used to compare and update records between specified environments. |
| **denominated-in currency** | The company currency in which financial reports are based. |
| **deployment server** | A server that is used to install, maintain, and distribute software to one or more enterprise servers and client workstations. |
| **detail** | The specific information that makes up a record or transaction. Contrast with summary. |
| **detail information** | Information that primarily relates to individual lines in a sales or purchase order. |
| **direct connect** | A transaction method in which a client application communicates interactively and directly with a server application. See also batch-of-one immediate, store-and-forward. |
| **direct input** | The system calculates the net units when you enter gross volume, temperature, and gravity or density. This data is generally entered during product receiving from the certificate that is prepared by an independent inspector. |
| **direct ship orders** | A purchase order that is issued to a third-party supplier who designates the destination as the customer. A direct ship sales order is also created for the customer. Direct ship orders occur when a product is not available from a company-owned or company-operated source, so the system creates an order to ship the product from a third-party source directly to the customer. Sometimes referred to as a drop ship or third-party supply. |
| **direct usage** | Consumption of resources that are attributable to specific production runs because the resources were directly issued to the schedule/order. |

| | |
|---|---|
| **director** | An EnterpriseOne user interface that guides a user interactively through an EnterpriseOne process. |
| **dirty cargo** | Term that refers to crude oil cargoes or other non-refined petroleum cargoes. See also clean cargo. |
| **dispatch planning** | Efficient planning and scheduling of product deliveries. Considerations include:<br>Dispatch groups<br>Scheduled delivery date<br>Scheduled delivery time<br>Preferred delivery date<br>Preferred delivery time<br>Average delivery time for that geographical location<br>Available resources<br>Special equipment requirements at the product's source or destination. |
| **displacement days** | The number of days that are calculated from today's date by which you group vouchers for payment. For example, if today's date is March 10 and you specify three displacement days, the system includes vouchers with a due date through March 13 in the payment group. Contrast with pay-through date. |
| **display sequence** | A number that the system uses to re-order a group of records on the form. |
| **distributed computing environment (DCE)** | A set of integrated software services that allows software which is running on multiple computers to perform seamless and transparently to the end-users. DCE provides security, directory, time, remote procedure calls, and files across computers running on a network. |
| **distributed data processing** | Processing in which some of the functions are performed across two or more linked facilities or systems. |
| **distributed database management system (DDBMS)** | A system for distributing a database and its control system across many geographically dispersed machines. |
| **do not translate (DNT)** | A type of data source that must exist on the AS/400 because of BLOB restrictions. |
| **double-byte character set (DBCS)** | A method of representing some characters by using one byte and other characters by using two bytes. Double-byte character sets are necessary to represent some characters in the Japanese, Korean, and Chinese languages. |
| **downgrade profile** | A statement of the hierarchy of allowable downgrades. Includes substitutions of items, and meeting tighter specifications for those products with wider or overlapping specification ranges. |
| **DTA** | Datenträgeraustausch. A Swiss payment format that is required by Telekurs (Payserv). |
| **dual pricing** | To provide prices for goods and services in two currencies. During the euro transition period, dual pricing between the euro and Economic and Monetary Union (EMU) member currencies is encouraged. |

| | |
|---|---|
| **dynamic link library (DLL)** | A set of program modules that are designed to be invoked from executable files when the executable files are run, without having to be linked to the executable files. They typically contain commonly used functions. |
| **dynamic partitioning** | The ability to dynamically distribute logic or data to multiple tiers in a client/server architecture. |
| **economy of scale** | A phenomenon whereby larger volumes of production reduce unit cost by distributing fixed costs over a larger quantity. Variable costs are constant; but fixed costs per unit are reduced, thereby reducing total unit cost. |
| **edit mode** | A processing mode or condition where the user can alter the information in a form. |
| **edit rule** | A method that is used for formatting user entries, validating user entries, or both, against a predefined rule or set of rules. |
| **embedded event rule** | An event rule that is specific to a particular table or application. Examples include form-to-form calls, hiding a field that is based on a processing option value, or calling a business function. Contrast with business function event rule. See also event rule. |
| **employee work center** | A central location for sending and receiving all EnterpriseOne messages (system and user-generated), regardless of the originating application or user. Each user has a mailbox that contains workflow and other messages, including Active Messages. With respect to workflow, the Message Center is MAPI compliant and supports drag-and-drop work reassignment, escalation, forward and reply, and workflow monitoring. All messages from the message center can be viewed through EnterpriseOne messages or Microsoft Exchange. |
| **Emulator** | An item of software or firmware that allows one device to imitate the functioning of another. |
| **encapsulation** | The ability to confine access to and manipulation of data within an object to the procedures that contribute to the definition of that object. |
| **engineering change order (ECO)** | A work order document that is used to implement and track changes to items and resulting assemblies. The document can include changes in design, quantity of items required, and the assembly or production process. |
| **enhanced analysis database** | A database containing a subset of operational data. The data on the enhanced analysis database performs calculations and provides summary data to speed generation of reports and query response times. This solution is appropriate when external data must be added to source data, or when historical data is necessary for trend analysis or regulatory reporting. See also duplicated database, enterprise data warehouse. |
| **enterprise server** | A computer containing programs that collectively serve the needs of an enterprise rather than a single user, department, or specialized application. |
| **EnterpriseOne object** | A re-usable piece of code that is used to build applications. Object types include tables, forms, business functions, data dictionary items, batch processes, business views, event rules, versions, data structures, and media objects. See also object. |

| | |
|---|---|
| **EnterpriseOne process** | Allows EnterpriseOne clients and servers to handle processing requests and execute transactions. A client runs one process, and servers can have multiple instances of a process. EnterpriseOne processes can also be dedicated to specific tasks (for example, workflow messages and data replication) to ensure that critical processes do not have to wait if the server is particularly busy. |
| **EnterpriseOne web development computer** | A standard EnterpriseOne Windows developer computer with the additional components installed:<br>Sun's JDK 1.1.<br>JFC (0.5.1).<br>Generator Package with Generator.Java and JDECOM.dll.<br>R2 with interpretive and application controls/form. |
| **environment workbench** | During the Installation Workbench process, Environment Workbench copies the environment information and Object Configuration Manager tables for each environment from the Planner data source to the System release number data source. It also updates the Environment Plan detail record to reflect completion. |
| **equivalent fuel** | A barrel of equivalent fuel supplies six million BTUs of heat. Fuel gas quantities are usually calculated as equivalent fuel barrels in economic calculations for refinery operations. |
| **escalation monitor** | A batch process that monitors pending requests or activities, and restarts or forwards them to the next step or user after they have been inactive for a specified amount of time. |
| **ESR** | Einzahlungsschein mit Referenznummer. A pay slip with a reference number. |
| **event rule** | [In EnterpriseOne] A logic statement that instructs the system to perform one or more operations that are based on an activity that can occur in a specific application, such as entering a form or exiting a field. |
| **exit bar** | [In EnterpriseOne] The tall pane with icons in the left portion of many EnterpriseOne program windows. |
| **facility** | An entity within a business for which you want to track costs. For example, a facility might be a warehouse location, job, project, work center, or branch/plant. Sometimes referred to as a business unit. |
| **fast path** | [In EnterpriseOne] A command prompt that allows the user to move quickly among menus and applications by using specific commands. |
| **file handle** | A temporary reference (typically a number) that is assigned to a file which has been opened by the operating system and is used throughout the session to access the file. |
| **file server** | A computer that stores files to be accessed by other computers on the network. |
| **find/browse** | A type of form used to:<br>Search, view, and select multiple records in a detail area.<br>Delete records.<br>Exit to another form.<br>Serve as an entry point for most applications. |

| | |
|---|---|
| **firm planned order (FPO)** | A work order that has reached a user defined status. When this status is entered in the processing options for the various manufacturing programs, messages for those orders are not exploded to the components. |
| **fiscal date pattern** | A representation of the beginning date for the fiscal year and the ending date for each period in that year. |
| **fix/inspect** | A type of form used to view, add, or modify existing records. A fix/inspect form has no detail area. |
| **fixed quantity** | A term that indicates the bill of material relationship between a parent item and its components or ingredients. When a bill of material component has a fixed quantity relationship to its parent, the amount of the component does not change when the software calculates parts list requirements for different work order quantities. Contrast with variable quantity. |
| **flexible account numbers** | The format of account numbers for journal entries. The format that you set up must be the three segments:<br>Business unit.<br>Object.<br>Subsidiary. |
| **form design aid (FDA)** | The EnterpriseOne GUI development tool for building interactive applications and forms. |
| **form exit** | [In EnterpriseOne] An option that is available as a button on the Form Exit bar or as a selection in the Form menu. It allows users to open an interconnected form. |
| **form interconnection** | Allows one form to access and pass data to another form. Form interconnections can be attached to any event; however, they are normally used when a button is clicked. |
| **form type** | The following form types are available in EnterpriseOne:<br>Find/browse.<br>Fix/inspect.<br>Header detail.<br>Headerless detail.<br>Message.<br>Parent/child.<br>Search/select. |
| **form-to-form call** | A request by a form for data or functionality from one of the connected forms. |
| **framework** | [In object-oriented systems] A set of object classes that provide a collection of related functions for a user or piece of software. |
| **frozen cost** | The cost of an item, operation, or process after the frozen update program is run; used by the Manufacturing Accounting system. |
| **frozen update program** | A program that freezes the current simulated costs, thereby finalizing them for use by the Manufacturing Accounting system. |

| | |
|---|---|
| **globally unique identifier (GUI)** | A 16-byte code in the Component Object Model that identifies an interface to an object across all computers and networks. |
| **handle** | [In programming] A pointer that contains the address of another pointer, which, in turn, contains the address of the desired object. |
| **hard commitment** | The number of items that are reserved for a sales order, work order, or both, from a specific location, lot, or both. |
| **hard error** | An error that cannot be corrected by a given error detection and correction system. |
| **header** | Information at the beginning of a table or form. Header information is used to identify or provide control information for the group of records that follows. |
| **header information** | Information that pertains to the entire order. |
| **hover help** | A help function that provides contextual information or instructions when a cursor moves over a particular part of the interface element for a predefined amount of time. |
| **ICMS** | Imposto sobre circulação de mercadoria e servicos. In Brazil, a state tax that is applied to the movement of merchandise and some services. |
| **ICMS Substituto** | Imposto sobre circulacao de mercadoria e servicos substituto. In Brazil, the ICMS tax that is charged on interstate transactions, or on special products and clients. |
| **ICMS Substituto-Markup** | See imposto sobre circulação de mercadoria e servicos substituto-markup. |
| **imposto de renda (IR)** | Brazilian income tax. |
| **imposto sobre produtos industrializados** | In Brazil, a federal tax that applies to manufactured goods (domestic and imported). |
| **imposto sobre services (ISS)** | In Brazil, tax on services. |
| **inbound document** | A document that is received from a trading partner using Electronic Data Interface (EDI). This document is also referred to as an inbound transaction. |
| **indented tracing** | Tracking all lot numbers of intermediates and ingredients that are consumed in the manufacture of a given lot of product, down through all levels of the bill of material, recipe, or formula. |
| **indexed allocations** | A procedure that allocates or distributes expenses, budgets, adjustments, and so on, among business units, based on a fixed percentage. |
| **indirect measurement** | Determining the quantity on-hand by:<br>Measuring the storage vessels and calculating the content's balance quantity.<br>or<br>Theoretically calculating consumption of ingredients and deducting them from the on-hand balance. |

| | |
|---|---|
| **indirect usage** | Determining what should have been used by multiplying receipt quantity of the parent times the quantity per statement in the formula, recipe, or bill of material. This transaction typically affects both consumption on schedule as well as issue from on-hand balances. |
| **in-process rework** | Recycling a semi processed product that does not meet acceptable standards. Further processing takes the product out of a given operation and sends it back to the beginning of that operation or a previous operation (for example, unreacted materials).<br><br>Rework that is detected prior to receipt of finished goods and corrected during the same schedule run. |
| **INPS withholding tax** | Instituto Nazionale di Previdenza Sociale withholding tax. In Italy, a 12% social security withholding tax that is imposed on payments to certain types of contractors. This tax is paid directly to the Italian social security office. |
| **inscrição estadual** | ICMS tax ID. In Brazil, the state tax ID. |
| **inscrição municipal** | ISS tax ID. In Brazil, the municipal tax ID. |
| **integrated toolset** | Unique to EnterpriseOne is an industrial-strength toolset that is embedded in the already comprehensive business applications. This toolset is the same toolset that is used by PeopleSoft to build EnterpriseOne interactive and batch applications. Much more than a development environment, however, the EnterpriseOne integrated toolset handles reporting and other batch processes, change management, and basic data warehousing facilities. |
| **integrity test** | A process that is used to supplement a company's internal balancing procedures by locating and reporting balancing problems and data inconsistencies. |
| **interbranch sales order** | A sales order that is used for transactions between branch/plants other than the selling branch/plant. |
| **Interoperability** | The ability of different computer systems, networks, operating systems, and applications to work together and share information. |
| **inventory pricing rule** | A discount method that is used for purchases from suppliers and sales to customers. The method is based on effectivity dates, up-to quantities, and a factor by which you can mark up or discount the price or cost. |
| **inventory turn** | The number of times that the inventory cycles, or turns over, during the year. A frequently used method to compute inventory turnover is to divide the annual costs of sales by the average inventory level. |
| **invoice** | An itemized list of goods that are shipped or services that are rendered, stating quantities, prices, fees, shipping charges, and so on. Companies often have their invoices mailed to a different address than where they ship products. In such cases, the bill-to address differs from the ship-to address. |
| **IP** | See imposto sobre produtos industrializados. |
| **IR** | See imposto de renda. |
| **IServer Service** | Developed by PeopleSoft, this Internet server service resides on the Web server and is used to speed up delivery of the Java class files from the database to the client. |

| | |
|---|---|
| **ISS** | See imposto sobre servicos. |
| **jargon** | An alternate data dictionary item description that EnterpriseOne or PeopleSoft World displays, based on the product code of the current object. |
| **java application server** | A component-based server that resides in the middle-tier of a server-centric architecture and provides middleware services for security and state maintenance, along with data access and persistence. |
| **JDBNET** | A database driver that allows heterogeneous servers to access each other's data. |
| **jde.ini** | A PeopleSoft file (or member for AS/400) that provides the runtime settings that are required for EnterpriseOne initialization. Specific versions of the file or member must reside on every machine that is running EnterpriseOne, including workstations and servers. |
| **JDE.LOG** | The main diagnostic log file of EnterpriseOne. Always located in the root directory on the primary drive. Contains status and error messages from the startup and operation of EnterpriseOne. |
| **JDEBASE Database Middleware** | PeopleSoft proprietary database middleware package that provides two primary benefits:<br>1. Platform-independent APIs for multidatabase access. These APIs are used in two ways:<br>a. By the interactive and batch engines to dynamically generate platform-specific SQL, depending on the data source request.<br>b. As open APIs for advanced C business function writing. These APIs are then used by the engines to dynamically generate platform-specific SQL.<br>2. Client-to-server and server-to-server database access. To accomplish this access, EnterpriseOne is integrated with a variety of third-party database drivers, such as Client Access 400 and open database connectivity (ODBC). |
| **JDECallObject** | An application programming interface that is used by business functions to invoke other business functions. |
| **JDEIPC** | Communications programming tools that are used by server code to regulate access to the same data in multiprocess environments, communicate and coordinate between processes, and create new processes. |
| **JDENET** | PeopleSoft proprietary middleware software. JDENET is a messaging software package. |
| **JDENET communications middleware** | PeopleSoft proprietary communications middleware package for EnterpriseOne. It is a peer-to-peer, message-based, socket-based, multiprocess communications middleware solution. It handles client-to-server and server-to-server communications for all EnterpriseOne supported platforms. |
| **just in time installation (JITI)** | EnterpriseOne's method of dynamically replicating objects from the central object location to a workstation. |
| **just in time replication (JITR)** | EnterpriseOne's method of replicating data to individual workstations. EnterpriseOne replicates new records (inserts) only at the time that the user needs the data. Changes, deletes, and updates must be replicated using Pull Replication. |

| | |
|---|---|
| **Kagami** | In Japan, summarized invoices that are created monthly (in most cases) to reduce the number of payment transactions. |
| **latitude** | The X coordinate of the location of an item in the warehouse. The system can use latitude, longitude, and height when suggesting locations for putaway, replenishment, and picking. |
| **laytime (or layhours)** | The amount of time that is allotted to a tanker at berth to complete loading or discharging cargo. This time is usually expressed in running hours, and is fixed by prior agreement between the vessel owner and the company that is chartering the vessel. Laytime is stipulated in the charter, which states exactly the total of number of hours that are granted at both loading and unloading ports, and indicates whether such time is reversible. A statement of "Seventy-Two Hours, Reversible" means that a total of 72 hours is granted overall at both ports, and any time saved at one port can be applied as a credit at the other port.<br><br>For example, if the vessel uses only 32 hours instead of 36 hours to load cargo, it can apply an additional four hours to the 36 hours allotted at the discharge port. Such considerations are important for purposes of computing demurrage. |
| **leading zeros** | A series of zeros that certain facilities in PeopleSoft systems place in front of a value that is entered. This situation normally occurs when you enter a value that is smaller than the specified length of the field. For example, if you enter 4567 in a field that accommodates eight numbers, the facility places four zeros in front of the four numbers that you enter. The result appears as 00004567. |
| **ledger type** | A code that designates a ledger which is used by the system for a particular purpose. For example, all transactions are recorded in the AA (actual amounts) ledger type in their domestic currency. The same transactions can also be stored in the CA (foreign currency) ledger type. |
| **level break** | The position in a report or text where a group of similar types of information ends and another one begins. |
| **libro IVA** | Monthly VAT report. In Italy, the term for the report that contains the detail of invoices and vouchers that were registered during each month. |
| **line of business** | A description of the nature of a company's work; also a tool to control the relationship with that customer, including product pricing. |
| **linked service type** | A service type that is associated with a primary service type. Linked service types can be cancelled, and the maintenance tasks are performed when the primary service type to which they are linked comes due. You can specify whether the system generates work orders for linked service types, as well as the status that the system assigns to work orders that have already been generated. Sometimes referred to as associated service types. See also primary service type and service type. |
| **livro razao** | In Brazil, a general ledger report. |
| **load balancing** | The act of distributing the number of processes proportionally to all servers in a group to maximize overall performance. |
| **location workbench** | During the Installation Workbench process, Location Workbench copies all locations that are defined in the installation plan from the Location Master table in the Planner data source to the System data source. |

| | |
|---|---|
| **log files** | Files that track operations for a process or application. Reviewing log files is helpful for troubleshooting problems. The file extension for log files is .LOG. |
| **logic data source** | Any code that provides data during runtime. |
| **logical compartment** | One of two ways that is identified in the transportation constants to display compartments on vehicles. Logical display numbers the compartments sequentially.<br>For example, if two vehicles are on a trip and each vehicle has three compartments, the logical display is 1,2,3,4,5,6. |
| **logical file** | A set of keys or indices that is used for direct access or ordered access to the records in a physical file. Several logical files can have different accesses to a physical. |
| **logical shelf** | A logical, not physical, location for inventory that is used to track inventory transactions in loan/borrow, or exchange agreements with other companies. See also logical warehouse. |
| **logical warehouse** | Not a physical warehouse containing actual inventory, but a means for storing and tracking information for inventory transactions in loan/borrow, or exchange agreements with other companies. |
| **longitude** | The Y coordinate of the location of an item in the warehouse. The system can use latitude, longitude, and height when suggesting locations for putaway, replenishment, and picking. |
| **LSV** | Lastschriftverfahren. A Swiss auto debit format that is required by Telekurs (Payserv). |
| **mail merge** | A mass-mail facility that takes names, addresses, and (sometimes) pertinent facts about recipients and merges the information into a form letter or a similarly basic document. |
| **mailmerge workbench** | [In EnterpriseOne] An application that merges Microsoft Word 6.0 (or higher) word-processing documents with EnterpriseOne records to automatically print business documents. |
| **main fuels** | Usually refers to bulk fuel products, but sometimes includes packaged products. |
| **maintenance loop** | See maintenance route. |
| **maintenance route** | A method of performing PMs for multiple pieces of equipment from a single preventive maintenance work order. A maintenance route includes pieces of equipment that share one or more identical maintenance tasks which can be performed at the same time for each piece of equipment. Sometimes referred to as maintenance loop. |
| **maintenance work order** | In PeopleSoft EnterpriseOne systems, a term that is used to distinguish work orders created for the performance of equipment and plant maintenance from other work orders, such as manufacturing work orders, utility work orders, and engineering change orders. |

| | |
|---|---|
| **manufacturing and distribution planning** | Planning that includes resource and capacity planning, and material planning operations. Resource and capacity planning allows you to prepare a feasible production schedule that reflects your demand forecasts and production capability. Material Planning Operations provides a short-range plan to cover material requirements that are needed to make a product. |
| **mapping** | A set of instructions that describes how one data structure passes data to another. |
| **master business function** | An interactive master file that serves as a central location for adding, changing, and updating information in a database. |
| **master business function** | A central system location for standard business rules about entering documents, such as vouchers, invoices, and journal entries. Master business functions ensure uniform processing according to guidelines that you establish. |
| **master table** | A database table that is used to store data and information that is permanent and necessary to the system's operation. Master tables might contain data such as paid tax amounts, supplier names, addresses, employee information, and job information. |
| **matching document** | A document that is associated with an original document to complete or change a transaction. For example, a receipt is the matching document of an invoice. |
| **media object** | An electronic or digital representation of an object. |
| **media storage objects** | Files that use one of the following naming conventions that are not organized into table format: Gxxx, xxxGT, or GTxxx. |
| **memory violation** | An error that occurs as the result of a memory leak. |
| **menu selection** | An option on a menu that initiates a software function directly. |
| **message center** | A central location for sending and receiving all EnterpriseOne messages (system- and user-generated), regardless of the originating application or user. |
| **messaging application programming interface (MAPI)** | An architecture that defines the components of a messaging system and how they behave. It also defines the interface between the messaging system and the components. |
| **metal content** | A series of properties of a blended product that help to determine its suitability for a prescribed purpose. |
| **metals management** | The process of maintaining information about the location and status of durable product containers such as liquid petroleum gas (LPG) cylinders. |
| **mobile inventory** | Inventory that is transferred from a depot to a barge or truck for milk-run deliveries. |
| **modal** | A restrictive or limiting interaction that is created by a given condition of operation. Modal often describes a secondary window that restricts a user's interaction with other windows. A secondary window can be modal with respect to its primary window or to the entire system. A modal dialog box must be closed by the user before the application continues. |

| | |
|---|---|
| **model work order** | For scheduled preventive maintenance or for a condition-based alert, a model work order functions as a template for the creation of other work orders. You can assign model work orders to service types and condition-based alerts. When the service type comes due or the alert is generated, the system automatically generates a work order that is based on information from the model work order. |
| **modeless** | Not restricting or limiting interaction. Modeless often describes a secondary window that does not restrict a user's interaction with other windows. A modeless dialog box stays on the screen and is available for use at any time, but also permits other user activities. |
| **multiple stocking locations** | Authorized storage locations for the same item number at locations, in addition to the primary stocking location. |
| **multitier architecture** | A client/server architecture that allows multiple levels of processing. A tier defines the number of computers that can be used to complete some defined task. |
| **named event rules (NER)** | Also called business function event rules. Encapsulated, re-usable business logic that is created by using event rules, rather than C programming. |
| **national language support (NLS)** | Mechanisms that are provided to facilitate internationalization of both system and application user interfaces. |
| **natureza da operação** | Transaction nature. In Brazil, a code that classifies the type of commercial transaction to conform to the fiscal legislation. |
| **negative pay item** | An entry in an account that indicates a prepayment. For example, you might prepay a supplier before goods are sent or prepay an employee's forecasted expenses for a business trip. The system stores these pending entries, assigning them a minus quantity as debit amounts in a designated expense account. After the prepaid goods are received or the employee submits an expense report, entering the actual voucher clears all of the negative pay items by processing them as regular pay items. Note that a negative pay item can also result from entering a debit memo (A/P) or a credit memo (A/R). |
| **net added cost** | The cost to manufacture an item at the current level in the bill of material. Thus, for manufactured parts, the net added cost includes labor, outside operations, and cost extras applicable to this level in the bill of material, but not materials (lower-level items). For purchased parts, the net added cost also includes the cost of materials. |
| **next status** | The next step in the payment process for payment control groups. The next status can be either WRT (write) or UPD (update). |
| **node** | A termination point for two or more communications links. A node can serve as the control location for forwarding data among the elements of a network or multiple networks, as well as performing other networking and, in some cases, local processing. |
| **non-inventory items** | See non-stock items. |
| **non-list price** | A price for bulk products that is determined by its own algorithms, such as a rolling average or commodity price plus. |
| **non-prime product** | A manufactured product with revenue potential that is less than the product planned for, or scheduled to be produced. |

| | |
|---|---|
| **non-stock items** | Items that the system does not account for as part of the inventory. For example, office supplies, or packaging materials can be non-stock items. |
| **nota fiscal** | In Brazil, a legal document that must accompany all commercial transactions. |
| **nota fiscal fatura** | In Brazil, a nota fiscal and invoice information. |
| **notula** | In Italy, the process whereby a business does not recognize value added tax until the payment of a voucher. |
| **object configuration manager (OCM)** | EnterpriseOne's object request broker and the control center for the runtime environment. It keeps track of the runtime locations for business functions, data, and batch applications. When one of these objects is called, the Object Configuration Manager directs access to it by using defaults and overrides for a given environment and user. |
| **object embedding** | When an object is embedded in another document, an association is maintained between the object and the application that created it; however, any changes made to the object are also only kept in the compound document. See also object linking. |
| **object librarian** | A repository of all versions, applications, and business functions that are re-usable in building applications. |
| **object linking** | When an object is linked to another document, a reference is created with the file in which the object is stored, as well as with the application that created it. When the object is modified, either from the compound document or directly through the file in which it is saved, the change is reflected in that application as well as anywhere it has been linked. See also object embedding. |
| **object linking and embedding (OLE)** | A technology for transferring and sharing information among applications by allowing the integration of objects from diverse applications, such as graphics, charts, spreadsheets, text, or an audio clip from a sound program. OLE is a compound document standard that was developed by Microsoft Corporation. It enables you to create objects with one application, and then link or embed them in a second application. Embedded objects retain their original format and links to the application that created them. See also object embedding, object linking. |
| **object management workbench (OMW)** | The change management system that is used for EnterpriseOne development. |
| **object-based technology (OBT)** | A technology that supports some of the main principles of object-oriented technology:<br>Classes.<br>Polymorphism.I<br>Inheritance.<br>Encapsulation. |

| | |
|---|---|
| **object-oriented technology (OOT)** | Brings software development past procedural programming into a world of re-usable programming that simplifies development of applications. Object orientation is based on the following principles:<br>Classes.<br>Polymorphism.I<br>Inheritance.<br>Encapsulation. |
| **offsetting account** | An account that reduces the amount of another account to provide a net balance. For example, a credit of 200 to a cash account might have an offsetting entry of 200 to an A/P Trade (liability) account. |
| **open database connectivity (ODBC)** | Defines a standard interface for different technologies to process data between applications and different data sources. The ODBC interface comprises set of function calls, methods of connectivity, and representation of data types that define access to data sources. |
| **open systems interconnection (OSI)** | The OSI model was developed by the International Standards Organization (ISO) in the early 1980s. It defines protocols and standards for the interconnection of computers and network equipment. |
| **order detail line** | A part of an order that contains transaction information about a service or item being purchased or sold, such as quantity, cost, price, and so on. |
| **order hold** | A flag that stops the processing of an order because it has exceeded the credit or budget limit, or has another problem. |
| **order-based pricing** | Pricing strategy that grants reductions in price to a customer. It is based upon the contents and relative size (volume or value) of the order as a whole. |
| **outbound document** | A document that is sent to a trading partner using EDI. This term is also referred to as an outbound transaction. |
| **outturn** | The quantity of oil that is actually received into a buyer's storage tanks when a vessel is unloaded. For various reasons (vaporization, clingage to vessel tank walls, and so on), the amount of a product pumped into shore tankage at unloading is often less than the quantity originally loaded onto the vessel, as certified by the Bill of Lading. Under a delivered or CIF outturn transaction, the buyer pays only for the barrels actually "turned out" by the vessel into storage.<br>When a buyer is paying CIF Bill of Lading figures, a loss of 0.5% of total cargo volume is considered normal. Losses in excess of 0.5%, however, are either chargeable to the seller or are covered by specialized insurance that covers partial, as well as total, loss of the cargo. |
| **overhead** | In the distillation process, that portion of the charge that leaves the top of the distillation column as vapor. This definition is strictly as it relates to ECS. |
| **override conversion method** | A method of calculating exchange rates that is set up between two specific currencies. For those specific currencies, this method overrides the conversion method in General Accounting Constants and does not allow inverse rates to be used when calculating currency amounts. |

| | |
|---|---|
| **package / package build** | A collection of software that is grouped into a single entity for modular installation. EnterpriseOne objects are installed to workstations in packages from the deployment server. A package can be compared to a bill of material or kit that indicates the necessary objects for that workstation and where the installation program can find them on the deployment server. It is a point-in-time "snapshot" of the central objects on the deployment server. |
| **package location** | The directory structure location for the package and its set of replicated objects. This location is usually \\deployment server\release\path_code\package\ package name. The replicated objects for the package are placed in the subdirectories under this path. This location is also where the package is built or stored. |
| **package workbench** | During the Installation Workbench process, Package Workbench transfers the package information tables from the Planner data source to the System - release number data source. It also updates the Package Plan detail record to reflect completion. |
| **packaged products** | Products that, by their nature, must be delivered to the customer in containers which are suitable for discrete consumption or resale. |
| **pane/panel** | A resizable subarea of a window that contains options, components, or other related information. |
| **paper clip** | An icon that is used to indicate that a media object is attached to a form or record. |
| **parent/child form** | A type of form that presents parent/child relationships in an application on one form:<br>The left portion of the form presents a tree view that displays a visual representation of a parent/child relationship.<br>The right portion of the form displays a detail area in browse mode. The detail area displays the records for the child item in the tree.<br>The parent/child form supports drag and drop functionality. |
| **parent/child relationship** | See parent/component relationship. |
| **parent/component relationship** | 1. In Capital Asset Management, the hierarchical relationship of a parent piece of equipment to its components. For example, a manufacturing line could be a parent and the machinery on the line could be components of the line. In addition, each piece of machinery could be a parent of still more components.<br>2. In Product Data Management, a hierarchical relationship of the components and subassemblies of a parent item to that parent item. For example, an automobile is a parent item; its components and subassemblies include: engine, frame, seats, and windows.<br>Sometimes referred to as parent/child relationship. |
| **partita IVA** | In Italy, a company fiscal identification number. |
| **pass-through** | A process where data is accepted from a source and forwarded directly to a target without the system or application performing any data conversion, validation, and so on. |
| **pay on consumption** | The method of postponing financial liability for component materials until you issue that material to its consuming work order or rate schedule. |

| | |
|---|---|
| **payment group** | A system-generated group of payments with similar information, such as a bank account. The system processes all of the payments in a payment group at the same time. |
| **PeopleSoft database** | See JDEBASE Database Middleware. |
| **performance tuning** | The adjustments that are made for a more efficient, reliable, and fast program. |
| **persistent object** | An object that continues to exist and retains its data beyond the duration of the process that creates it. |
| **pervasive device** | A type of intelligent and portable device that provides a user with the ability to receive and gather information anytime, from anywhere. |
| **planning family** | A means of grouping end items that have similarity of design or manufacture. |
| **plug-in** | A small program that plugs into a larger application to provide added functionality or enhance the main application. |
| **polymorphism** | A principle of object-oriented technology in which a single mnemonic name can be used to perform similar operations on software objects of different types. |
| **portal** | A Web site or service that is a starting point and frequent gateway to a broad array of on-line resources and services. |
| **Postfinance** | A subsidiary of the Swiss postal service. Postfinance provides some banking services. |
| **potency** | Identifies the percent of an item in a given solution. For example, you can use an 80% potent solution in a work order that calls for 100% potent solution, but you would use 25% more, in terms of quantity, to meet the requirement (100 / 80 = 1.25). |
| **preference profile** | The ability to define default values for specified fields for a user defined hierarchy of items, item groups, customers, and customer groups. In Quality Management setup, this method links test and specification testing criteria to specific items, item groups, customers, or customer groups. |
| **preflush** | A work order inventory technique in which you deduct (relieve) materials from inventory when the parts list is attached to the work order or rate schedule. |
| **preventive maintenance cycle** | The sequence of events that make up a preventive maintenance task, from its definition to its completion. Because most preventive maintenance tasks are commonly performed at scheduled intervals, parts of the preventive maintenance cycle repeat, based on those intervals. |
| **preventive maintenance schedule** | The combination of service types that apply to a specific piece of equipment, as well as the intervals at which each service type is scheduled to be performed. |
| **primary service type** | A service type to which you can link related service types. For example, for a particular piece of equipment, you might set up a primary service type for a 1000-hour inspection and a linked service type for a 500-hour inspection. The 1000-hour inspection includes all of the tasks performed at 500 hours. When a primary service type is scheduled to be performed, the system schedules the linked service type. See also linked service type. |

| | |
|---|---|
| **pristine environment** | An EnterpriseOne environment that is used to test unaltered objects with PeopleSoft demonstration data or for training classes. You must have this environment so you can compare pristine objects that you modify. |
| **processing option** | A data structure that allows users to supply parameters that regulate the execution of a batch program or report. |
| **product data management (PDM)** | In PeopleSoft EnterpriseOne software, the system that enables a business to organize and maintain information about each item which it manufactures. Features of this system, such as bills of material, work centers, and routings, define the relationships among parents and components, and how they can be combined to manufacture an item. PDM also provides data for other manufacturing systems including Manufacturing Accounting, Shop Floor Management, and Manufacturing and Distribution Planning. |
| **product line** | A group of products with similarity in manufacturing procedures, marketing characteristics, or specifications that allow them to be aggregated for planning; marketing; and, occasionally, costing. |
| **product/process definition** | A combination of bill of material (recipe, formula, or both) and routing (process list). Organized into tasks with a statement of required consumed resources and produced resources. |
| **production environment** | An EnterpriseOne environment in which users operate EnterpriseOne software. |
| **program temporary fix (PTF)** | A representation of changes to PeopleSoft software that your organization receives on magnetic tapes or diskettes. |
| **project** | [In EnterpriseOne] A virtual container for objects being developed in Object Management Workbench. |
| **projected cost** | The target expenditure in added value for material, labor, and so on, during manufacture. See also standard cost. |
| **promotion path** | The designated path for advancing objects or projects in a workflow. |
| **protocollo** | See registration number. |
| **PST** | Provincial sales tax. A tax that is assessed by individual provinces in Canada. |
| **published table** | Also called a “Master” table, this is the central copy to be replicated to other machines and resides on the “publisher” machine. The Data Replication Publisher Table (F98DRPUB) identifies all of the published tables and their associated publishers in the enterprise. |
| **publisher** | The server that is responsible for the published table. The Data Replication Publisher Table (F98DRPUB) identifies all of the published tables and their associated publishers in the enterprise. |
| **pull replication** | One of the EnterpriseOne methods for replicating data to individual workstations. Such machines are set up as pull subscribers that use EnterpriseOne’s data replication tools. The only time that pull subscribers are notified of changes, updates, and deletions is when they request such information. The request is in the form of a message that is sent, usually at startup, from the pull subscriber to the server machine that stores the Data Replication Pending Change Notification table (F98DRPCN). |

| | |
|---|---|
| **query by example (QBE)** | Located at the top of a detail area, this area is used to search for data to display in the detail area. |
| **rate scheduling** | A method of scheduling product or manufacturing families, or both.<br>Also a technique to determine run times and quantities of each item within the family to produce enough of each individual product to satisfy demand until the family can be scheduled again. |
| **rate type** | For currency exchange transactions, the rate type distinguishes different types of exchange rates. For example, you can use both period average and period-end rates, distinguishing them by rate type. |
| **real-time** | Pertaining to information processing that returns a result so rapidly that the interaction appears to be instantaneous. |
| **receipt routing** | A series of steps that is used to track and move items within the receipt process. The steps might include in-transit, dock, staging area, inspection, and stock. |
| **referential integrity** | Ensures that a parent record cannot be deleted from the database when a child record for exists. |
| **regenerable** | Source code for EnterpriseOne business functions can be regenerated from specifications (business function names). Regeneration occurs whenever an application is recompiled, either for a new platform or when new functionality is added. |
| **register types and classes** | In Italian VAT Summary Reporting, the classification of VAT transactions. |
| **relationship** | Links tables together and facilitates joining business views for use in an application or report. Relationships that are created are based on indexes. |
| **rélevé d'identité bancaire (RIB)** | In France, the term that indicates the bank transit code, account number, and check digit that are used to validate the bank transit code and account number. The bank transit code consists of the bank code and agency code. The account number is alphanumeric and can be as many as 11 characters. PeopleSoft supplies a validation routine to ensure RIB key correctness. |
| **remessa** | In Brazil, the remit process for A/R. |
| **render** | To include external data in displayed content through a linking mechanism. |
| **repasse** | In Brazil, a discount of the ICMS tax for interstate transactions. It is the adjustment between the interstate and the intrastate ICMS tax rates. |
| **replenishment point** | The location on or near the production line where additional components or subassemblies are to be delivered. |
| **replication server** | A server that is responsible for replicating central objects to client machines. |
| **report design aid (RDA)** | The EnterpriseOne GUI tool for operating, modifying, and copying report batch applications. |
| **repost** | In Sales, the process of clearing all commitments from locations and restoring commitments, based on quantities from the Sales Order Detail table (F4211). |
| **resident** | Pertaining to computer programs or data while they remain on a particular storage device. |

| | |
|---|---|
| **retorno** | In Brazil, the receipt process for A/R. |
| **RIB** | See rélevé d'identité bancaire. |
| **ricevute bancarie (RiBa)** | In Italy, the term for accounts receivable drafts. |
| **riepilogo IVA** | Summary VAT monthly report. In Italy, the term for the report that shows the total amount of VAT credit and debit. |
| **ritenuta d'acconto** | In Italy, the term for standard withholding tax. |
| **rollback** | [In database management] A feature or command that undoes changes in database transactions of one or more records. |
| **rollup** | See cost rollup. |
| **row exit** | [In EnterpriseOne] An application shortcut, available as a button on the Row Exit bar or as a menu selection, that allows users to open a form that is related to the highlighted grid record. |
| **runtime** | The period of time when a program or process is running. |
| **SAD** | The German name for a Swiss payment format that is accepted by Postfinance. |
| **SAR** | See software action request. |
| **scalability** | The ability of software, architecture, hardware, or a network to support software as it grows in size or resource requirements. |
| **scripts** | A collection of SQL statements that perform a specific task. |
| **scrub** | To remove unnecessary or unwanted characters from a string. |
| **search/select** | A type of form that is used to search for a value and return it to the calling field. |
| **selection** | Found on PeopleSoft menus, selections represent functions that you can access from a menu. To make a selection, type the associated number in the Selection field and press Enter. |
| **serialize** | To convert a software object into a stream of bytes to store on a disk or transfer across a network. |
| **server map** | The server view of the object configuration mapping. |
| **server workbench** | During the Installation Workbench process, Server Workbench copies the server configuration files from the Planner data source to the System release number data source. It also updates the Server Plan detail record to reflect completion. |
| **service interval** | The frequency at which a service type is to be performed. Service intervals can be based on dates, periods, or statistical units that are user defined. Examples of statistical units are hours, miles, and fuel consumption. |
| **service type** | An individual preventive maintenance task or procedure, such as an inspection, lubrication, or overhaul. Service types can apply to a specific piece of equipment or to a class of equipment. You can specify that service types come due based on a predetermined service interval, or whenever the task that is represented by the service type becomes necessary. |

| | |
|---|---|
| **servlet** | A [small] program that extends the functionality of a Web server by generating dynamic content and interacting with Web clients by using a request-response paradigm. |
| **share path** | The network node under which one or more servers or objects reside. |
| **shop floor management** | A system that uses data from multiple system codes to help develop, execute, and manage work orders and rate schedules in the enterprise. |
| **silent mode** | A method for installing or running a program that does not require any user intervention. |
| **silent post** | A type of post that occurs in the background without the knowledge of the user. |
| **simulated cost** | After a cost rollup, the cost of an item, operation, or process according to the current cost scenario. This cost can be finalized by running the frozen update program. You can create simulated costs for a number of cost methods—for example, standard, future, and simulated current costs. See also cost rollup. |
| **single-byte character set (SBCS)** | An encoding scheme in which each alphabetic character is represented by one byte. Most Western languages, such as English, can be represented by using a single-byte character set. |
| **single-level tracking** | Finding all immediate parents where a specific lot has been used (consumed). |
| **single-voyage (spot) charter** | An agreement for a single voyage between two ports. The payment is made on the basis of tons of product delivered. The owner of the vessel is responsible for all expenses. |
| **slimer** | A script that changes data in a table directly without going through a regular database interface. |
| **smart field** | A data dictionary item with an attached business function for use in the Report Design Aid application. |
| **SOC** | The Italian term for a Swiss payment format that is accepted by Postfinance. |
| **soft commitment** | The number of items that is reserved for sales orders or work orders in the primary units of measure. |
| **soft error** | An error from which an operating system or program is able to recover. |
| **software action request (SAR)** | An entry in the AS/400 database that is used for requesting modifications to PeopleSoft software. |
| **SOG** | The French term for a Swiss payment format that is accepted by Postfinance. |
| **source directory** | The path code to the business function source files belonging to the shared library that is created on the enterprise server. |
| **special period/year** | The date that determines the source balances for an allocation. |

| | |
|---|---|
| **specification merge** | The Specification merge is comprised of three merges:<br>Object Librarian merge (via the Object Management Workbench).<br>Versions List merge.<br>Central Objects merge.<br>The merges blend customer modifications with data that accompanies a new release. |
| **specification table merge workbench** | During the Installation Workbench process, Specification Table Merge Workbench runs the batch applications that update the specification tables. |
| **specifications** | A complete description of an EnterpriseOne object. Each object has its own specification, or name, which is used to build applications. |
| **spot charter** | See single-voyage charter. |
| **spot rates** | An exchange rate that is entered at the transaction level. Spot rates are not used on transactions between two EMU member currencies because exchange rates are irrevocably fixed to the euro. |
| **stamp tax** | In Japan, a tax that is imposed on drafts payable, receipts over 30000 Japanese yen, and all contracts. The party that issues any of the above documents is responsible for this tax. |
| **standalone** | Operating or capable of operating independently of certain other components of a computer system. |
| **standard cost** | The expected, or target cost of an item, operation, or process. Standard costs represent only one cost method in the Product Costing system. You can also calculate, for example, future costs or current costs. However, the Manufacturing Accounting system uses only standard frozen costs. |
| **standard costing** | A costing method that uses cost units that are determined before production. For management control purposes, the system compares standard costs to actual costs and computes variances. |
| **subprocess** | A process that is triggered by and is part of a larger process, and that generally consists of activities. |
| **subscriber table** | The Subscriber table (F98DRSUB), which is stored on the Publisher Server with the Data Replication Publisher table (F98DRPUB), that identifies all of the subscriber machines for each published table. |
| **summary** | The presentation of data or information in a cumulative or totaled manner in which most of the details have been removed. Many systems offer forms and reports that summarize information which is stored in certain tables. Contrast with detail. |
| **super backflush** | To create backflush transactions for material, labor, or both, against a work order at predefined pay points in the routing. By doing so, you can relieve inventory and account for labor amounts at strategic points throughout the manufacturing process. |
| **supersession** | Specification that a new product is replacing an active product on a specified effective date. |

| | |
|---|---|
| **supplemental data** | Additional types of data for customers and suppliers. You can enter supplemental data for information such as notes, comments, plans, or other information that you want in a customer or supplier record. The system maintains this data in generic databases, separate from the standard master tables (Customer Master, Supplier Master, and Address Book Master). |
| **supplying location** | The location from which inventory is transferred once quantities of the item on the production line have been depleted. In kanban processing, the supplying location is the inventory location from which materials are transferred to the consuming location when the containers are replenished. |
| **system code** | A numeric or alphanumeric designation that identifies a specific system in EnterpriseOne software. |
| **system function** | [In EnterpriseOne] A named set of pre-packaged, re-usable instructions that can be called from event rules. |
| **table access management (TAM)** | The EnterpriseOne component that handles the storage and retrieval of user defined data. TAM stores information such as data dictionary definitions; application and report specifications; event rules; table definitions; business function input parameters and library information; and data structure definitions for running applications, reports, and business functions. |
| **table conversion workbench** | During the Installation Workbench process, Table Conversion Workbench runs the table conversions that change the technical and application tables to the format for the new release of EnterpriseOne. It also updates the Table Conversions and Controls detail records to reflect completion. |
| **table design aid (TDA)** | An EnterpriseOne GUI tool for creating, modifying, copying, and printing database tables. |
| **table event rules** | Use table event rules to attach database triggers (or programs) that automatically run whenever an action occurs against the table. An action against a table is referred to as an event. When you create an EnterpriseOne database trigger, you must first determine which event will activate the trigger. Then, use Event Rules Design to create the trigger. Although EnterpriseOne allows event rules to be attached to application events, this functionality is application-specific. Table event rules provide embedded logic at the table level. |
| **table handle** | A pointer into a table that indicates a particular row. |
| **table space** | [In relational database management systems] An abstract collection of containers in which database objects are stored. |
| **task** | [In Solution Explorer and EnterpriseOne Menu] A user defined object that can initiate an activity, process, or procedure. |
| **task view** | A group of tasks in Solution Explorer or EnterpriseOne Menu that are arranged in a tree structure. |
| **termo de abertura** | In Brazil, opening terms for the transaction journal. |
| **termo de encerramento** | In Brazil, closing terms for the transaction journal. |
| **three-tier processing** | The task of entering, reviewing, approving, and posting batches of transactions. |

| | |
|---|---|
| **three-way voucher match** | The process of comparing receipt information to supplier's invoices to create vouchers. In a three-way match, you use the receipt records, the purchase order, and the invoice to create vouchers. |
| **threshold percentage** | In Capital Asset Management, the percentage of a service interval that you define as the trigger for maintenance to be scheduled. For example, you might set up a service type to be scheduled every 100 hours with a threshold percentage of 90 percent. When the equipment accumulates 90 hours, the system schedules the maintenance. |
| **throughput agreement** | A service agreement in which a business partner agrees to store and manage product for another business partner for a specified time period. The second partner actually owns the stock that is stored in the first partner's depot, although the first partner monitors the stock level; suggests replenishments; and unloads, stores, and delivers product to the partner or its customers. The first partner charges a fee for storing and managing the product. |
| **throughput reconciliation** | Reconcile confirmed sales figures in a given period with the measured throughput, based on the meter readings. This process is designed to catch discrepancies that are due to transactions not being entered, theft, faulty meters, or some combination of these factors. This reconciliation is the first stage. See also operational reconciliation. |
| **token** | [In Object Management Workbench] A flag that is associated with each object which indicates whether you can check out the object. |
| **tolerance range** | The amount by which the taxes that you enter manually can vary from the tax that is calculated by the system. |
| **TP monitor** | Transaction Processing monitor. A monitor that controls data transfer between local and remote terminals and the applications that originated them. TP monitors also protect data integrity in the distributed environment and can include programs that validate data and format terminal screens. |
| **tracing** | The act of researching a lot by going backward, to discover its origin. |
| **tracking** | The act of researching a lot by going forward, to discover where it is used. |
| **transaction set** | An electronic business transaction (EDI Standard document) composed of segments. |
| **transclude** | To include the external data in the displayed content through a linking mechanism. |
| **transfer order** | An order that is used to ship inventory between branch/plants within your company and to maintain an accurate on-hand inventory amount. An interbranch transfer order creates a purchase order for the shipping location and a sales order for the receiving location. |
| **translation adjustment account** | An optional G/L account used in currency balance restatement to record the total adjustments at a company level. |
| **translator software** | The software that converts data from an application table format to an EDI Standard Format, and from EDI Standard Format to application table format. The data is exchanged in an EDI Standard, such as ANSI ASC X12, EDIFACT, UCS, or WINS. |

| | |
|---|---|
| **tree structure** | A type of graphical user interface that displays objects in a hierarchy. |
| **trigger** | Allows you to attach default processing to a data item in the data dictionary. When that data item is used on an application or report, the trigger is invoked by an event which is associated with the data item. EnterpriseOne also has three visual assist triggers:<br>Calculator.<br>Calendar.<br>Search form. |
| **two-way voucher match** | The process of comparing purchase order detail lines to the suppliers' invoices to create vouchers. You do not record receipt information. |
| **universal batch engine (UBE)** | [In EnterpriseOne] A type of application that runs a noninteractive process. |
| **unnormalized** | Data that is a random collection of data elements with repeating record groups scattered throughout. Also see Normalized. |
| **user overrides merge** | The User Overrides merge adds new user override records into a customer's user override table. |
| **user-defined code (UDC)** | A value that a user has assigned as being a valid entry for a given or specific field. |
| **utility** | A small program that provides an addition to the capabilities which are provided by an operating system. |
| **variable numerator allocations** | A procedure that allocates or distributes expenses, budgets, adjustments, and so on, among business units, based on a variable. |
| **variable quantity** | A term that indicates the bill of material relationship between a parent item and its components or ingredients. When a bill of material component has a variable quantity relationship to its parent, the amount of the component changes when the software calculates parts list requirements for different work order quantities. Contrast with fixed quantity. |
| **variance** | 1. In Product Costing and Manufacturing Accounting, the difference between the frozen standard cost, the current cost, the planned cost, and the actual cost. For example, the difference between the frozen standard cost and the current cost is an engineering variance. Frozen standard costs come from the Cost Components table, and the current costs are calculated by using the current bill of material, routing, and overhead rates.<br>2. In Capital Asset Management, the difference between revenue that is generated by a piece of equipment and costs that are incurred by the equipment. |
| **versions list merge** | The Versions List merge preserves any non-XJDE and non-ZJDE version specifications for objects that are valid in the new release as well as their processing options data. |
| **VESR** | Verfahren Einzahlungsschein mit Referenznummer. The processing of an ESR pay slip with reference line through accounts receivable and accounts payable. |
| **visual assist** | Forms that can be invoked from a control to assist the user in determining what data belongs in the control. |

| | |
|---|---|
| **voucher logging** | The process of entering vouchers without distributing amounts to specific G/L accounts. The system initially distributes the total amount of each voucher to a G/L suspense account, where it is held until you redistribute it to the correct G/L account. |
| **wareki date format** | In Japan, a calendar format, such as Showa or Heisei. When a new emperor begins to reign, the government chooses the title of the date format and the year starts over at one. For instance, January 1, 1998, is equal to Heisei 10, January 1st. |
| **wash down** | A minor cleanup between similar product runs. Sometimes used in reference to the sanitation process of a food plant. |
| **wchar_t** | An internal type of a wide character. Used for writing portable programs for international markets. |
| **web server** | A server that sends information as requested by a browser and uses the TCP/IP set of protocols. |
| **work order life cycle** | In Capital Asset Management, the sequence of events through which a work order must pass to accurately communicate the progress of the maintenance tasks that it represents. |
| **workfile** | A system-generated file that is used for temporary data processing. |
| **workflow** | According to the Workflow Management Coalition, worlflow means “the automation of a business process, in whole or part, during which documents, information, or tasks are passed from one participant to another for action, according to a set of procedural rules.” |
| **workgroup server** | A network server usually containing subsets of data that are replicated from a master network server. |
| **WorldSoftware architecture** | The broad spectrum of application design and programming technology that PeopleSoft uses to achieve uniformity, consistency, and complete integration throughout its software. |
| **write payment** | A step in processing payments. Writing payments includes printing checks, drafts, and creating a bank tape table. |
| **write-off** | A method for getting rid of inconsequential differences between amounts. For example, you can apply a receipt to an invoice and write off the difference. You can write off both overpayments and underpayments. |
| **Z file** | For store and forward (network disconnected) user, EnterpriseOne store-and-forward applications perform edits on static data and other critical information that must be valid to process an order. After the initial edits are complete, EnterpriseOne stores the transactions in work tables on the workstation. These work table are called Z files. When a network connection is established, Z files are uploaded to the enterprise server; and the transactions are edited again by a master business function. The master business function then updates the records in your transaction files. |
| **z-process** | A process that converts inbound data from an external system into an EnterpriseOne software table or converts outbound data into an interface table for an external system to access. |

| | |
|---|---|
| **zusammenfassende melding** | In Germany, the term for the EU Sales Listing. |

# Index

## A



## B



## C

# D

## E



## F



## H

## I



## K



## L



## M

## N



## O



## P

# Q



# R



# S

## T



## U