# Design Overview

### Centralized Training Environment

The Centralized Training Environment imitates a client server environment. When utilizing the Centralized Training Environment, all projects and objects created in class must have a unique name. Students should replace all instances of *xx*, included in all project and object names throughout their course exercises, with the last two digits of their sign on and all instances of *y* with an alpha character assigned by the Instructor. This will ensure that all of their projects and objects are unique to the system. (i.e. TRN***08***101, R5509***08***COL, P550101***08***)

### Business Purpose

The purpose of class exercises is to provide practical, hands-on experience using development tools to create business applications. These exercises conform to the lecture topics, and each exercise is a building block in the application creation process.

In the directed exercises that follow, you will design an Address Book system. The directed exercises, which are instructor-led, usually followed by exercises that you do on your own. The business application developed during the exercises is the Item Master application, which is part of a typical inventory system. The final project in this class is a case study, in which you will apply the skills and concepts from the lecture and exercises. In the case study, you will develop a Purchase Order application that works with the Address Book system and the Item Master application.

### Overview of the Address Book Application

The Address Book is an online version of a traditional Rolodex card file. You use the Address Book to manage the information about the people and companies with whom you do business. An application is necessary to maintain the Address Book Master table.

As you design an Address Book system, the Address Book application will provide you with the ability to locate and browse through a list of customers, employees, and suppliers. It will also provide you with the ability to select and modify a record, add a new record, and delete a record.

### Overview of the Item Master Application

An Inventory system tracks all of the items that a business has in its inventory. Information about each item, such as item number, description, supplier, quantity in inventory, and reorder point, is usually maintained in a master file. An application is necessary to maintain the Item Master table.

As you design and build the Item Master application throughout the exercises, the application will provide you with the ability to locate and browse through a list of items, select an existing item and modify it, add a new item, and delete an item.

## Overview of the Purchase Order Application

A purchase order system is used to order inventory items. Inventory is replenished by creating a purchase order, which is then sent to a supplier. A purchase order typically includes information such as order number, order type, order date, requested ship date, supplier number, ship-to location, and a list of the individual items ordered.

In the case study, you will design a simple purchase order application. This application will provide you with the ability to locate and browse through a list of purchase orders, select and modify a purchase order, add a new purchase order, and delete a purchase order.

# Object Management Workbench

# Exercise: Adding Projects

### Business Purpose

All development in is done in the context of a project. All the objects needed for a given development process should be included in a separate project. Since you will be developing two applications in the exercises that follow, you need to create two projects to hold the objects that make up the Address Book application and the Item Master application.

### Objective

Using the following data, create two projects in Object Management Workbench. Your project descriptions and characteristics must be identical to the following data.

► **Steps to complete**

### Add a Project for the Address Book Application

1. In Solution Explorer, type OMW in the Fast Path field and press Enter to access Object Management Workbench (P98220).

2. On Object Management Workbench, click Add to create a project.

3. On Add J.D. Edwards Object to the Project, choose the OMW Project option and click OK.

4. On Project Revisions, enter the following information on the Summary tab:

| Field | Value | Comment |
|---|---|---|
| Project ID | TRN001xx | |
| Description | Class Address Book | |
| Type | 02 | Enhancement |
| Severity | 03 | Not Critical/No Workaround |

**Note**

The following step is performed for training purposes only. If you create custom objects you should use Product System Code 55 to 59.

5. Click the Visual Assist button for the Product System Code.

   a. In the Select User Defined Code form choose the Form menu and click Revisions

   b. Form the Work With User Defined Codes form click the Add button on the toolbar.

   c. Scroll to the last grid row in the User Defined Codes form.

d. Enter the following information into the last grid row.

| Codes | Description 1 | Description 2 | Special Handling | Hard Coded |
|---|---|---|---|---|
| 55xx | Student xx | | | N |

6. Complete the Summary tab with the following information.

| | | |
|---|---|---|
| Product System Code | 55xx | |
| Release | | Choose the Current Release |

7. Enter the following values on the Dates tab:

| Field | Value | Comment |
|---|---|---|
| Data Started | | The default date that the system supplies. |
| Planned Completion Date | | The date that is five days after Date Started date. |

Do not make any entries on the Category Codes tab. These codes are used to further categorize development and modification of objects.

8. On the Attachments tab, enter “This project was created to add the Class Address Book application.”

   a. Click OK

9. Add yourself as a Developer to the project.

10. Promote your project to status of 21.

### Add a Project for the Item Master Application

11. On Object Management Workbench, click Add to create a project.

12. Enter the following information on the Summary tab:

| Field | Value | Comment |
|---|---|---|
| Project ID | TRN002xx | |
| Description | Class Item Master | |
| Type | 02 | Enhancement |
| Severity | 03 | Not Critical/No Workaround |

**Note**

The following step is performed for training purposes only. If you create custom objects you should use Product System Code 55 to 59.

13. Click the Visual Assist button for the Product System Code.

    a. In the Select User Defined Code form choose the Form menu and click Revisions

    b. Form the Work With User Defined Codes form click the Add button on the toolbar.

    c. Scroll to the last grid row in the User Defined Codes form.

    d. Enter the following information into the last grid row.

| Codes | Description 1 | Description 2 | Special Handling | Hard Coded |
|---|---|---|---|---|
| 56xx | Student xx | | | N |

14. Complete the Summary tab with the following information.

| | | |
|---|---|---|
| Product System Code | 56xx | Reserved for Clients |
| Release | | Select the current release |

15. Enter the following values on the Dates tab:

| Field | Value | Comment |
|---|---|---|
| Date Started | | The default date that the system supplies |
| Planned Completion Date | | The date that is five days after Date Started date |

    Do not make any entries on the Category Codes tab. These codes are used to further categorize development and modification of objects.

16. On the Attachments tab, enter “This project was created to add the Item Master application.”

17. Add yourself as a Developer to the project.

18. Promote your project to status of 21.

## Exercise: Understanding Object Management Workbench

### Business Purpose

Complete these review questions to reinforce your understanding of the key concepts about Object Management Workbench.

### ▶ Steps to complete

1. You use Object Management Workbench to do the following:

2. What is the purpose of checking in an object?

3. Only one user can check out an object at a time. Are other users able to view the design of an object that they cannot check out?

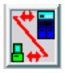
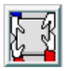
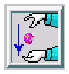
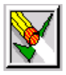
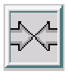
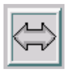
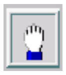
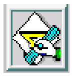
# Exercise: Identifying Action Buttons

### Business Purpose

In Object Management Workbench, the buttons in the center of the form are used to perform actions on a selected object. The actions you can perform for each object depend on your role and the status of the project in which the selected object resides. The action buttons are visual indicators of the actions you are allowed to perform on specific object types.

### ▶ Steps to complete

In the space provided, describe the action each button performs on an object. In addition, give an example of when you might perform each of these actions.

| | |
|---|---|
|  | |
|  | |
|  | |
|  | |
|  | |
|  | |
|  | |
|  | |

| | |
|---|---|
| 💾 | |
| 🔄 | |
| ⇦<br>⇨ | |

## User Defined Codes

## Directed Exercise: Adding User Defined Code 55xx|Ry

### Business Purpose

A user defined code is a predefined table of valid values that is attached to a data item in the data dictionary. The system validates the information that the user enters in a field with the values in user defined codes for the code type and code values that are associated with the data item. The software is delivered with many user defined codes already defined, but new codes can be added at any time depending on business needs.

The Class Address Book application requires a user defined code to indicate valid values for the region in which an employee, a supplier, or a customer is located. This exercise demonstrates how to add a new user defined code.

### Objective

Using the data supplied below, create a user defined code type and its associated values. Upon completion of this exercise, user defined code 55xx|Ry should appear under the Objects node of your Class Address Book project.

### ► Steps to complete

1. In Solution Explorer, type OMW in the Fast Path field to access Object Management Workbench (P98220).

2. In your Class Address Book project, create user defined code Type Ry for system 55xx.

   a. Choose the Objects node in the Class Address Book project and click Add on the toolbar.

      The Add OneWorld Objects to the Project form appears.

   b. Click the User Defined Code option in the Control Table Objects group box, and click OK.

   c. On the User Defined Code Types form, enter the following values:

**Note**

Please replace the **y** with the alpha character your instructor has assigned you.

| Field | Value | Comment |
|---|---|---|
| Product Code | 55xx | |
| User Defined Codes | Ry | |
| Description | Region | |
| Code Length | 2 | |
| 2nd Line | | Off |
| Numeric | | Off |

d. Click OK.

The User Defined Codes form appears

3. Set up the user defined code values for 55xx|Ry using the following data:

**Note**
Leave the Special Handling value blank for all records.

| Codes | Description | Special Handling | Hard Coded |
|---|---|---|---|
| CE | Central | | N |
| NE | Northeast | | N |
| NW | Northwest | | N |
| SE | Southeast | | N |
| SW | Southwest | | N |

4. Click OK.

The Object Management Workbench form appears.

55xx|Ry - Region now appears with a UDC button under the Objects node of your Class Address Book Project.

**Note**
The 55xx|Ry UDC code will be associated with data dictionary item SAAC01xx in the data dictionary exercises.

# Exercise: Adding User Defined Code 56xx|Py

### Business Purpose

A user defined code is a predefined table of valid values that is attached to a data item in the data dictionary. The system validates the information that the user enters in a field with the values in user defined codes for the code type and code values that are associated with the data item. The is delivered with many user defined codes already defined, but new codes can be added at any time depending on business needs.

The Class Item Master application requires a user defined code to indicate whether items in storage are perishable or non-perishable. This exercise demonstrates how to add a new UDC for this application.

### Objective

Using the data supplied below, create a user defined code Type and its associated values. Upon completion of this exercise, you should have user defined code 56xx|Py under the Objects node of your Item Master project.

### ► Steps to complete

1. In your Class Item Master project, create user defined code type Py for system 56xx.

**Note**

Replace the y with the alpha character your instructor assigned you.

| Field | Value | Comment |
|---|---|---|
| Product Code | 56xx | |
| User Defined Codes | Py | |
| Description | Storage Type | |
| Code Length | 1 | |
| 2nd Line | | Off |
| Numeric | | Off |

2. Set up the following user defined code values for 56xx|Py:

**Note**
Leave the Special Handling value blank (null) for all records.

| Codes | Description | Special Handling | Hard Coded |
|---|---|---|---|
| P | Perishable | | N |
| N | Non-perishable | | N |

3. User defined codes provide a structure to ensure the following:

____________________________________________________________

____________________________________________________________

____________________________________________________________

4. Explain why you would create a custom list of user defined code values:

____________________________________________________________

____________________________________________________________

____________________________________________________________

____________________________________________________________

**Note**

The 56xx|Py UDC code will be associated with data dictionary item SASTRITMxx in the data dictionary exercises.

## Next Numbers

## Directed Exercise: Adding a Next Numbers Table

### Business Purpose

When you enter a record, such as a new customer in the Class Address Book or a new item in the Class Item Master table, you can either manually assign a number to the record or let the Next Numbers program assign one for you. The Next Numbers program allows the system to find the next available number in the Next Numbers table (F0002) and assigns that number to the new record.

Using Next numbers assures that the records in a table start with a predefined number and then increment sequentially. You can also set up a next numbers table to assign check digits for any set of standard next numbers. Check digits prevent errors caused by transposition during data entry. For example, activating check digits in the Class Address Book for suppliers prevents a voucher from being assigned to the wrong supplier if digits are transposed during voucher entry.

It is recommends that you use check digits for next numbers only if a transposition during data entry is likely to create errors.

Next numbers are assigned from an array. The combination of system code and index defines how the next number will be assigned.

### Objective

Using the following data, create two next numbers tables and indices. One index is for the address number in the Class Address Book application, and the other index is for the item number in the Class Item Master application. Your tables should look like those shown in this exercise.

### ▶ Steps to complete

Follow along with the instructor to create the next numbers tables and their corresponding indices.

1. From Solution Explorer, type NN in the Fast Path field to access the Next Numbers application.

2. On Work With Next Numbers, verify that no entries exist for system codes 55xx and 56xx.

3. For the system to automatically assign a number for each Address Book record that you enter, you need to create a table for system 55xx and then add Address Number as an index in that table.

4. To add an entry for system 55xx, click Add on the toolbar.

5. On Set Up Next Numbers by System, complete the form as shown:

   **System**: 55xx

   **Use**: Address

   **Next Number**: 300

   **Check Digit Used**: check

6. Click OK to return to the Work With Next Numbers form.

**Note**
Next Numbers are assigned from an array. The combination of system code and index defines how the next number will be assigned. Address Number is Index 1 for System Code 55xx. You will need to know this number when you override the next number index to data dictionary item SAAddressNumberAxx (SAAN8Axx).

7. For the system to automatically assign a number for each inventory item that you enter, you need to create a table for system 56xx and then add Item Number as an index in that table.

8. To add an entry for system 56xx, click Add on the toolbar.

9. On Set Up Next Numbers by System, complete the form as shown:

   **System**: 56xx

   **Use**: Item Number

   **Next Number**: 900

   **Check Digit Used**: no check

10. Click OK to return to the Work With Next Numbers form, and then click Close to exit the Next Numbers application.

**Note**

Use the System 56xx Item Number index for dictionary item SAItemNumber (SAITM) in your Item Master application. Instead of changing the next number for SAITM in the data dictionary, you will override the next number index at the application level. The approach you take when designing applications depends on whether a data dictionary item is used in two or more applications that require different numbers.

**Caution**

Do not change any next numbers after you start using the applications. Changing the numbers can result in duplicates, as well as the inability to locate previously-added numbers.

Do not change the sequence of the next numbers in the table. Each next number must remain on its current line because programs reference a specific line in the table.

# Data Dictionary

# Exercise: Working with Data Items

### Business Purpose

The data dictionary is an application used to create and modify data items that are used in all applications.

Before you create new data items for an application, you must determine whether an existing data item meets your specifications. Creating duplicate data items could eventually affect system performance and lead to unnecessary maintenance.

You can modify a data item to meet your needs; however, other applications might be affected by your modifications.

### Objective

Upon completion of this exercise, you will be able to do the following:

- Verify that the data dictionary has the data items that you need to create your Class Address Book application (P5501xx) and your Class Item Master application (P5601xx).
- Modify a data dictionary data item for use in your applications.
- Create a new data dictionary data item.

### ► Steps to complete

1. Access the Data Dictionary from Solution Explorer.

2. Copy each of the following data items and rename it:

| **OLD Name** | | **NEW Name** | |
|---|---|---|---|
| **Data Item** | **Alias** | **Data Item** | **Alias** |
| SASalary | SASAL | SASalaryxx | SASALxx |
| SAAddressNumberA | SAAN8A | SAAddressNumberAxx | SAAN8Axx |
| SARegion | SAAC01 | SARegionxx | SAAC01xx |
| SAItemStorage | SASTRITM | SAItemStoragexx | SASTRITMxx |

**Note**
The copied and renamed data items will appear in the Default project.

Column names beyond 10 characters (an 8 character Alias plus a 2 character Column Prefix) may not be supported in all Databases.

3. If the specifications for a these data dictionary items do not match those in the following tables, move that item into either the Class Address Book project or the Class Item Master project, depending on which one it affects.

4. Modify the data item using values from the table.

5. If you do not change the characteristics of a data item, remove it from your default project by clicking the right-pointing arrow action button.

**Caution**
Do not use the Delete option on the toolbar to remove the item from your project.

### Class Address Book Application (P5501xx)

6. Before you can create the P5501xx application, you must have a data item that contains the Monthly Salary for an employee. The specifications follow:

### Item Specifications

| Field | Value | Comment |
|---|---|---|
| Data Item | SASalaryxx | |
| Alias | SASALxx | |
| Glossary Group | D | Primary Data Element |
| Item Parent | Blank | |
| Description | Monthly Salary | |
| Product Code | 98SA | |
| Data Type | 9 | Numeric |
| Size | 15 | |
| File Decimals | Blank | |
| Class | CURRENCY | |
| Display Decimals | 2 | |
| Control Type | 4 | Generic Edit |
| Item Occurrences | Blank | |
| Row Description | Monthly Salary | |
| Column Title | Monthly<br>Salary | |
| Upper Case Only | | Off |
| Row Security | | Off |
| Allow Blank Entry | | On |
| Auto Include | | Off |
| Do Not Total | | Off |

### Item Glossary

| |
|---|
| If the glossary text does not exist, enter a meaningful definition for this data item. |

### Default Value

| |
|---|
| No default value |

### Visual Assist

| |
|---|
| Calculator |

### Edit Rule

| |
|---|
| No edit rule |

### Display Rule

| Field | Value |
|---|---|
| Rule | CODE<br>K |

### Next Number

| |
|---|
| No Next Numbering |

7. The Address Book System requires a data item that contains the address number. Verify that this data item exists and has the following specifications:

### Item Specifications

| Field | Data | Comment |
|---|---|---|
| Data Item | SAAddressNumberAxx | |
| Alias | SAAN8Axx | |
| Glossary Group | D | Primary Data Element |
| Item Parent | Blank | |
| Description | Address Number | |
| Product Code | 98SA | |
| Data Type | 9 | Numeric |
| Size | 8 | |
| File Decimals | Blank | |
| Class | Blank | |
| Display Decimals | 0 | |

| Field | Data | Comment |
|---|---|---|
| Control Type | 4 | Generic Edit |
| Item Occurrences | Blank | |
| Row Description | Address Number | |
| Column Title | Address<br>Number | |
| Upper Case Only | | Off |
| Row Security | | Off |
| Allow Blank Entry | | Off |
| Auto Include | | Off |
| Do Not Total | | Off |

### Item Glossary

| |
|---|
| If the glossary text does not exist, enter a meaningful definition for this data item. |

### Default Value

| |
|---|
| No default value |

### Visual Assist

| |
|---|
| No visual assist |

### Edit Rule

| |
|---|
| No edit rule |

### Display Rule

| Field | Value |
|---|---|
| Rule | CODE<br>Z |

### Next Number

| Field | Value | Comment |
|---|---|---|
| Row Security | 55xx<br>1 | System code<br>Index |

8. For data item SARegionxx (SAAC01xx), change the data item Edit Rule to use the UDC type and code that you created in the UDC directed exercise. Move this data item to the Objects node of your Class Address Book project.

### Edit Rule

| Field | Value | Comment |
|---|---|---|
| Rule | UDC<br>55xx<br>Ry | <br>Product Code<br>Record Type |

### Class Item Master Application (P5601xx)

9. Before you can create the P5601xx application, you must have a data item for a code that specifies whether an inventory item is perishable or non-perishable. The specifications follow:

### Item Specifications

| Field | Data | Comment |
|---|---|---|
| Data Item | SAItemStoragexx | |
| Alias | SASTRITM | |
| Glossary Group | D | Primary Data Element |
| Item Parent | | Blank |
| Description | Item Storage | |
| Product Code | 98SA | |
| Data Type | 1 | Character |
| Size | 1 | |
| File Decimals | Blank | |
| Class | Blank | |
| Display Decimals | 0 | |
| Control Type | 1 | User Defined Code Edit |
| Item Occurrences | | Blank |
| Row Description | Item Storage | |
| Column Title | Item<br>Storage | |
| Upper Case Only | | On |
| Row Security | | Off |
| Allow Blank Entry | | Off |
| Auto Include | | Off |
| Do Not Total | | Not Applicable |

### Item Glossary

If the glossary text does not exist, enter a meaningful definition for this data item.

### Default Value

No default value

### Visual Assist

| Field | Value | Comment |
|---|---|---|
| Search Form | Select User Define Code | W0005SB |

### Edit Rule

| Field | Value | Comment |
|---|---|---|
| Rule | UDC<br>56xx<br>Py | |

### Display Rule

No display rule

### Next Number

No next numbering

# Exercise: Creating a Data Item

### ▶ Steps to complete

The Address Book System requires a data item that will be used to indicate whether an employee, supplier, or customer is considered active or inactive. You must create this data dictionary item.

1. In the Class Address Book project, create a data item using the following specifications:

**Item Specifications**

| Field | Data | Comment |
|---|---|---|
| Data Item | SAAddressBkStatusxx | |
| Alias | SASTATxx | |
| Glossary Group | D | Primary Data Element |
| Item Parent | | Blank |
| Description | Status Active/Inactive Flag | |
| Product Code | 55xx | |
| Product Code Reporting | 55xx | |
| Data Type | 1 | Character |
| Size | 1 | |
| File Decimals | | Blank |
| Class | | Blank |
| Display Decimals | 0 | |
| Control Type | 4 | Generic Edit |
| Item Occurrences | | Blank |
| Row Description | Status | |
| Column Title | Status | |
| Upper Case Only | | On |
| Row Security | | Off |
| Allow Blank Entry | | Off |
| Auto Include | | Off |
| Do Not Total | | Not Applicable |

**Item Glossary**

This field serves as a flag indicating whether an employee, customer or vendor is considered active or inactive.

**Default Value**

No default value

**Visual Assist**

No visual assist

**Edit Rule**

No edit rule

**Display Rule**

No display rule

**Next Number**

No next numbering

2. Click OK to save your new data dictionary item.

# Exercise: Searching the Data Dictionary

### Objective

Upon completion of this exercise, you will be able to search for data items using the Work With Data Dictionary Items selection on the Data Dictionary Design menu.

### ► Steps to complete

Several data items that you need to build the Item Master table already exist in the data dictionary.

1. From Solution Explorer, type DD in the Fast Path field to access the Data Dictionary Design menu, and then choose Work With Data Dictionary Items.

2. Search the data dictionary for each alias in the following table.

3. After you verify that an alias exists, use the table to enter its data item name.

| | Alias | Data Item |
|---|---|---|
| 1 | SAITM | |
| 2 | SADSC1 | |
| 3 | SADSC2 | |
| 4 | SAAN8Axx | |
| 5 | SATYPE | |
| 6 | SAPREQ | |
| 7 | SAUOM1 | |
| 8 | SAPRRC | |
| 9 | SAPQOH | |
| 10 | SAROP | |
| 11 | SASTRITMxx | |
| 12 | USER | |
| 13 | PID | |
| 14 | UPMJ | |
| 15 | JOBN | |
| 16 | UPMT | |

## Exercise: Understanding the Data Dictionary

Complete these review questions to reinforce your understanding of the key concepts presented in the data dictionary exercises.

1. Name three advantages that the data dictionary provides to the application design process.

2. What are the names and location of the local data dictionary specification tables?

3. What is the function of the Check Digit Used option on the Next Numbers form? How does turning on Check Digit Used affect sequential numbering?

## Table Design

# Directed Exercise: Creating the Class Address Book Table – F5501xx

### Business Purpose

You do not always need to create a new table for each new application or form. Sometimes you can use an existing table, as long as it has the data that your application needs. The Class Address Book application that you are designing does require a new table, so you need to create one using Object Management Workbench.

### Objective

Upon completion of this exercise, you will be able to do the following:

- Use Object Management Workbench to create the F5501xx table.
- Define the F5501xx table using Table Design Aid.
- Use naming standards.
- Discuss audit trail fields.
- Search for data items using the QBE line in the Data Dictionary Browser window.
- Add data items to the Columns view.
- Explain the function of the four Table Design views: Columns, Indices, Data Dictionary Browser, and Properties.
- Create and name a primary unique index for the F5501xx table.
- Generate the table to create a physical table in which to store the data.
- Understand the difference between generating a table and generating an index.

### ► Steps to complete

1. In Solution Explorer, type OMW in the Fast Path field to access Object Management Workbench (P98220).

2. Add a Table object to the Class Address Book project.

   a. Choose the Objects node of your project.

   b. Click Add on the toolbar.

   c. Click the Table option in the Object Librarian Objects group box, and then click OK.

d. Enter the following values in the Add Objects window and click OK:

| Field | Value |
|---|---|
| Object Name | F5501xx |
| Description | Class Address Book Master |
| Product Code | 55xx |
| Product System Code | 55xx |
| Object Use | 2 |
| Column Prefix | AB |

3. On the Design Tools tab, click Start Table Design Aid.

4. Use the Data Dictionary Browser in the Table Design Aid to select these data items from the data dictionary by either double-clicking or dragging each of the data items from the Data Dictionary Browser window to the Columns window.

|  | Alias | Data Item |
|---|---|---|
| 1 | SAAN8Axx | SAAddressNumberAxx |
| 2 | SAAT1 | SASearchType |
| 3 | SAALPH | SAAlphaName |
| 4 | SAADD2 | SAAddressLine2 |
| 5 | SAADD3 | SAAddressLine3 |
| 6 | SACTY1 | SACity |
| 7 | SAADDS | SAState |
| 8 | SAADDZ | SAPostalCode |
| 9 | SAStatxx | SAAddressBookStatusxx |
| 10 | SACTR | SACountry |
| 11 | SAAC01xx | SARegionxx |
| 12 | SASALxx | SASalaryxx |
| 13 | USER | UserId |
| 14 | PID | ProgramId |
| 15 | UPMJ | DateUpdated |
| 16 | JOBN | WorkStationId |
| 17 | UPMT | TimeLastUpdated |

**Note**
To provide an audit trail, Standards require that you include USER, PID, UPMJ, JOBN, and UPMT in every transaction table. Because these fields are included, you can determine the user name associated with each record in a table, the program that created the entry, the date on which the entry was made or updated, the workstation that is associated with the entry, and the time that the entry was made.

5. Create a primary unique index for the table.

   a. Choose SAAddressNumberAxx, drag it to the Indices window and drop it onto the Class Address Book Master icon.

      When you create the index, it has the name Untitled.

   b. Rename the index as Address Number.

      Focus on new index and click once on the text Untitled to enter edit mode.

      The index is marked with the letter P to indicate that it is the primary index.

6. Save your table and Exit from Table Design Aid.

7. Generate the table so that you will have a physical table in which to store the data for your Class Address Book application.

   a. To generate the table, click the Table Operations tab and then select Generate Table.

      Verify that the data source is *Business Data - TEST* and the Password is the same as the Object Owner Id.

   b. click OK.

      **Caution**
      Do not generate the table if you have only created a new index or added additional keys to an existing index. When you generate a table, any existing data in the table is deleted.

      If you create a new index or add additional keys to an existing index, you must generate the index. This process modifies the .h file, but you will not lose existing data.

**Note**
When you generate a table, the software creates both the physical table and the index in the database. Generating an index creates only the physical index in the database.

After a table is defined in the software, it can be generated in multiple databases.

From the File menu in table design you can print the table design. This is useful to view the data items, alias, and primary keys.

8. After you generate the table, click OK to exit Table Design.

   The F5501xx – Class Address Book Master table appears as an object under the Class Address Book project.

# Directed Exercise: Creating the Class Phones Table – F5515xx

### Business Purpose

Each record in the Class Address Book can have more than one phone number associated with it. For example, a supplier probably has one or more business numbers, a fax number, and possibly a cell phone number. A customer might have a business number, a home number, a cell phone number, and so on. Because the number of phone numbers required for each record in the Class Address Book is unknown, create a separate table just for the phone information. Using a separate table, each record in the Class Address Book can have as many phone numbers as necessary.

### Objective

Upon completion of this exercise, you will be able to do the following:

- Use Object Management Workbench to create the F5515xx table.
- Define the F5515xx table using Table Design Aid.
- Search for data items using the QBE line in the Data Dictionary Browser window.
- Add data items to the Columns view.
- Explain the function of the four Table Design views (Columns, Indices, Data Dictionary Browser, and Properties).
- Create and name a primary unique index for the F5515xx table.
- Generate the table to create a physical table in which to store the data.
- Understand the difference between generating a table and generating an index.

### ► Steps to complete

1. In Solution Explorer, type OMW in the Fast Path field to access Object Management Workbench (P98220).

2. Add a Table object to the Class Address Book project.

    a. Choose the Objects node of your project.

    b. Click Add on the toolbar.

    c. Click the Table option in the Object Librarian Objects group box, and then click OK.

    d. Enter the following values in the Add Objects window, and then click OK:

| Field | Value |
|---|---|
| Object Name | F5515xx |
| Description | Class Phones |
| Product Code | 55xx |
| Product System Code | 55xx |

| Field | Value |
|---|---|
| Object Use | 2 |
| Column Prefix | PH |

3. On the Design Tools tab, click Start Table Design Aid.

4. Use the Data Dictionary Browser in the Table Design Aid to choose the following data items from the data dictionary by either double-clicking or dragging each of the data items from the Data Dictionary Browser window to the Columns window.

| | Alias | Data Item |
|---|---|---|
| 1 | SAAN8Axx | SAAddressNumberAxx |
| 2 | SALNID | SALineNumber |
| 3 | SAAR1 | SAAreaCode |
| 4 | SAPH1 | SAPhoneNumber |
| 5 | SAPHTP | SAPhoneNumberType |
| 6 | SAALPH | SANameAlpha |
| 7 | USER | UserId |
| 8 | PID | ProgramId |
| 9 | UPMJ | DateUpdated |
| 10 | JOBN | WorkStationId |
| 11 | UPMT | TimeLastUpdated |

5. Create a Primary unique index for the table.

   a. Choose SAAddressNumberAxx, drag it to the Indices window, and drop it onto the Class Phones icon.

   b. Click *Yes* to add a new index using this column.

   c. Click and drag SALineNumber and drop it onto the *Untitled* index in the Indices window.

      When you create the index, it has the name *Untitled*.

   d. Rename the index Addr Nbr, Line Id.

      Focus on the new index and click once on the text *Untitled* to enter edit mode.

      The index is marked with the letter P to indicate that it is the primary index.

**Note**
The total length of the index name cannot exceed 19 characters if the index has two or more fields. If you exceed 19 characters, the compiler displays the warning "Redefinition is not identical …." The length of the index name affects fetches that use the wrong index ID in business functions.

6. Save your table and exit Table Design Aid.

7. Generate the table so that you will have a physical table in which to store the data for your Class Address Book application.

   a. Verify that the data source is *Business Data - TEST* and the Password is the same as the Object Owner Id.

   b. click OK.

**Caution**
Do not generate the table if you have only created a new index or added additional keys to an existing index. When you generate a table, any existing data in the table is deleted.

If you create a new index or add additional keys to an existing index, you must generate the index. This process modifies the .h file, but you will not lose existing data.

**Note**
When you generate a table, the software creates both the physical table and the index in the database. Generating an index creates only the physical index in the database.

After a table is defined in the software, it can be generated in multiple databases.

8. After you generate the table, click OK to exit Table Design.

   The F5515xx – Class Phones table appears as an object under the Class Address Book project.

# Exercise: Creating the Item Master Table – F5601xx

### Business Purpose

Selecting data items from the data dictionary is the first step in developing the Item Master application P5601xx. The next step is designing and creating the Item Master table F5601xx.

A table stores sets of data in columns and rows. Each column is a data item. Each row is a record. You can create one or more tables for use in an application.

You will use the Table Design Aid tool to combine data items and indexes into a table definition for the Item Master table F5601xx. This table stores the data that your Class Item Master application P5601xx uses; therefore, you should add this new table to the Objects node of your Class Item Master project.

### Objective

Upon completion of this exercise, you will be able to do the following:

- Use Object Management Workbench to create the F5601xx table.
- Define the F5601xx table using Table Design Aid.
- Search for data items using the QBE line in the Data Dictionary Browser window.
- Add data items to the Columns view.
- Explain the function of the four Table Design views: Columns, Indices, Data Dictionary Browser, and Properties.
- Create and name a primary unique index for the F5601xx table.
- Generate the table to create a physical table in which to store the data.
- Understand the difference between generating a table and generating an index.

### ► Steps to complete

1. Using the following information, add a table object to the Class Item Master project:

| Field | Value |
|---|---|
| Object Name | F5601xx |
| Description | Class Item Master |
| Product Code | 56xx |
| Product System Code | 56xx |
| Object Use | 2 |
| Column Prefix | IT |

2. Design the table using the data dictionary item aliases listed in the following table:

| | Alias | Data Item |
|---|---|---|
| 1 | SAITM | SAItemNumber |
| 2 | SADSC1 | SADescription |

| | Alias | Data Item |
|---|---|---|
| 3 | SADSC2 | SACatalogDescription |
| 4 | SAAN8Axx | SAAddressNumberAxx |
| 5 | SATYPE | SAItemType |
| 6 | SAPREQ | SAQuantityOnOrder |
| 7 | SAUOM1 | SAUnitofMeasure |
| 8 | SAPRRC | SAAmountUnit |
| 9 | SAPQOH | SAQuantityOnHand |
| 10 | SAROP | SAReorderPoint |
| 11 | SASTRITMxx | SAItemStoragexx |
| 12 | USER | UserId |
| 13 | PID | ProgramId |
| 14 | UPMJ | DateUpdated |
| 15 | JOBN | WorkStationId |
| 16 | UPMT | TimeLastUpdated |

3. Create a primary unique index for the F5601xx table using SAItemNumber as the primary unique identifier.

   Index name = Item Number

   **Caution**
   Do not generate the table if you have only created a new index or added additional keys to an existing index. When you generate a table, any existing data in the table is deleted.

   If you create a new index or add additional keys to an existing index, you must generate the index. This process modifies the .h file, but you will not lose existing data.

4. Generate the table so that you will have a physical table in which to store the data for your Class Item Master application.

   Verify that the data source is *Business Data - TEST* and the password is the same as the Object Owner Id.

**Note**
When you generate a table, the software creates both the physical table and the index in the database. Generating an index creates only the physical index in the database.

5. After you generate the table, click OK to exit Table Design.

   The F5601xx – Class Item Master table appears as an object under the Class Item Master project.

# Universal Table Browser

## Exercise: Using the Universal Table Browser

### Business Purpose

Universal Table Browser is a tool that allows you to view tables in different databases. This tool lets you verify the existence of a table, view data in a table, and determine the structure of a table.

### Objective

Upon completion of this exercise, you will be able to do the following:

- Access the Universal Table Browser from menu GH902.
- Use the Fast Path UTB to access the Universal Table Browser.
- Locate tables in the Universal Table Browser.

### ► Steps to complete

Use the Universal Table Browser to locate the tables that you created in the previous exercises: F5501xx, F5515xx, and F5601xx.

1. From Solution Explorer, access the Universal Table Browser by typing UTB in the Fast Path field.
2. On Universal Table Browser, choose Open Table from the File menu.
3. Type F5501xx in the Table field.
4. For the data source name, click the Visual Assist button and choose *Business Data – TEST.*.
5. Click OK, and the F5501xx table appears.
6. Repeat the previous steps for tables F5515xx and F5601xx.

# Business View Design

# Directed Exercise: Business View – V5501Axx

### Business Purpose

After you define, create, and generate a physical table, you must design a business view that is based on the table definition that you created. A business view is an object that you create using Object Management Workbench.

You use a business view to define the data elements from a table or from multiple tables. You use the Business View Design Aid to select only the data items that are required for your Class Address Book application (P5501xx).

The software uses the business view to generate the SQL statements that the P5501xx application needs to retrieve data from supported databases.

### Objective

Upon completion of this exercise, you will be able to do the following:

- Use Object Management Workbench to create the V5501Axx business view in the Class Item Master project..
- Use the Business View Design tool to select data items required for your P5501xx application.
- Discuss how a business view links an application to one or more tables.

### ▶ Steps to complete

Follow along with the instructor to create the V5501Axx business view.

1. In Solution Explorer, type OMW in the Fast Path field to access Object Management Workbench (P98220).

2. Add a Business View object to the Class Address Book project.

    a. Choose the Objects node of your project.

    b. Click Add on the toolbar.

    c. Click the Business View option in the Object Librarian Objects group box, and then click OK.

3. Enter the following values in the Add Objects window:

| Field | Value |
|---|---|
| Object Name | V5501Axx |
| Description | Class Address Book -- All Columns |
| Product Code | 55xx |
| Product System Code | 55xx |
| Object Use | 300 |

4. Click OK.

5. On the Design Tools tab, click Start the Business View Design Aid.

6. Enter F5501xx in the QBE Object Name field in the Available Tables window.

7. Drag the F5501xx table to the Table Joins window.

8. Focus on the table in the Table Joins window and click Select All Columns from the Table menu.

9. Save the business view and then Exit the Business View Design Aid.

10. Click OK to return to Object Management Workbench.

    The V5501Axx – Class Address Book – All Columns business view appears as an object under the Class Address Book project.

## Directed Exercise: Business View – V5501Sxx

### Business Purpose

After you have defined, created, and generated a physical table, you must design a business view that is based on the table definition that you created. A business view is an object that you create using Object Management Workbench.

You use a business view to define the data elements from a table or from multiple tables. You use the Business View Design Aid to select only the data items that are required for an application. You are creating the V5501Sxx business view for a Class Address Book Search and Select form. This type of form usually requires a separate business view for performance reasons.

The software uses the business view to generate the SQL statements that an application needs to retrieve data from the supported databases.

### Objective

Upon completion of this exercise, you will be able to do the following:

- Use Object Management Workbench to create the V5501Sxx business view.
- Use the Business View Design tool to select data items required for your application.
- Discuss how a business view links an application to one or more tables.

**Note**

The letter at the end of the business view name indicates the particular view that you are using among the many views that you might have created. For example, V5501Axx is the first view over the F5501xx table, and V5501Bxx is the second view. In this exercise, you use an S (V5501Sxx) to easily identify this business view as one that is specifically designed for a Search and Select form.

### ► Steps to complete

Follow along with the instructor to create the V5501Sxx business view.

1. In Solution Explorer, type OMW in the Fast Path field to access Object Management Workbench (P98220).

2. Add a business view object to the Class Address Book project.

   a. Choose the Objects node of your project.

   b. Click Add on the toolbar.

   c. Click the Business View option in the Object Librarian Objects group box, and then click OK.

3. Enter the following values in the Add Objects window:

| Field | Value |
|---|---|
| Object Name | V5501Sxx |
| Description | Class Address Book Search & Select |
| Product Code | 55xx |
| Product System Code | 55xx |
| Object Use | 300 |

4. Click OK.

5. On the Design Tools tab, click Start the Business View Design Aid.

6. Enter F5501xx in the QBE Object Name field in the Available Tables window.

7. Drag the F5501xx table to the Table Joins window.

8. Choose the Table Joins window and then double-click data items SASearchType (SAAT1) and SAAlphaName (SAALPH) to move them to the Selected Columns window.

**Note**
You should have the following three data items listed in the Selected Columns window:

| | | |
|---|---|---|
| SAAN8Axx | Address Number | F5501xx |
| SAAT1 | Sample Application - Search Type | F5501xx |
| SAALPH | Sample Application - Alpha Name | F5501xx |

9. Save the business view and then Exit the Business View Design Aid.

10. Click OK to return to Object Management Workbench.

    The V5501Sxx– Class Address Book Search and Select business view appears as an object under the Class Address Book project.

## Directed Exercise: Business View – V5515Axx

### Business Purpose

After you have defined, created and generated a physical table, you must design a business view that is based on the table definition that you created. A business view is an object that you create using Object Management Workbench.

You use a business view to define the data elements from a table or from multiple tables. You use the Business View Design Aid to select only the data items that are required for your Class Address Book application P5501xx.

The software uses the business view to generate the SQL statements that the P5501xx application needs to retrieve data from the supported databases.

### Objective

Upon completion of this exercise, you will be able to do the following:

- Use Object Management Workbench to create the V5515Axx business view.
- Use the Business View Design tool to select data items that are required for your P5501xx application.
- Discuss how a business view links an application to one or more tables.

### ► Steps to complete

Follow along with the instructor to create the V5515Axx business view.

1. In Solution Explorer, type OMW in the Fast Path field to access Object Management Workbench (P98220).
2. Add a business view object to the Class Address Book project.
   a. Choose the Objects node of your project.
   b. Click Add on the toolbar.
   c. Click the Business View option in the Object Librarian Objects group box, and click OK.
3. Enter the following values in the Add Objects window:

| Field | Value |
|---|---|
| Object Name | V5515Axx |
| Description | Class Phones -- All Columns |
| Product Code | 55xx |
| Product System Code | 55xx |
| Object Use | 300 |

4. Click OK.

5. On the Design Tools tab, click Start the Business View Design Aid.

6. Enter F5515xx in the QBE Object Name field in the Available Tables window.

7. Drag the F5501xx table to the Table Joins window.

8. Focus on the table in the Table Joins window and click Select All Columns from the Table menu.

9. Save the business view and then Exit the Business View Design Aid.

10. Click OK to return to Object Management Workbench.

    The V5515Axx – Class Phones – All Columns business view appears as an object under the Class Address Book project.

# Exercise: Business View – V5601Axx

### Business Purpose

To design a useful business view, you must carefully consider the data items that you need for your application. Create a business view for your Class Item Master application to reinforce what you have learned about business views.

### Objective

Upon completion of this exercise, you will be able to do the following:

- Use Object Management Workbench to create the V5601Axx business view.
- Use the Business View Design tool to select data items required for your P5601xx application.
- Discuss how a Business view links an application to one or more tables.

► **Steps to complete**

1. Use Object Management Workbench (Business View Design Aid) to create a business view for the Class Item Master table (F5601xx) for all columns.

| Field | Value | Comment |
|---|---|---|
| Object Name | V5601Axx | |
| Description | Class Item Master -- All Columns | |
| Product Code | 56xx | |
| Product System Code | 56xx | |
| Object Use | 300 | |
| Table | F5601xx | |
| Columns | | All |

2. Explain the performance issues that must be considered when you design business views.

# Forms Design

## Directed Exercise: Designing a Find/Browse Form (P5501xx)

### Business Purpose

After you have created the table and business view objects, you will use Object Management Workbench to define an application object and begin your application design.

You use the Application Design Aid to combine forms, controls, and event logic into applications. The tool provides you with a workspace for your form and the grid within this form. In this directed exercise, you will create the Find Browse form for your Class Address Book application (P5501xx).

### Objective

Upon completion of this exercise, you will be able to do the following:

- Create an application
- Choose a Find/Browse form type
- Attach a business view to the form
- Choose form properties and form options
- Choose the form and the grid
- Add form and grid fields
- Move and size controls
- Add filter fields
- Add menu and toolbar exits
- Test a form

### ▶ Steps to complete

Your instructor will complete this exercise with you.

1. Add an interactive application object to the Class Address Book project. The name and descriptive information are as follows:

| Field | Value |
|---|---|
| Object Name | P5501xx |
| Description | Class Address Book |
| Product Code | 55xx |
| Product System Code | 55xx |
| Object Use | 1 |

2. Click OK.

3. Click the Design Tools tab, and then click Start Form Design Aid.

4. On the Form Design Aid, choose the Form menu click Create and then Find/Browse.

5. On Find/Browse Form Properties, type Work With Addresses in the Title field and click OK.

6. Choose the form background and link a business view that is associated with the Class Address Book application to the form.

   The business view defines the columns that you want to choose from the F5501xx table.

   a. From the Form menu, Business View, choose *Add Business View…*

   b. On Individual Object Search and Select, type V5501Axx for the Object Name, and then click Find.

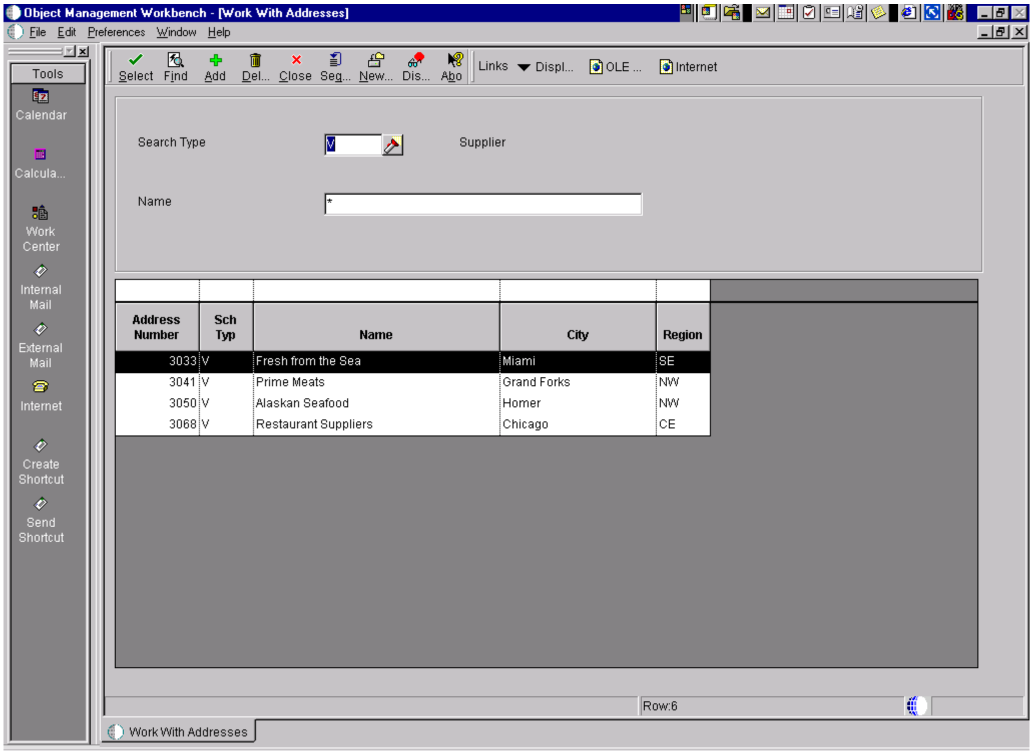
   c. Choose V5501Axx when it appears in the grid and click Select.

7. Add database elements to the form by inserting business view fields.

   a. Add Search Type and Alpha Name as form controls by selecting Business View Columns Browser… from the View menu.

   b. Place these controls on the form, as shown in the picture that follows.

Work With Addresses (Find/Browse) Form:



Search Type V Supplier

Name *

| Address Number | Sch Typ | Name | City | Region |
|---|---|---|---|---|
| 3033 | V | Fresh from the Sea | Miami | SE |
| 3041 | V | Prime Meats | Grand Forks | NW |
| 3050 | V | Alaskan Seafood | Homer | NW |
| 3068 | V | Restaurant Suppliers | Chicago | CE |

Row:6

Work With Addresses

8. Change the Alpha Name label to Name by right-clicking the label and choosing Properties.

   a. On Static Text Properties, click the Override Text option to turn it on, and then change the name.

9. Change filter criteria for the two filter controls.

    a. Choose the Search Type control.

    b. Right-click and choose Properties to access the Edit Properties window.

    c. Choose Filter, set the Filter Criteria to =, and set the Wildcard Display to Yes.

    d. Choose the Name control.

    e. Right-click and choose Properties to access the Edit Properties window.

    f. Choose Filter, set the Filter Criteria to >=, and set the Wildcard Display to Yes.

10. Align the Search Type and Name controls.

    a. Choose both controls by using the mouse pointer and the control key.

    b. Choose the control farthest to the left.

    c. From the Layout Menu, choose Align and then Left.

11. Add the Associated Description to the form.

    a. Choose the Search Type control.

    b. From the Edit menu, choose Associate Description.

    c. Place the control to the right of the Search Type field.

12. Add the grid columns to the grid in the order that they appear in the diagram, left to right.

    a. Choose the grid.

    b. From the View menu, choose Business View Columns Browser….

    c. Choose each data item and click the Insert Column button.

    d. After you have selected all items, close the Business View Columns Browser.

    e. Choose Column Sort Order in the Property Browser and sort the grid by Address Number, and then Click OK.

    **Note**
    Pressing the Control key and the Left or Right arrow keys will move the selected grid column.

13. Increase the width of the grid so that it matches the form illustration.

14. Change grid columns: Alpha Name to Name, City City to City, and Reg to Region.

    a. Double-click or right-click each column to be changed.

    b. On Grid Column Properties, click Override Text, and then change the name.

    c. Click OK.

15. Create a group box around Search Type and Name.

    a. Draw a box around the Search Type and Name fields.

    b. From the Insert menu, choose Group Box.

    c. Remove the name of the group box by double- or right-clicking the box to access Group Box Properties.

    d. Size the group box to match the form illustration.

16. Add Delete and Add buttons to the Tool Bar.

    a. From the Form menu, choose Menu/Toolbar Exits.

    b. On Menu Exits, choose Insert.

    c. On Menu Exit Properties, choose Delete from the Class drop-down menu, and then click OK.

    d. Repeat steps b and c for the Add button.

    e. On Menu Exits, click Close.

17. Set the Tab Sequence.

    a. From the Form menu, choose Tab Sequence.

    b. Click each control in the order in which you want the user to move through the form.

    c. After the sequence is defined, choose Tab Sequence from the Form menu to turn it off.

18. Test the form.

19. Save the form, and then exit.

20. On Interactive Application Design, click OK to return to Object Management Workbench.

    The P5501xx Class Address Book application appears in the Class Address Book project.

## Exercise: Designing a Find/Browse Form (P5601xx)

### Objective

Upon completion of this exercise, you will be able to do the following:

- Apply what you have learned about creating an application.
- Create a Find/Browse form type.
- Attach a business view.
- Add form controls.

### Before You Begin

The Class Item Master application consists of two forms. The first form (Find/Browse) allows you to search for and delete records from your Item Master table. The second form (Fix/Inspect) allows you to add and modify records in the Item Master table. You will create the Find/Browse form in this exercise.

### ► Steps to complete

1. Use the following information to add an interactive application object to the Class Item Master project:

| Field | Value |
|---|---|
| Object Name | P5601xx |
| Description | Class Item Master |
| Product Code | 56xx |
| Product System Code | 56xx |
| Object Use | 1 |

2. Use business view V5601Axx for the form.

3. Set up the form so that it allows you to search for items using the Item Type and Item Description fields as filters.

4. Create Toolbar buttons that allow you find, add, delete, and select records.

5. Verify that your entry form follows all standards and is aligned properly.

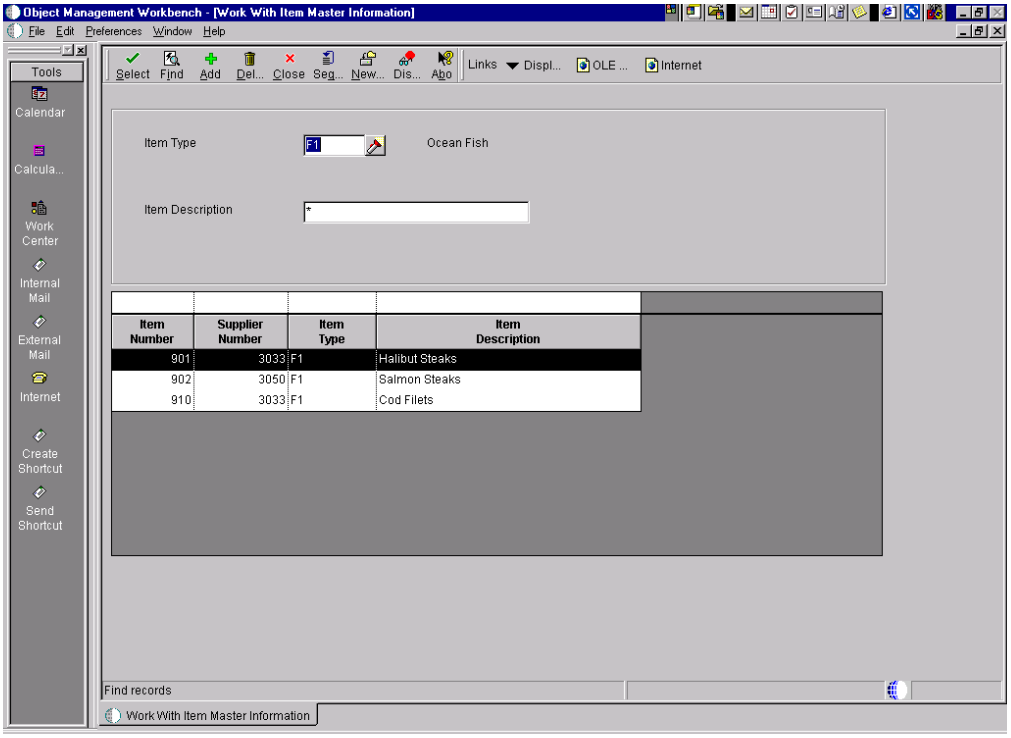
6. Verify that your grid contains the Item Number, Supplier Number, Item Type, and Item Description fields.

**Hint**
Supplier Number is the same as the Address Number field.

7. Sort the grid by Item Number.

8. Verify that your form is similar to the following form.

Work With Item Master Information (Find/Browse) Form:


Object Management Workbench - [Work With Item Master Information]
File Edit Preferences Window Help
Tools
Calendar
Calcula...
Work Center
Internal Mail
External Mail
Internet
Create Shortcut
Send Shortcut
Select Find Add Del... Close Seq... New... Dis... Abo
Links Displ... OLE ... Internet
Find records
Work With Item Master Information

|  |  |  |
|---|---|---|
| Item Type | F1 | Ocean Fish |
| Item Description | * |  |

| Item Number | Supplier Number | Item Type | Item Description |
|---|---|---|---|
| 901 | 3033 | F1 | Halibut Steaks |
| 902 | 3050 | F1 | Salmon Steaks |
| 910 | 3033 | F1 | Cod Filets |

# Directed Exercise: Designing a Fix/Inspect Form (P5501xx)

### Business Purpose

After you have created the table and business view objects and the Find/Browse form, which is the entry point to the application, you will use the Form Design Aid to create a second form to maintain the Class Address Book records.

You use the Form Design Aid to combine forms, controls, and event logic into applications. The tool provides you with a workspace for your form. In this directed exercise, you will create the Fix/Inspect form with tabs for your Class Address Book application (P5501xx) and attach business view V5501Axx.

### Objective

Upon completion of this exercise, you will be able to do the following:

- Create an application with Quick Form.
- Choose a Fix/Inspect form type.
- Attach a business view.
- Use a Tab Control.
- Add form controls.
- Associate descriptions.
- Move and size controls.
- Add data dictionary fields.
- Describe how to distinguish between a data dictionary field and a business view field on the form.
- Disconnect control text from control variables.
- Explain alignment options.

### ► Steps to complete

Follow along with your instructor to create the Class Address Book Revisions (Fix Inspect) form using Quick Form.

**Note**

This Fix/Inspect form is for demonstration purposes only. You will create an additional Fix/Inspect form that uses the Tab control to use during the remainder of the class.

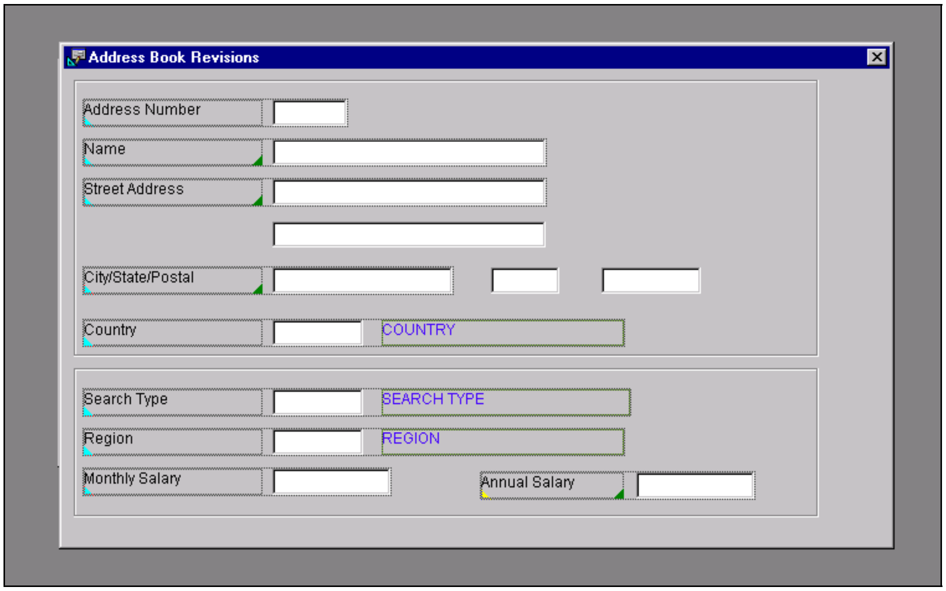
1. In Object Management Workbench, choose the P5501xx Class Address Book application in the Class Address Book project, and click Design.
2. On the Design Tools tab, click Start Form Design Aid.
3. In Form Design Aid, choose Fix/Inspect from the Create on the Form menu.
4. On Fix/Inspect Form Properties, type Address Book Revisions in the Title field.

5. Attach business view V5501Axx.

   a. From the Form menu, choose Quick Form.

   b. Choose the controls for the form and arrange them as shown in the following illustration.

Address Book Revisions (Fix/Inspect) Form without Tab Controls:



Address Book Revisions

Address Number

Name

Street Address

City/State/Postal

Country COUNTRY

Search Type SEARCH TYPE

Region REGION

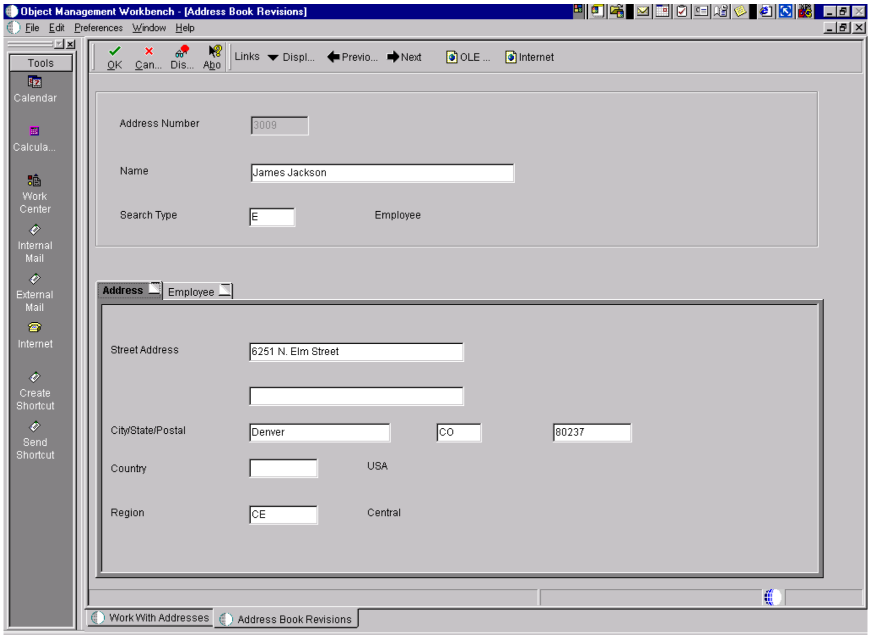
Monthly Salary

Annual Salary

## ▶ Steps to complete

Build the Fix/Inspect form in the following illustrations with the instructor.

Address Book Revisions (Fix/Inspect) Form – Address Tab:



Address Number 3009

Name James Jackson

Search Type E Employee

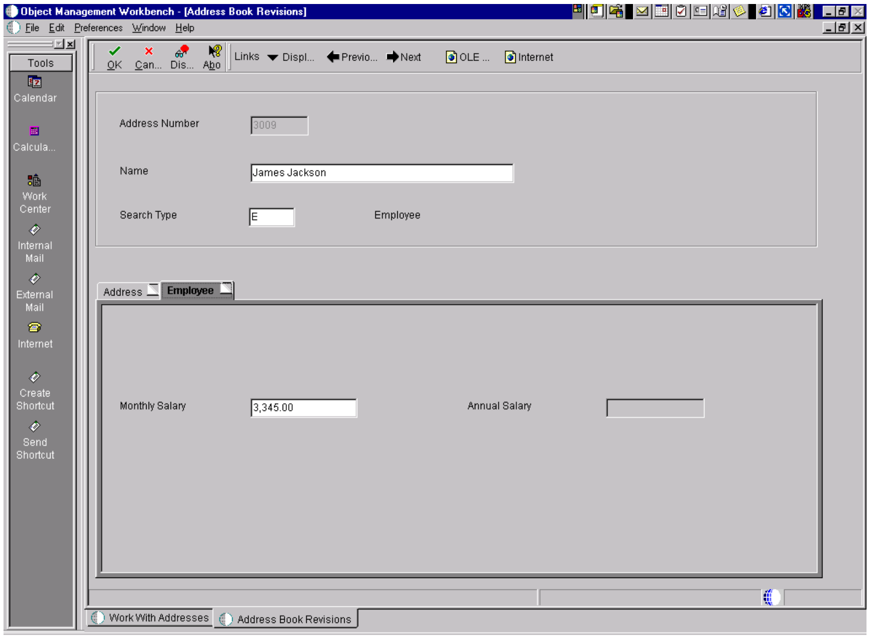
Street Address 6251 N. Elm Street

City/State/Postal Denver CO 80237

Country USA

Region CE Central

Address Book Revisions -- Employee Tab:



Address Number: 3009

Name: James Jackson

Search Type: E Employee

Monthly Salary: 3,345.00

Annual Salary:

1. In Form Design Aid, choose Fix/Inspect from the Create on the Form menu.

2. On Fix/Inspect Form Properties, type Address Book Revisions in the Title field.

3. Attach business view V5501Axx.

4. From the Insert menu, choose Tab Control.

5. Name the first tab control Address.

6. To add a second tab page, choose Tab Page from the Insert menu.

7. Name the second tab Employee.

### ► To complete the header section of the form

1. Insert Address Number, Alpha Name, and Search Type as business view fields.

2. Align the Address Number and Alpha Name controls using the Layout toolbar.

3. Change Alpha Name to Name.

    a. Access the *Default curser on add mode* property in the options tab for the Name control and change it to YES.

4. Add an associated description for Search Type.

5. Disable the Address Number control by accessing the control properties and turning on the Disabled option.

6. Create a group box.

### ► To create the Address Tab layout

1. Insert Address Line 2, Address Line 3, City, State, Postal Code, Country, and Region as business view fields.

2. Change Address 2 to Street Address.

3. Disconnect and clear the static text from the Address Line 3 control.

4. Disconnect the static text from the controls for State and Postal Code so that you can clear the text and move the controls.

5. Change the static text for City to City/State/Postal and move the State and Postal controls next to the City control.

6. Add Associated Descriptions for Country and Region.

### ► To create the Employee Tab layout

1. Insert Monthly Salary (SASalary) as a business view field.

2. Add a control for Annual Salary by inserting a data dictionary field.

    a. In the Data Dictionary Browser…, type math01 in the QBE line under Alias and then click and drag the item to the place that you want it to appear on the form.

    b. Change the static text and the control names to Annual Salary.

3. Disable the Annual Salary control by accessing the control properties and turning on the Disabled option.

**Note**
In a future exercise, you will create logic that calculates the Annual Salary and updates this field automatically.

### ► To create the Tab Sequence

1. Choose Tab Sequence from the Form menu to turn it on.

    After you define the sequence, choose Tab Sequence from the Form menu to turn it off.

2. Save and Test the form.

## Exercise: Designing a Fix/Inspect Form (P5601xx)

### Business Purpose

After you have created the table and business view objects, you will create a second form, the Fix/Inspect form, for the Class Item Master application (P5601xx). The Fix/Inspect form allows you to add and modify records in your Item Master table.

### Objective

Upon completion of this exercise, you will be able to do the following:

- Apply what you have learned about creating an application.
- Create a Fix/Inspect form type.
- Attach a business view.
- Design a form layout and add form controls.
- Add group boxes.
- Create text variables.
- Insert data dictionary fields.
- Discuss the use of next numbering.

### ► Steps to complete

1. What type of form is typically used to add or modify single records?

---

2. Create your single record maintenance form using the answer to number one as your form type.

   Your form should include the following data items:

| | Alias | Data Item |
|---|---|---|
| 1 | SAITM | SAItemNumber |
| 2 | SADSC1 | SADescription |
| 3 | SADSC2 | SACatalogDescription |
| 4 | SAAN8A | SAAddressNumberA |
| 5 | SATYPE | SAItemType |
| 6 | SAPREQ | SAQuantityOnOrder |
| 7 | SAUOM1 | SAUnitofMeasure |
| 8 | SAPRRC | SAAmountUnit |
| 9 | SAPQOH | SAQuantityOnHand |
| 10 | SAROP | SAReorderPoint |

3. Which business view includes these items?

---

4. Is next numbering appropriate for any of the controls on this form and, if so, which ones?

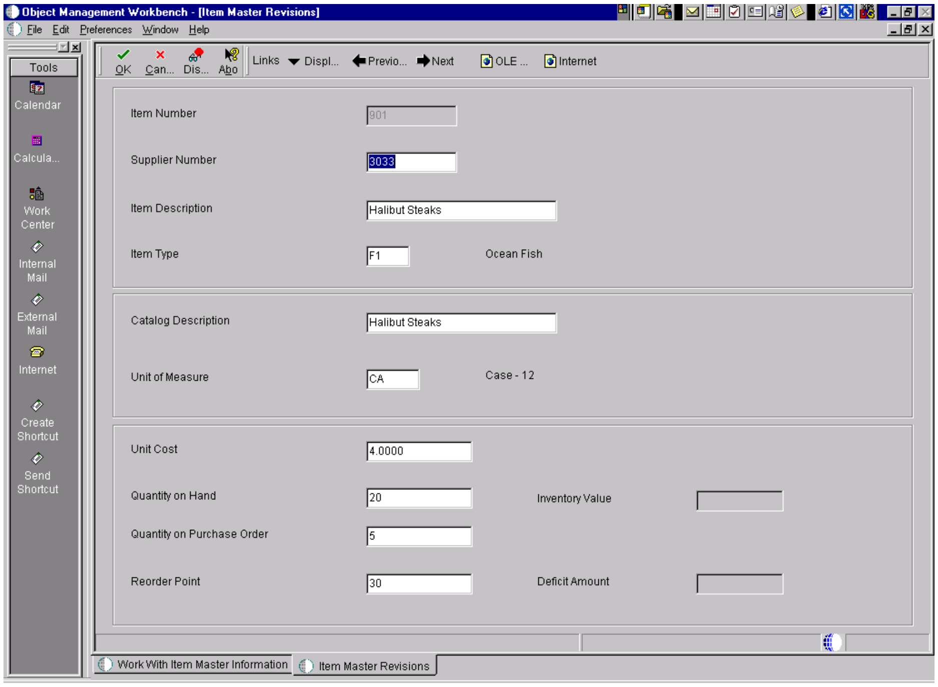
---

5. If a data dictionary item has a next number index assigned to it, how can you prevent the system from automatically assigning a next number to that field on a form?

---

6. Create the Item Master Revisions form to appear like the following example.

Item Master Revisions (Fix/Inspect) Form:



| | | | |
|---|---|---|---|
| Item Number | 901 | | |
| Supplier Number | 3033 | | |
| Item Description | Halibut Steaks | | |
| Item Type | F1 | Ocean Fish | |
| Catalog Description | Halibut Steaks | | |
| Unit of Measure | CA | Case - 12 | |
| Unit Cost | 4.0000 | | |
| Quantity on Hand | 20 | Inventory Value | |
| Quantity on Purchase Order | 5 | | |
| Reorder Point | 30 | Deficit Amount | |

## Note

| Alias | Description | Class | Data Type |
|---|---|---|---|
| MATH01 | Math Numeric | Currency | Number |
| EV01 | Alpha Numeric | Upper Case | Character |
| AA | Amount Field | Currency | Number |

## Exercise: Understanding Forms

1. Explain the difference between database items and dictionary items. Do dictionary items update tables?

2. What is the effect of deleting a data item from an existing business view?

3. How would you override a data dictionary default trigger on a data item in your form?

## Directed Exercise: Creating a Form Interconnection (P5501xx)

### Business Purpose

You use form interconnections to automatically pass data from one form, the source, to another form, the target. Now that you have designed two forms for the P5501xx Class Address Book application, you need to create form interconnections so that you can access the Fix/Inspect form from the Find/Browse form.

### Objectives

Upon completion of this exercise, you will be able to do the following:

- Use menu and toolbar exits to create form exits
- Understand the difference between modal and modeless processing
- Add event rules to select a target form
- Pass a value from the source form to the target form

### ► Steps to complete

1. In the Class Address Book project, choose the Class Address Book P5501xx application, and then choose Design.
2. Click the Design Tools tab, and then click Start Form Design Aid.
3. Choose the Work with Addresses (Find/Browse) form.
4. From the Form menu, choose Menu/Toolbar Exits.
    a. Choose the &Add menu exit and click the Event Rules button.
5. For the Button Clicked event, click the Form Interconnect button.
    a. On Work with Applications, type P5501xx in the Object Name field in the QBE line, and then click Find.
    b. Choose P5501xx.
    c. On Work With Forms, choose the Address Book Revisions form.

       Do not change the Data Structure window. When adding a record, no values exist to pass from the Work With Addresses form to the Address Book Revisions form.
    d. Click the Modeless option.
6. Click OK.

   The Event Rules Design window displays the following entry:

```
Call( App:P5501xx , Form: W5501xxC)
```

**Note**
The name of your form might not be W5501xxC.

7. Save and exit from the Event Rules Design window.

8. On Menu/Toolbar Exits, choose the &Select and click the Event Rules button.

9. For the Button Clicked event, click the Form Interconnect button.

   a. On Work with Applications, type P5501xx in the Object Name field of the QBE line, and then click Find.

   b. Choose P5501xx.

   c. On Work With Forms, choose the Address Book Revisions form.

   d. Change the Data Structure window to the following value.

      GC AddressNumber → mnSAAddressNumber

   e. Click the Modeless option.

10. Click OK.

    The Event Rules Design window displays the following entry:

```
Call( App:P5501xx , Form: W5501xxC )
```

**Note**

The name of your form might not be W5501xxC.

11. Save and exit from the Event Rules Design window.

12. Close the Menu/Toolbar Exits window.

13. Save and exit from the Class Address Book application.

14. On the Design Tools tab, choose Run the Application.

15. Add one record on the Address Book Revisions (Fix/Inspect) form.

16. Exit Object Management Workbench.

17. From the Universal Table Browser, verify that the F5501xx table shows the data that you entered in the previous step.

    Do not continue to add records to the Class Address Book until you verify that the data is being written to the table.

18. When you have verified that data appears in the F5501xx table, add a minimum of three customer, three employee, and three supplier records.

    a. Choose one record in the grid of Work With Addresses to test the form interconnection that you created for the Select button menu exit.

       Choosing a record in the grid should display the corresponding record in the Address Book Revisions form.

## Exercise: Creating a Form Interconnection (P5601xx)

### Business Purpose

You use form interconnections to automatically pass data from one form, the source, to another form, the target. Now that you have designed two forms for the P5601xx Class Item Master application, you need to create form interconnections so that you can access the Fix/Inspect form from the Find/Browse form.

### Objectives

Upon completion of this exercise, you will be able to do the following:

- Use menu and toolbar exits to create form exits
- Understand the difference between modal and modeless processing
- Add event rules to select the target form
- Pass a value from the source form to the target form

### ► Steps to complete

1. Access the P5601xx Find/Browse form, and then create a form interconnection on the Add button that takes the user to the P5601xx Fix/Inspect form.

2. Create a form interconnection on the Select button that takes the user to the P5601xx Fix/Inspect form.

3. On the Design Tools tab, click Run the Application, and then add one record on the Item Master (Fix/Inspect) form.

4. Exit Object Management Workbench.

5. From the Universal Table Browser, verify that the F5601xx table shows the data that you entered in the previous step.

   Do not continue to add records into the Class Item Master application until you verify that the system added the data to the table.

6. When you have verified that data appears in the F5601xx table, add at least six inventory items to the Item Master form, and assign the same item type to at least two records.

7. Choose one record in the grid of Work With Item Master to test the form interconnection that you created for the Select button menu exit.

   Choosing a record in the grid should display the corresponding record on the Item Master Revisions form.

# Directed Exercise: Designing a Headerless Detail Form (P5501xx)

### Business Purpose

After you have created the table and business view objects for the Class Address Book application (P5501xx), you will create a Headerless Detail form to display phone number information in the grid of the form.

You use the Form Design Aid to combine forms, controls, and event logic into applications. The tool provides you with a workspace for your form. In this directed exercise, you will create the Headerless Detail form for your Class Address Book application (P5501xx).

### Objectives

Upon completion of this exercise, you will be able to do the following:

- Create a Headerless Detail form
- Attach a business view
- Sort the grid by line number
- Add a filter (no wildcard) for address number
- Disable the Address Number control
- Create an auto find on the grid
- Add a Delete button
- Test the form

### ▶ Steps to complete

Follow along as your instructor demonstrates how to create the Headerless Detail form.

1. On Object Management Workbench, choose the P5501xx Class Address Book application in the Class Address Book project and click Design.

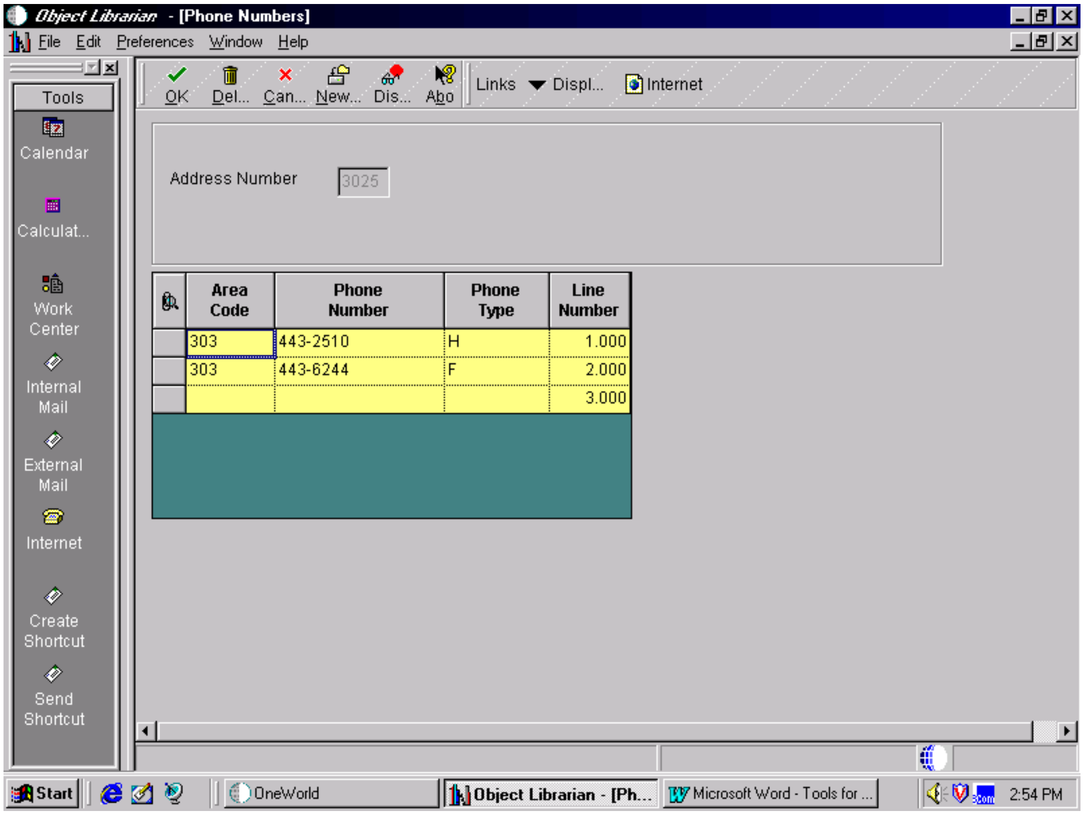
2. Click the Design Tools tab, and then click Start Form Design Aid.

3. On Form Design Aid, from the Form menu choose Create and then choose Headerless Detail.

4. On Headerless Detail Form Properties, type Phone Numbers in the Title field.

5. Attach business view V5515Axx.

Phone Numbers (Headerless Detail) Form:



Address Number 3025

| Area Code | Phone Number | Phone Type | Line Number |
|---|---|---|---|
| 303 | 443-2510 | H | 1.000 |
| 303 | 443-6244 | F | 2.000 |
|  |  |  | 3.000 |

6. Add Address Number as a form control from the Business View Column Browser.

   a. Verify that Address Number has a Filter Criteria of =.

   b. Access the properties for the Address Number control and click the Disabled option.

   c. Under Overrides, disable Next Numbering.

7. Create a group box around Address Number.

8. Add the grid columns according to the illustration by double-clicking the data item in the Business View Column Browser.

   a. Sort the grid by line number.

   b. In Grid Properties, choose Options, and then click the Automatically Find On Entry option.

**Note**
Because you included the Automatically Find on Entry option on the grid, you do not need a Find button on the menu toolbar.

9. From the Form menu, choose Toolbar exits and add a Delete button.

10. For the form properties, choose Options and click End Form on Add.

11. Turn on the Tab Sequence by choosing Tab Sequence from the Form menu.

    a. After you define the sequence, choose Tab Sequence from the Form menu to turn it off.

12. Test the form.

13. Save and exit from the form.

14. On Interactive Application Design, click OK to return to Object Management Workbench.

## Directed Exercise: Designing a Search and Select Form (P5501S)

### Business Purpose

After you have created the table and business view objects for the Class Address Book application (P5501xx), you will create a Search and Select form to allow you to search for records in the Class Address Book table (F5501xx).

The purpose of a Search and Select form is to locate a value and return it to the calling field. Because it is attached to a specific field and is called using a visual assist, after you design the Search and Select form, you must attach it to the specific data item for which it was created. You do this using the Visual Assist trigger in the data dictionary.

You use a Search and Select form to view records from only one table. To improve performance, you use a separate application and a unique business view.

In this exercise, you create a separate interactive application (P5501Sxx) for the Class Address Book Search & Select form. You attach the business view that was designed specifically for this form (V5501Sxx). After you complete the form, you attach it as a visual assist override to data dictionary item SAAN8Axx (Supplier Number) on the Item Master Revisions form in your Class Item Master Application.

### Objectives

Upon completion of this exercise, you will be able to do the following:

- Create a Search and Select form
- Understand why Search and Select forms usually have their own business views
- Attach the Search and Select form to a data dictionary item
- Test the form

### ► Steps to complete

Follow along as your instructor demonstrates how to create the Search and Select form.

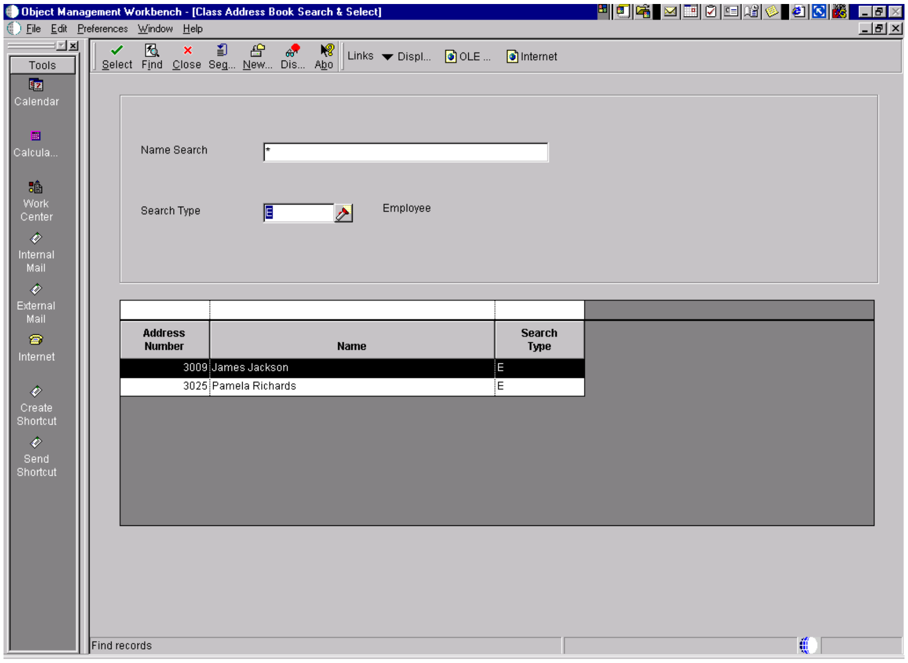
1. Add an interactive application object to the Class Address Book project.

   The name and descriptive information follows:

| Field | Value | Comment |
|---|---|---|
| Object Name | P5501Sxx | |
| Description | Class Address Book Search & Select | |
| Product Code | 55xx | Reserved for Clients |
| Product System Code | 55xx | Reserved for Clients |
| Object Use | 1 | Programs (Default) |

2. Click OK.
3. Click the Design Tools tab, and then click Start Form Design Aid.
4. On Form Design Aid from the Form menu choose Create and then Search & Select.

5. On Search & Select Form Properties, type Class Address Book Search & Select in the Title field.



Object Management Workbench - [Class Address Book Search & Select]
File Edit Preferences Window Help
Select Find Close Seq... New... Dis... Abo
Links Displ... OLE ... Internet
Tools
Calendar
Calcula...
Work Center
Internal Mail
External Mail
Internet
Create Shortcut
Send Shortcut
Find records

Name Search *

Search Type E Employee

| Address Number | Name | Search Type |
|---|---|---|
| 3009 | James Jackson | E |
| 3025 | Pamela Richards | E |

6. On Class Address Book Search & Select, from the Form menu, choose Business View, Add Business View ….

   a. On the Search & Select Form properties window, type V5501Sxx for the object name, and then click Find.

   b. Choose V5501S when it appears in the grid.

7. Add Alpha Name and Search Type as form controls from the Business View Columns Browser.

   a. Verify that Alpha Name has a Filter option of >= and that Wildcard Display is set to Yes.

   b. Verify that Search Type has a Filter option of = and that Wildcard Display is set to Yes.

   c. Align the Search Type and Alpha Name.

   d. Add an associated description to Search Type.

   e. Create a group box around Search Type and Alpha Name.

8. To add the grid columns, select the grid and then double-click the data item in the Business View Columns Browser.

   a. Verify that the grid is sorted by address number.

   b. Rename the grid column Alpha Name to Name.

   c. Rename the grid column Sch Typ to Search Type.

9. Set the Tab Sequence by choosing Tab Sequence from the Form menu.

    a. After you define the sequence, choose Tab Sequence from the Form menu to turn it off.

### ► To test the application

1. On Form Properties, click Entry Point.

2. On the Form menu, click Run.

3. When the Search & Select form appears, click Find.

    The records you previously entered in the Class Address Book should appear.

4. Test the Name Search and Search Type controls to verify that they are working correctly.

    For example, if you choose E for the search type, only the employees should appear. A search using the Name Search control should display all names that are equal to and greater than the criteria that you entered.

5. Save and exit from the Class Address Book Search & Select application.

6. Save and exit from Form Design Aid.

7. On Interactive Application Design, click OK to return to Object Management Workbench.

8. On Object Management Workbench, check in the P5501Sxx application by clicking the Check In action button.

    **Caution**
    The P5501Sxx interactive application must be checked in so that the Search & Select form is available for the data dictionary procedures that follow.

### ► To attach the Search & Select form to data dictionary Item SAAN8Axx and test

1. On Object Management Workbench, choose the Class Item Master P5601xx application in your Class Item Master project and click Design.

2. Click the Start Form Design Aid button in the Interactive Application Design window.

3. On the Item Master Revisions form, double-click the Supplier Number control.

    a. Choose the Overrides tab in the Grid Column Properties dialog box.

    b. Override the Visual Assist option to be the Search Form W5501xxA.

    c. Save and exit the Class Item Master application.

4. In the Interactive Application Design window, click the Run the Application button.

5. On Work With Item Master Information, click Add.

A Visual Assist displays for the Supplier Number, which displays your Search & Select form when clicked.

The supplier that you chose appears in the QBE line on the Work With Item Master Information form.

**Note**
If the visual assist was added to the data item in the data dictionary, the visual assist would appear globally; that is, wherever Address Number is used. This would not be appropriate in some applications, so the properties would have to be overridden to remove the visual assist wherever it is inappropriate.

## Exercise: Questions and Answers for Forms Design

### Business Purpose

This exercise is a series of questions that will reinforce what you have learned so far about forms design.

1. To add a new record to a table, which type of form would you use?

2. You are adding a new record to a table, but the form you are using does not have a grid. Which type of form is it?

3. Which type of form do you use to locate a record in a table?

4. To delete more than one detail record from the same form, which form would you use?

5. Which form would you use to locate a value and return it to the search field?

6. Use the following type of form to display a request or action from the user:

7. Which type of form has a tree format?

8. What is the primary difference between a Headerless Detail and a Header Detail form?

9. Which type of form is the initial entry point to an application?

10. Which is the only form type that has more than one business view?

# Exercise: Which Form Should I Use?

### Business Purpose

One of the most challenging aspects of forms design is determining which type of form to create for an application. You need to understand the available form types, because each form has characteristics that facilitate different tasks. In this exercise, you are asked to identify various types of forms and list some of their characteristics.

### Objective

For each example shown, do the following:

- Identify the type of form
- List at least one characteristic of each form type

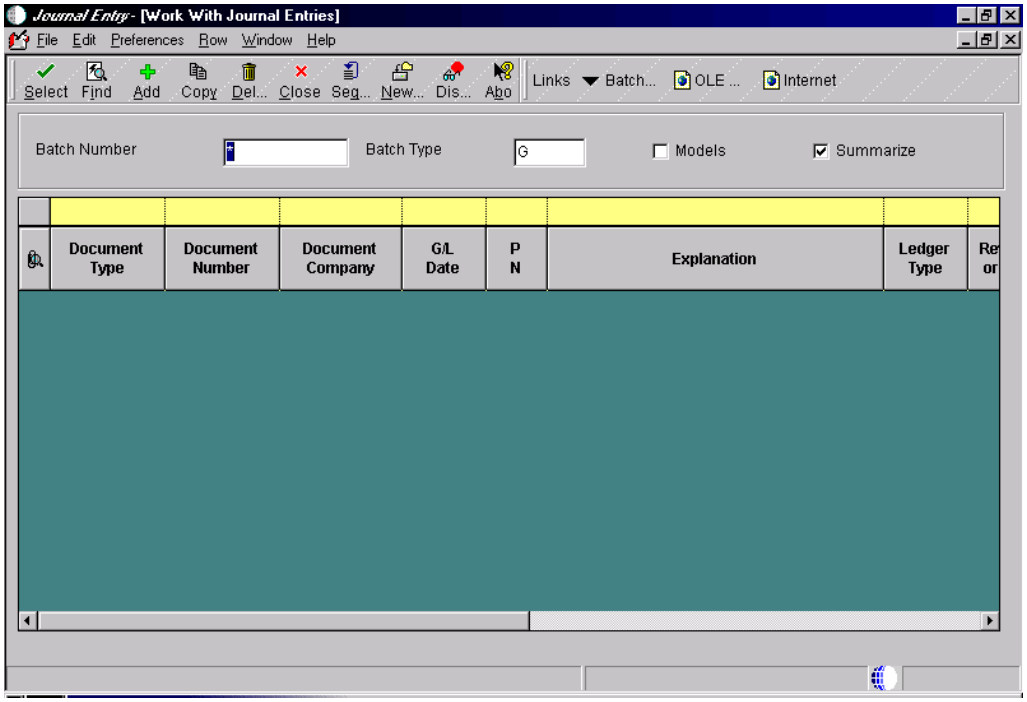
### ▶ Steps to complete

Write an answer to each of the following questions.

1. What type of form is Form 1?

   a. Identify one feature of this form:

Form 1:



Batch Number *

Batch Type G

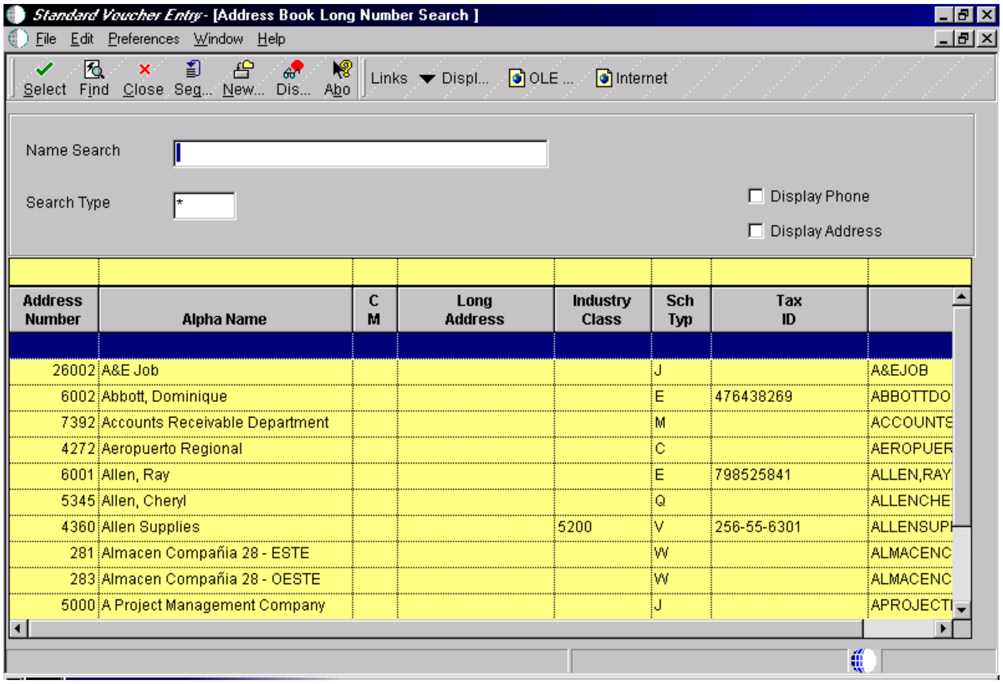
☐ Models

☑ Summarize

2. What type of form is Form 2?

a. Identify one feature of this form:

Form 2:



*Standard Voucher Entry* - [Address Book Long Number Search ]

Name Search

Search Type *

□ Display Phone

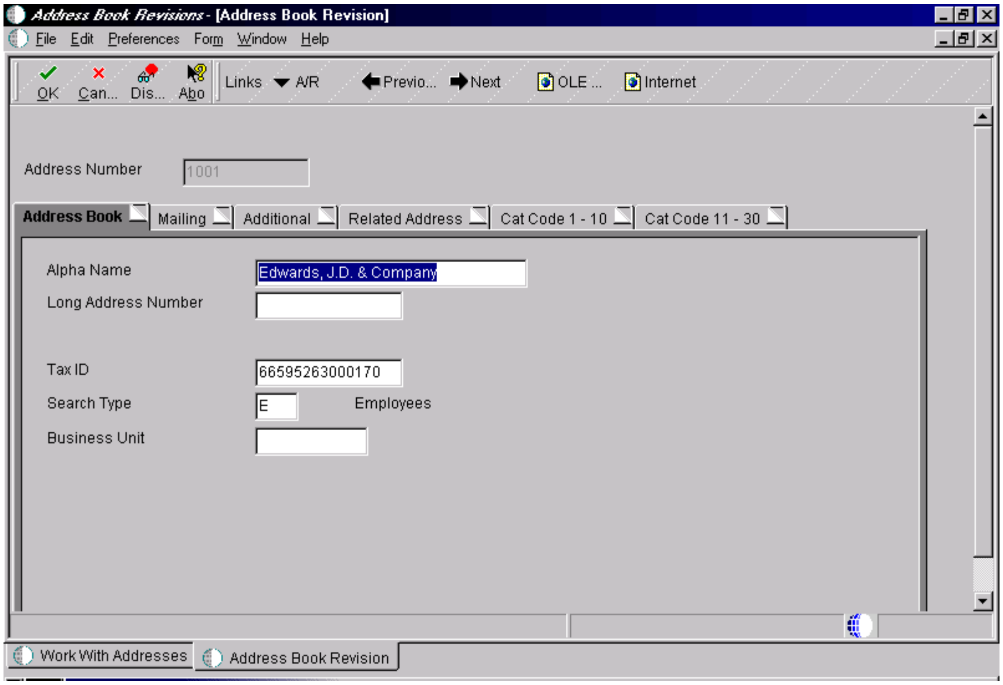
□ Display Address

| Address Number | Alpha Name | C M | Long Address | Industry Class | Sch Typ | Tax ID | |
|---|---|---|---|---|---|---|---|
| 26002 | A&E Job | | | | J | | A&EJOB |
| 6002 | Abbott, Dominique | | | | E | 476438269 | ABBOTTDO |
| 7392 | Accounts Receivable Department | | | | M | | ACCOUNTS |
| 4272 | Aeropuerto Regional | | | | C | | AEROPUER |
| 6001 | Allen, Ray | | | | E | 798525841 | ALLEN,RAY |
| 5345 | Allen, Cheryl | | | | Q | | ALLENCHE |
| 4360 | Allen Supplies | | | 5200 | V | 256-55-6301 | ALLENSUP |
| 281 | Almacen Compañia 28 - ESTE | | | | W | | ALMACENC |
| 283 | Almacen Compañia 28 - OESTE | | | | W | | ALMACENC |
| 5000 | A Project Management Company | | | | J | | APROJECTI |

3. What type of form is Form 3?

a. Identify one feature of this form:

Form 3:



Address Book Revisions - [Address Book Revision]

Address Number: 1001

Alpha Name: Edwards, J.D. & Company

Long Address Number:

Tax ID: 66595263000170

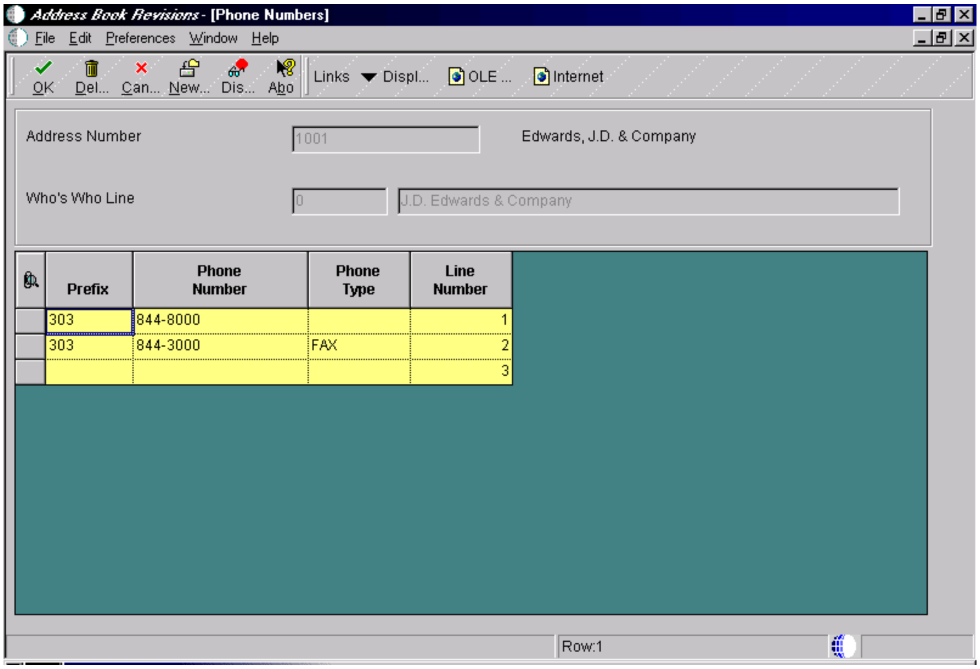
Search Type: E Employees

Business Unit:

4. What type of form is Form 4?

a. Identify one feature of this form:

Form 4:



Address Number 1001 Edwards, J.D. & Company

Who's Who Line 0 J.D. Edwards & Company

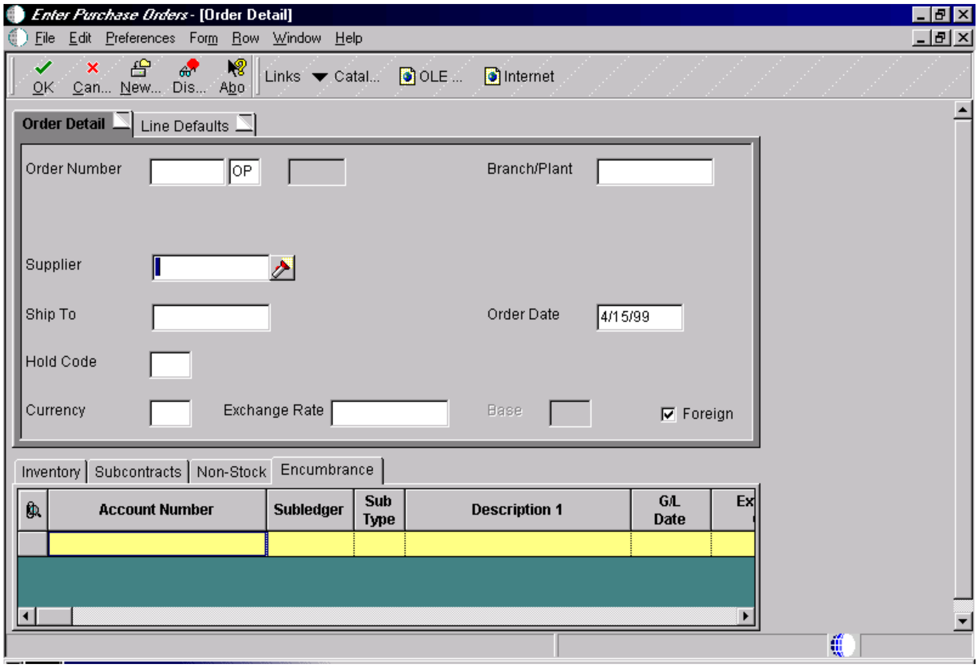
| Prefix | Phone Number | Phone Type | Line Number |
|---|---|---|---|
| 303 | 844-8000 | | 1 |
| 303 | 844-3000 | FAX | 2 |
| | | | 3 |

Row:1

5. What type of form is Form 5?

a. Identify one feature of this form:

Form 5:



Order Number OP

Branch/Plant

Supplier

Ship To

Order Date 4/15/99

Hold Code

Currency

Exchange Rate

Base

☑ Foreign

| | Account Number | Subledger | Sub Type | Description 1 | G/L Date | Ex |
|---|---|---|---|---|---|---|
| | | | | | | |

# Event Rules

# Directed Exercise: Adding Event Rules to Class Address Book (P5501xx)

## Business Purpose

The Class Address Book application contains records for employees, customers, and suppliers. When a user is working with an employee record, the Monthly Salary and Annual Salary controls should appear on the form. When a user is working with a record that is not an employee, these fields should not appear. Additionally, when the Monthly Salary field contains a value, the system calculates the annual salary automatically. Finally, a Form exit must be added to the Address Book Revisions form so that it can access the Class Phone Numbers form.

## Objective

Upon completion of this exercise, you will be able to do the following:

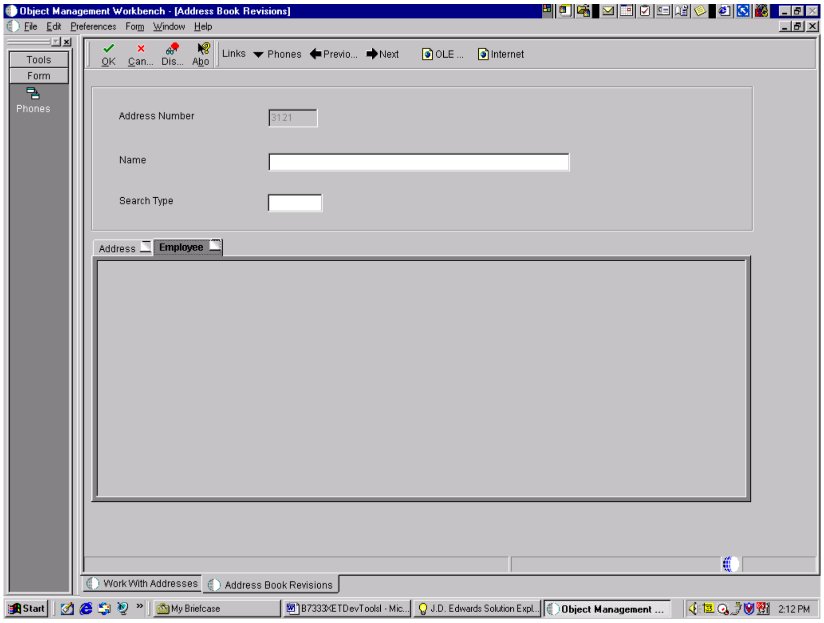
- Create conditional if/then/else statements
- Create assignments
- Create event rule logic to hide and show controls based on condition
- Use event rules to perform a calculation
- Add a Form exit to the Class Address Book Phone Numbers form

## ▶ Steps to complete

1. Display the Monthly Salary and Annual Salary controls only if the Address Book record is for an employee.

   Consider what might happen if a record were originally designated as an employee and then changed to another search type. For details, see the following form examples:

   When you add a record, the Employee tab on Address Book Revisions is empty:

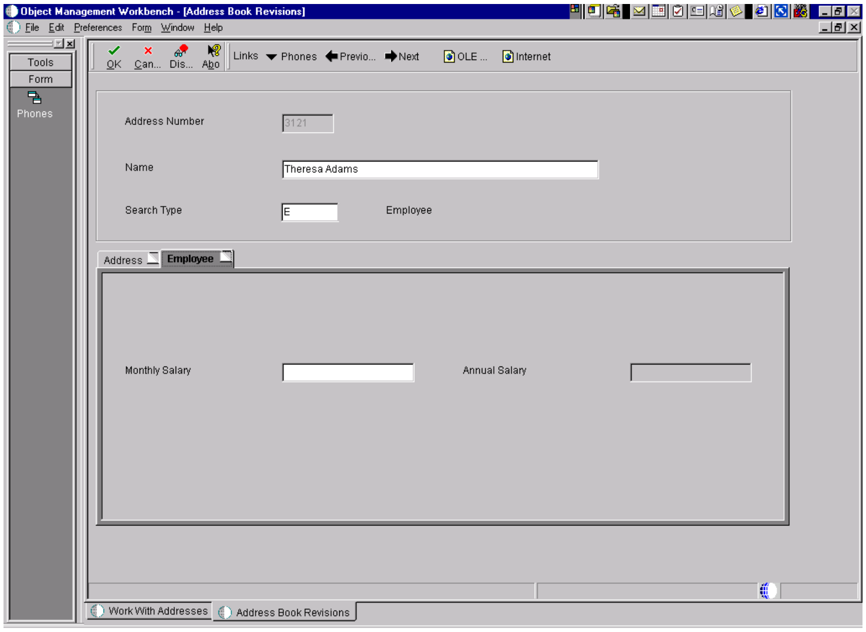


Address Number 3121

Name

Search Type

If you enter Search Type E, Monthly Salary and Annual Salary controls display on the Employee tab.


Object Management Workbench - [Address Book Revisions]
File Edit Preferences Form Window Help
OK Can... Dis... Abo
Links Phones Previo... Next OLE ... Internet
Tools
Form
Phones
Address
Employee
Work With Addresses
Address Book Revisions

Address Number 3121

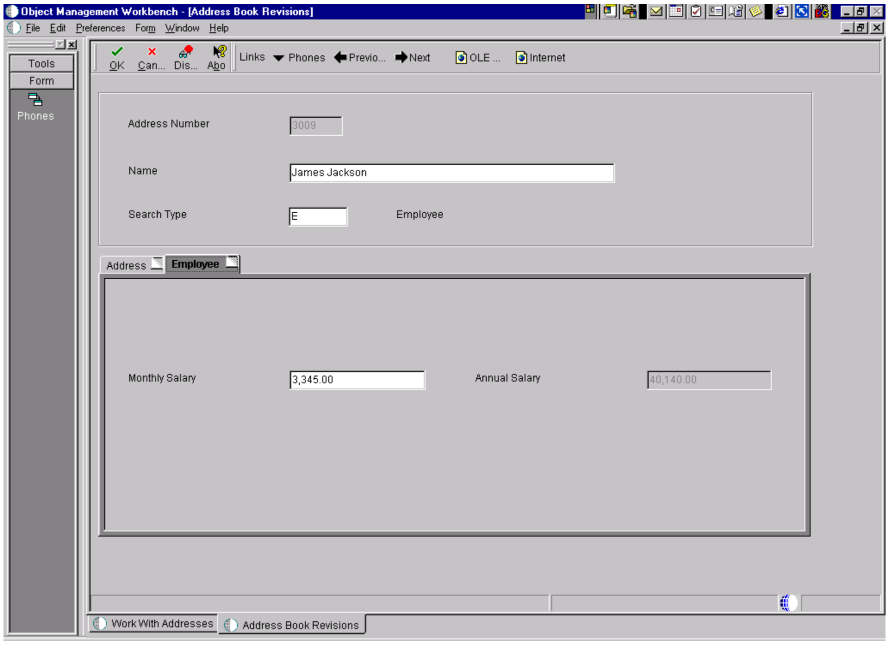
Name Theresa Adams

Search Type E Employee

Monthly Salary

Annual Salary

If you enter a Monthly Salary, the system automatically calculates the Annual Salary.


Object Management Workbench - [Address Book Revisions]
File Edit Preferences Form Window Help
OK Can... Dis... Abo
Links Phones Previo... Next OLE ... Internet
Tools
Form
Phones
Address
Employee
Work With Addresses
Address Book Revisions

Address Number 3009

Name James Jackson

Search Type E Employee

Monthly Salary 3,345.00

Annual Salary 40,140.00

## Hints

This logic needs to occur on multiple events. Browse the System Functions to Hide/Display a control.

On Post Dialog is Initialized, add the following event rules:

```
If BC Sample Application - Search Type (F5501xx) is equal to “E”
Show Control(FC Monthly Salary)
Show Control(FC Annual Salary)
Else
Hide Control(FC Monthly Salary)
Hide Control(FC Annual Salary)
End If
```

On Control is Exited/Changed-Inline (for Search Type), add the following event rules:

```
If BC Sample Application - Search Type (F5501) is equal to "E"
Show Control(FC Monthly Salary)
Show Control(FC Annual Salary)
Else
Hide Control(FC Monthly Salary)
Hide Control(FC Annual Salary)
End If
```

2. Calculate Annual Salary

3. Again, consider what would happen to the data in the Monthly Salary and Annual Salary controls if a record were originally entered as an Employee and then changed to another search type.

   **Hint**
   This calculation will need to occur on multiple events.

On Post Dialog is Initialized on the form and Control is Exited/Changed-Inline for the Monthly Salary control, add the following event rule:

```
FC Annual Salary = [FC Monthly Salary]*12
```

4. On the Address Book Revisions form, add a Form exit that takes the user to the Class Address Book (P5501xx) headerless detail Phone Numbers form.

   You will pass the Address Number from the Address Book Revisions form to Address Number on the Phone Numbers form.

   a. Choose the Address Book Revisions (Fix/Inspect) form.

   b. From the Form menu, choose Menu/Toolbar Exits….

   c. On Menu/Toolbar Exits…, click the Insert button.

d. On Menu Exit Properties, click the Class drop-down list and choose Form, and then click OK.

e. On Menu/Toolbar Exits, click the Insert button.

f. In the Menu Exit Properties dialog box, choose User Defined from the Class drop-down menu and enter the following information:

   Short Text = &Phones

   Long Text = Phones.

g. Click OK.

h. On Menu/Toolbar Exits, choose the &Phones record and click the Event Rules button.

i. For the Button Clicked event, click the Form Interconnect button.

j. On Work With Applications, type P5501xx in the Object Name field of the QBE line, and then click Find.

k. Choose P5501xx.

l. On Work With Forms, choose the Phone Numbers form and click the Select button.

m. On Form Interconnections, change the Data Structure to pass the FC Address Number to mnSAAddressNumber.

   Do not change the second entry (mnSALineNumber). It should remain <NOT Assigned>.

n. Click OK.

   The Event Rules Design window contains the following entry:

```
Call( App:P5501xx , Form: W5501xxB )
```

**Note**
Your form might be named something other than W5501xxB.

o. Save and exit from the Event Rules Design window.

p. Click Close.

q. Save and exit the Class Address Book application.

r. On the Design Tools tab, click Run the Application.

s. On the Work With Addresses form, find and choose a record.

t. On the Address Book Revisions form, choose Phones from the Form exit bar.

u. Enter several phone numbers.

   You must enter the line number for each phone number that you add to the form.

# Exercise: Adding Event Rules to Class Item Master (P5601xx)

### Business Purpose

The Class Item Master application contains inventory records for items that have been purchased. If a valid supplier number appears for an item on the Item Master Revisions form, the user might want to call that supplier. With a Form exit to the Class Phone Numbers form, the user can access the supplier's phone numbers directly from the Item Master Revisions form. Additional event rule logic is required to calculate inventory value and deficit amount, as well as to display controls and handle blank fields.

### Objectives

Upon completion of this exercise, you will be able to do the following:

- Add a Form exit to the Class Address Book Phone Numbers form
- Create conditional if/then/else statements
- Create assignments
- Create event rule logic to hide and show controls based on condition
- Use event rules to perform a calculation
- Insert a default value into a field using conditional logic and an assignment

### ► Steps to complete

1. On the Item Master Revisions form, add a Form exit that takes the user to the Class Address Book (P5501xx) Phone Numbers form.

   a. Pass the Supplier Number from the Item Master Revisions form to the Address Number on the Phone Numbers form.

   b. Condition the form interconnection to occur only if a value is entered in the Supplier Number field.

2. Calculate the Inventory Value by multiplying the Quantity on Hand by the Unit Cost.

   **Hint**
   This calculation must occur on multiple events.

3. Calculate the Deficit Amount by adding the Quantity on Hand to the Quantity on Purchase Order and then subtracting the Reorder Point.

   **Hint**
   This calculation must occur on multiple events.

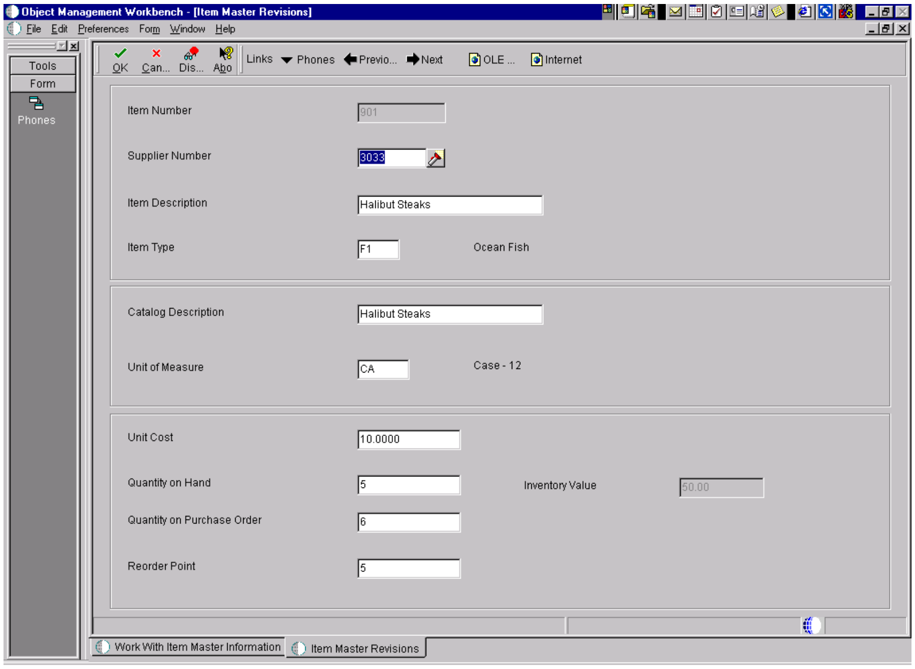
4. Display the Deficit Amount control only if its calculated value is less than zero.

**Hints**
This logic must occur on multiple events. Browse the System Functions to hide or display a control.

5. If the user leaves the Catalog Description field blank, enter the item description as the default value for the Catalog Description field.

## Examples of the Form

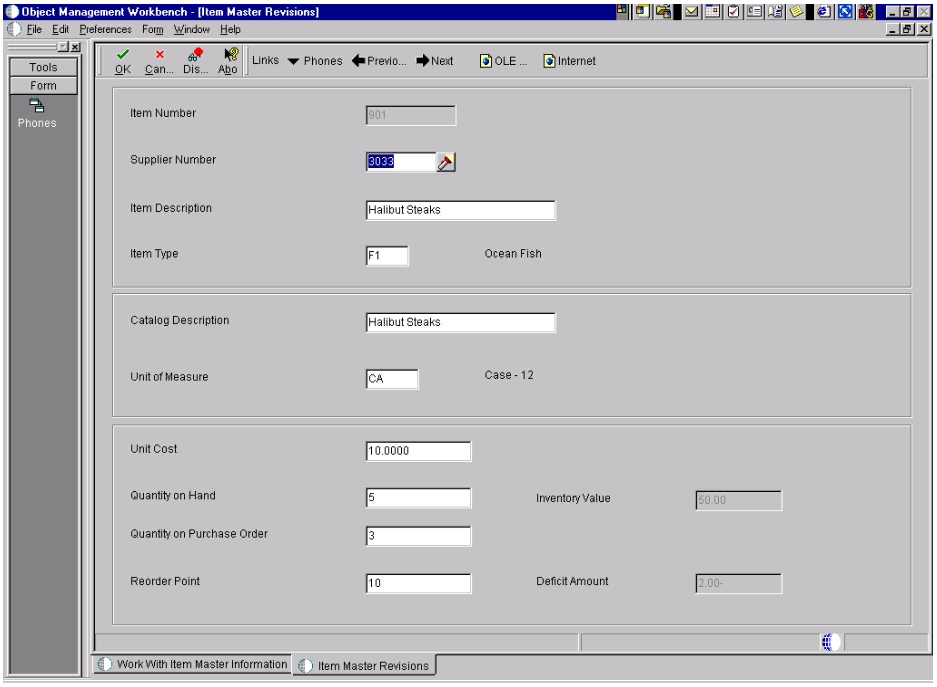
The Deficit Amount is not a negative value:



Object Management Workbench - [Item Master Revisions]
File Edit Preferences Form Window Help
Tools
Form
Phones
OK
Can...
Dis...
Abo
Links
Phones
Previo...
Next
OLE ...
Internet
Work With Item Master Information
Item Master Revisions

| | | | |
|---|---|---|---|
| Item Number | 901 | | |
| Supplier Number | 3033 | | |
| Item Description | Halibut Steaks | | |
| Item Type | F1 | Ocean Fish | |
| Catalog Description | Halibut Steaks | | |
| Unit of Measure | CA | Case - 12 | |
| Unit Cost | 10.0000 | | |
| Quantity on Hand | 5 | Inventory Value | 50.00 |
| Quantity on Purchase Order | 6 | | |
| Reorder Point | 5 | | |

The Deficit Amount is a negative value:



Object Management Workbench - [Item Master Revisions]
File Edit Preferences Form Window Help
Tools
Form
Phones
OK Can... Dis... Abo
Links Phones Previo... Next OLE ... Internet
Work With Item Master Information
Item Master Revisions

| | | | |
|---|---|---|---|
| Item Number | 901 | | |
| Supplier Number | 3033 | | |
| Item Description | Halibut Steaks | | |
| Item Type | F1 | Ocean Fish | |
| Catalog Description | Halibut Steaks | | |
| Unit of Measure | CA | Case - 12 | |
| Unit Cost | 10.0000 | | |
| Quantity on Hand | 5 | Inventory Value | 50.00 |
| Quantity on Purchase Order | 3 | | |
| Reorder Point | 10 | Deficit Amount | 2.00- |

## Optional Exercises

If time permits, complete the following additional steps.

► **Steps to complete**

1. On the Work With Item Master Information (Find/Browse) form (P5601xx), add a Row exit to the Phone Numbers form.

   **Hint**
   On a Find/Browse form, you will be focused on a specific record in the grid when you choose to exit to the Phone Numbers form. Because each record is a row in the grid, this exit becomes a Row exit instead of a Form exit.

2. On the Work With Addresses (Find/Browse) form (P5501xx), add a Row exit to the Phone Numbers form.

## Automatic Line Numbering

## Directed Exercise: Using Event Rule Variables for Automatic Line Numbering (P5501xx)

### Business Purpose

In the Address Book application, the Phone Numbers form requires line numbers. Until this point, when you added phone numbers you entered the associated line numbers manually. Typically, an input-capable grid increments the number of each grid line automatically until the last record appears. After the system displays all records from the table in the grid, the last line is blank except for the line number, which already has the next line number assigned to it. This feature is called automatic line numbering. You can add this feature to your form using event rules. You must complete the following four actions to enable automatic line numbering:

- Create a variable to hold the value of the line number
- Initialize the variable
- Number the lines as each line is pulled from the database
- Increment the line number and assign the new value to the next available line

### ► Steps to complete

1. On Object Management Workbench, choose the Class Address Book P5501xx application in the Class Address Book project, and then click Design.
2. Click the Design Tools tab, and then click Start Form Design Aid.
3. Choose the Phone Numbers (Headerless Detail) form.
4. Use data dictionary item LNID to create a variable to hold the value of the line number:
    a. Access Form Event Rules.
    b. On any event, click the Variable button.
    c. On Event Rules Variables, enter LNID in the DD Item field, and then click Add.
    d. In Variable Options Selection, under Scope, click Form.
    e. Change the User Name field to mnLineNumberCounter_LNID.
5. Click Finish and then OK.
6. Initialize the variable when entering the Phone Numbers form.

    The suggested event to use for initialization is Post Dialog is Initialized. Create the following assignment:

```
VA frm_mnLineNumberCounter_LNID = 0
```

7. As each line is retrieved from the database, determine whether the line number of that record is greater than the value of the variable.

If the line number is greater than the value of the variable, assign the value of that line number to the variable to ensure that the variable will always be equal to the highest line number in the table before it is incremented by 1 for the final line in the grid.

- Use the form event Grid Record is Fetched.

```
If BC Line Number (F5515xx) is greater than VA frm_mnLineCounterNumber_LNID
```

```
VA frm_mnLineCounterNumber_LNID = BC Line Number (F5515xx)

End If
```

8. Using the grid event Add Last Entry Row to Grid, increment the variable by 1 and assign the new value to the next available line.

```
VA frm_mnLineCounterNumber_LNID = [VA frm_mnLineCounterNumber_LNID]+1

GC LineNumber = VA frm_mnLineCounterNumber_LNID
```

9. Disable the Line Number control in the grid by choosing Grid Column Properties and then clicking Disable Input.

10. Save and exit the application.

11. On the Design Tools tab, choose Run the Application.

12. Choose a record and access the Phone Numbers form to verify that line numbers are automatically assigned when a new phone number entry is added to the grid.

# Media Objects

# Directed Exercise: Creating Media Objects for the Class Address Book Application (P5501xx)

### Business Purpose

Media objects are used to link information to application transactions. For example, you can use media objects to attach documents or images to a form or to a record in a grid. You can use event rules and system functions to attach media objects, or you can use standard processing within Forms Design Aid to implement media objects without defining any event rules. The standard processing method is demonstrated in this exercise.

### Objectives

Upon completion of this exercise, you will be able to do the following:

- Create a media object data structure
- Attach a media object to an application
- Use a text, image, OLE, or shortcut attachment

### ► Steps to complete

To enable an application for media objects, you need a GT type data structure to pass arguments from the application table to the media object table. You can create a GT object type in Object Management Workbench or choose an existing one. To see the entire process from beginning to end, this exercise begins with creating the data structure.

1. Create the media object data structure GT5501xx.

    a. On Object Management Workbench, choose the Class Address Book Project and click Add.

    b. Click the Media Object Data Structure option in the Object Librarian Objects group box.

    c. Click OK.

    d. Enter the following values in the Add Objects window:

| Field | Value | Comment |
|---|---|---|
| Object Name | GT5501xx | |
| Description | Class Address Book Objects | |
| Product Code | 55xx | Reserved for Clients |
| Product System Code | 55xx | Reserved for Clients |
| Object Use | 360 | Data Structures |

    e. Click OK.

    f. On the Design Tools tab, click Data Structure Design.

g. Find the alias SAAN8Axx in the data dictionary window, drag and drop it to the Structure Members window, and click OK.

   This field is the field on the grid of the Find/Browse form that will be used as the row key to which the media object is attached.

h. On Data Structure Design, click OK again to return to Object Management Workbench.

   The GT5501xx data structure now appears under the Objects node of the Class Address Book project.

2. Enable Media Object capabilities for the Work With Addresses (P5501xx) form.

   a. On Object Management Workbench, choose the Class Address Book application (P5501xx) in the Class Address Book project, and click Design.

   b. Click the Design Tools tab, and then click Start Form Design Aid.

   c. Choose the Work With Addresses (Find/Browse) form.

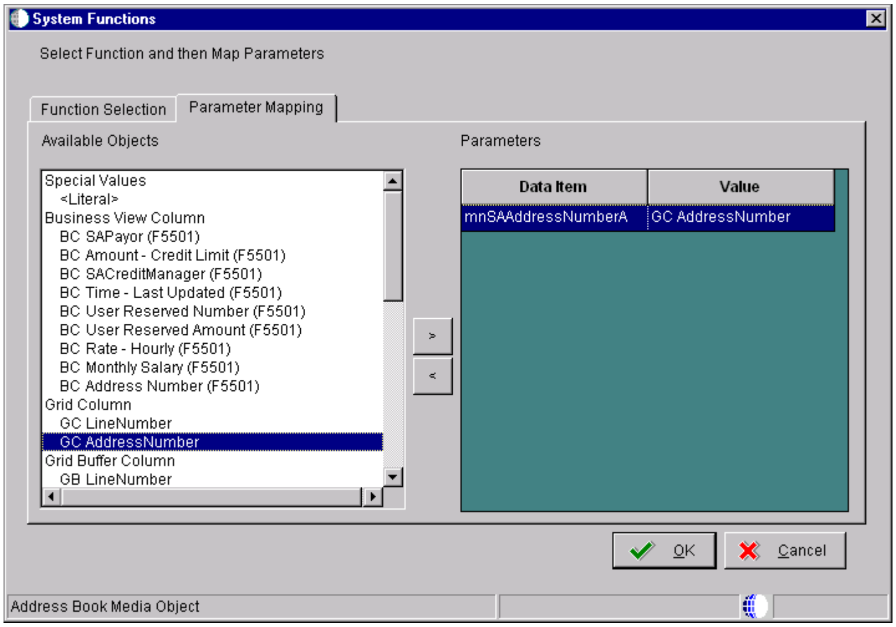
   d. From the Form menu, choose Media Objects Setup.

   e. Click Enable Automatic Media Object Functionality.

      - Verify that the Media Objects & Document Handling option is turned on.
      - In the Form Level Defaults group box, verify that the Edit Mode option is turned on.
      - In the Row Level Defaults group box, verify that the Edit Mode option is turned on.
      - Because we are going to design an attachment for a particular record in the grid of our Find/Browse form, click the Define Row Key button.

   f. On System Functions, double-click Media Object Structures.

   g. Scroll to GT5501xx, and double-click it.



Select Function and then Map Parameters

Available Objects

Special Values
  <Literal>
Business View Column
  BC SAPayor (F5501)
  BC Amount - Credit Limit (F5501)
  BC SACreditManager (F5501)
  BC Time - Last Updated (F5501)
  BC User Reserved Number (F5501)
  BC User Reserved Amount (F5501)
  BC Rate - Hourly (F5501)
  BC Monthly Salary (F5501)
  BC Address Number (F5501)
Grid Column
  GC LineNumber
  GC AddressNumber
Grid Buffer Column
  GB LineNumber

Parameters

| Data Item | Value |
|---|---|
| mnSAAddressNumberA | GC AddressNumber |

Address Book Media Object

h. Complete the Parameter Mapping as follows:

Data Item = mnSAAddressBookNumberA

Value = GC AddressNumber.

i. Click OK

j. On Work With Addresses, access Grid Properties to verify that Hide Row Headers is not turned on.

If Hide Row Headers is turned on, the media object paperclip icon does not appear on the form.

k. Save and exit the application.

3. On the Design Tools tab, click Run the Application.

Since you enabled Media Object functionality, an Attachments selection now appears on the Row toolbar and menu bar.

4. Add a Text Object.

a. Find all of the records in the grid, choose a record, and then choose Attachments.

b. On Media Objects, add a text object by choosing New and then Text from the File menu.

c. Type the text, and then choose Save & Exit from the File menu.

The record with the media object has a paper clip icon.

**Note**

To make the paper clip icon visible in the status bar, hover the cursor over the row header. Paper clips appear for any record with an attached media object.

5. Choose a record, and add an image object.

   a. Choose a record that does not have an attachment in the grid, and then choose Attachments.

   b. On Media Objects, add an Image object by choosing New and then Image from the File menu.

   c. On Select an Image, find and choose an image to attach.

   d. Choose Save & Exit from the File menu.

## Solution Explorer

# Exercise: Adding Favorites

### Business Purpose

You have several Tasks that you access regularly. Rather than choose them from the J.D. Edwards Menus Task View, you decide to set up Favorites for more efficient access. In your Favorites, you want to include the Object Management Workbench application, the Object Management menu, and the Security Maintenance menu.

### Objectives

- Add an application to the Favorites Task View
- Add a menu to the Favorites Task View
- View the selections that you added to the Favorites Task View

### ► Steps to complete

1. From Solution Explorer, access the J.D. Edwards Menus Task View.
2. Locate the Object Management Workbench application (P98220).
   a. Send this application to your Favorites task view.
3. Locate the Object Management menu (GH9081).
   a. Send this menu to your Favorites task view.
4. Locate the Security Maintenance menu (GH9052).
   a. Send this menu to your Favorites task view.
5. On the Toolbar, click the Task View button and choose Favorites to view your new list of favorite menu items.

# Exercise: Creating a New Task View

### Business Purpose

To access the Class Address Book and Item Master applications outside of Object Management Workbench, you need to add them to a task view. Although you could add them to an existing task view or add them to your Favorites, you can also create a custom task view for these applications. In this exercise, you will create a custom task view for the Class Address Book and Item Master applications.

### Objective

Create the Class Applications custom task view for the Class Address Book and Item Master applications.

▶ **Steps to complete**

1. In Solution Explorer, add a new task view using the following values:

   a. From the Tools menu choose Add New Task View.

| Field | Value |
|---|---|
|  | 03 |
|  | CLASS APPLICATIONS xx |
|  | Class Applications xx |
|  | Off |
|  | Off |

2. Use the Task View drop-down option on the toolbar to display your new task view.

**Note**
You might need to sign off of Solution Explorer and then sign on again to view your new task view in the Navigate To A Task View drop-down menu on the side bar.

## Exercise: Adding Tasks to Your New Task View

### Business Purpose

To access the Class Address Book and Item Master applications, you need to add them as tasks to a task view. In this exercise, add these applications as tasks to the Class Applications custom task view that you created in the previous exercise.

### Objective

Using the information given, add two new tasks to the Class Applications xx task view. The tasks should take the user to either the Class Address Book application or the Item Master application, depending on which one is selected.

### ▶ Steps to complete

1. In Solution Explorer, access the Class Applications xx task view.
2. Insert the Class Address Book application as a new executable task.
3. Enter the following values on the Common tab:

| Field | Value | Comment |
|---|---|---|
| Task Id | | Blank - the system supplies a default value |
| Task Name | Class Address Book | |
| Product Code | 55xx | |
| Activator Type | | <blank> |
| Client Platform | | <blank> |
| Jargon | | <blank> |
| Country Code | | <blank> |
| Required | | Off |
| Active | | On |

4. Fill out the Executable tab with the appropriate values.

   No entries are necessary on the Resource tab.

5. Insert the Class Address Book application as a new executable task a second time.
6. Enter the following values on the Common tab:

| Field | Value | Comment |
|---|---|---|
| Task Id | | Blank (the system supplies a default value) |
| Task Name | Class Address Book--Suppliers | |

| | | |
|---|---|---|
| Product Code | 55xx | |
| Activator Type | | <blank> |
| Client Platform | | <blank> |
| Jargon | | <blank> |
| Country Code | | <blank> |
| Required | | Off |
| Active | | On |

7. Fill out the Executable tab with the appropriate values.

   No entries are necessary on the Resource tab.

8. Insert the Item Master application as a new executable task.

9. Enter the following values on the Common tab:

| **Field** | **Value** | **Comment** |
|---|---|---|
| Task Id | | Blank (the system supplies a default value) |
| Task Name | Class Item Master | |
| Product Code | 56xx | |
| Activator Type | | <blank> |
| Client Platform | | <blank> |
| Jargon | | <blank> |
| Country Code | | <blank> |
| Required | | Off |
| Active | | On |

10. Fill out the Executable tab with the appropriate values.

    No entries are necessary on the Resource tab.

11. Test the tasks to verify that they function properly.

    For now, the Class Address Book and the Class Address Book – Suppliers tasks are identical

# Processing Options

# Directed Exercise: Creating Processing Options for Address Book (P5501xx)

### Business Purpose

Processing options control how an interactive or batch application processes data. You can attach unique processing options to different versions of the same application to control various options at runtime. For example, you can use processing options to set up default values for an application, to customize an application for different companies or users, to control the format of forms and reports, and so on. In this exercise, you will create a processing option that allows users to select a default search type for the Search Type filter on the Work With Addresses form (P5501xx). You will create a second processing option to automatically pass the value for supplier to the Search Type filter on the Work With Addresses form. Finally, you will create an interactive version for each processing option and then attach these versions to different tasks on the Class Applications task view.

### Objective

Upon completion of this exercise, you will be able to do the following:

- Create a data structure for a processing option template
- Add event rules to receive a default value from a processing option
- Create interactive versions for Blind Execution and for Prompt for Values
- Modify tasks to include processing options

### ► Steps to complete

1. Create the Processing Option data structure T5501xx:

   a. On Object Management Workbench, choose the Class Address Book Project and then click Add.

   b. Click the Data Structures option in the Object Librarian Objects group box.

   c. Enter the following values in the Add Objects window:

| Field | Value |
|---|---|
| Object Name | T5501xx |
| Description | Class Address Book Processing Options |
| Product Code | 55xx |
| Product System Code | 55xx |
| Object Use | 360 |
| Object Type | Processing Option Template |

   d. Click OK.

   e. Click the Design Tools tab and then click Start the Processing Option Design Aid.

f. On Processing Options Design, right-click <NewTab> and choose Current Tab Properties.

g. In the Short Name field, type Default.

   The Long Name field is Default Search Type.

h. In the Data Dictionary Browser portion of the window, type SAAT1 in the Alias column and press Enter.

i. When the data item appears, drag it to the left, under the Default tab.

j. Click Sample Application Search Type and change it to the following:

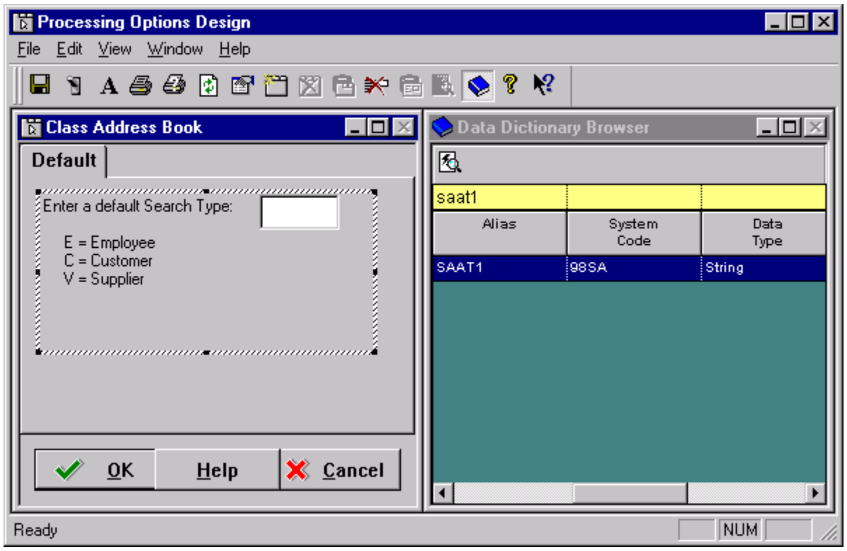
   Enter a default Search Type:

   E = Employee

   C = Customer

   V = Supplier

The following form illustrates the Processing Options Setup:



Enter a default Search Type:

E = Employee
C = Customer
V = Supplier

saat1

| Alias | System Code | Data Type |
|---|---|---|
| SAAT1 | 98SA | String |

k. Save and exit.

l. Click OK to return to Object Management Workbench.

   The T5501xx processing option data structure now appears as an object in the Class Address Book project.

2. Modify the Work With Addresses form in application P5501xx.

   a. On Object Management Workbench, choose the P5501 Class Address Book application in the Class Address Book project, and then click Design.

   b. Click the Design Tools tab, and then click Start Form Design Aid.

   c. From the File menu, choose Select Processing Options.

d. On Select Processing Options Template, type T5501xx in the Object Name field in the QBE line, and then click Find.

e. Choose T5501xx and click OK.

f. Choose the Work With Addresses (Find/Browse) form.

g. Under Form Event Rules, choose Post Dialog is Initialized and add the following event rules:

```
If PO szSASearchType is greater than <Blank>

FC Search Type = PO szSASearchType

End If
```

h. Save and exit the application.

3. Create a version of P5501xx that prompts the user to enter a default search type on the processing option form.

a. On Design Tools, click Version List.

b. On Work With Interactive Versions, click Add.

c. Enter the following values on the Version Add form:

| Field | Value |
|---|---|
| Version | VER0001xx |
| Version Title | Class Version --Prompt |
| Prompting Options | 2 |
| Security | 0 |

d. Click OK.

e. Click OK again on the Interactive Version Design form to return to the Work With Interactive Versions form.

The VER0001xx version that you just created appears as an entry in the grid.

4. Create a blind version of P5501xx that automatically assigns a V (for Supplier) as the Search Type.

a. On the Design Tools tab, click Version List.

b. On Work With Interactive Versions, click Add.

c. Enter the following values on the Version Add form:

| Field | Value |
|---|---|
| Version | VER0002xx |
| Version Title | Class Version – Suppliers (Blind) |

| | |
|---|---|
| Prompting Options | 1 |
| Security | 0 |

d. Click OK.

e. On the General tab, click the Processing Options button.

The Processing Options window appears.

f. Enter V as a default for the Search Type field.

g. Click OK.

h. Click OK again on Interactive Version Design to return to the Work With Interactive Versions form.

The VER0002xx version that you just created appears as an entry in the grid.

5. Test the VER0002xx blind version.

a. On Work With Interactive Versions, choose the VER0002xx version.

b. Choose Run from the Row menu.

The Work With Addresses form appears with V in the Search Type field.

**Note**
You will not be able to test the VER0001xx prompt version until it is associated with a task.

c. Click Close.

d. On Work With Interactive Versions, click Close.

e. On Interactive Application Design, click OK to return to Object Management Workbench.

The versions that you created appear as objects in the Class Address Book project.

f. Click Close to return to Solution Explorer.

6. Add the VER0001xx prompt version to the Class Address Book task in the Class Applications task view.

a. In Solution Explorer, access the Class Applications task view.

b. Choose the Class Address Book task, right-click, and choose Task Revisions.

7. On Task Revisions, click the Executable tab and then enter the following information:

| Field | Value | Comment |
|---|---|---|
| Application | P5501xx | |
| Version | VER0001xx | |
| Form | | <blank> |
| Option Code | 3 | |
| Form Mode | | <blank> |
| Application Type | | <blank> |

   a. Click OK to return to the Class Applications task view.

   b. Test the Class Address Book task by double-clicking it.

   c. The Processing Options window appears, prompting you to add a Search Type.

   d. Enter a Search Type and click OK.

   e. The Work With Addresses form appears with the search type value that you entered in the Processing Options window.

8. Add the VER0002xx version to the Class Address Book – Suppliers task in the Class Applications task view.

   a. In Solution Explorer, access the Class Applications task view.

   b. Choose the Class Address Book – Suppliers task, right-click, and choose Task Revisions.

   c. On Task Revisions, click the Executable tab, and then enter the following information:

| Field | Value | Comment |
|---|---|---|
| Application | P5501xx | |
| Version | VER0002xx | |
| Form | | <blank> |
| Option Code | 1 | |
| Form Mode | | <blank> |
| Application Type | | <blank> |

   d. Click OK to return to the Class Applications task view.

   e. Test the Class Address Book – Suppliers task by double-clicking it.

   f. The Work With Addresses form appears with V in the Search Type field.

# Exercise: Creating Processing Options for Class Item Master (P5601xx)

### Business Purpose

Create a processing option to allow users to select a default item type to pass to the Item Type filter on the Work With Item Master Information Find/Browse form (P5601xx).

### Objective

Upon completion of this exercise, you will be able to do the following:

- Create a processing option template data structure
- Add event rules to receive a default value from a processing option.
- Create interactive versions for Blind Execution and for Prompt for Values
- Modify tasks to include processing options

### ► Steps to complete

1. Create a data structure for a processing option template to pass a default item type (SATYPE) to the P5601xx application.

2. Modify the Work With Item Master Information form in application P5601xx, adding event rules to receive the default value from the processing option.

3. Create a blind version of P5601xx that automatically assigns a value as the item type.

4. Add the blind version that you created to the Item Master task in the Class Applications task view.

5. Test the processing option.

## Optional Exercises

If time permits, complete these additional steps to create an additional task and processing option for the Class Item Master. This task will allow the user to select the item type before accessing the Class Item Master application.

### ▶ Steps to complete

1. Add another processing option for the Class Item Master application that prompts the user to enter a value for the item type on the Work With Item Master Information form.

2. Add a task to the Class Applications task view for the processing option that you just created. Name it Class Item Master – Select Type.

3. Test the version to verify that it functions properly.

4. Revise the task to use Option Code 2 (Prompt for Version) rather than Option Code 3 (Prompt for Values).

5. Double-click the task to test how it functions with the Prompt for Version option code.

## Radio Buttons

# Directed Exercise: Creating Radio Buttons – P5501xx (Fix/Inspect)

## Business Purpose

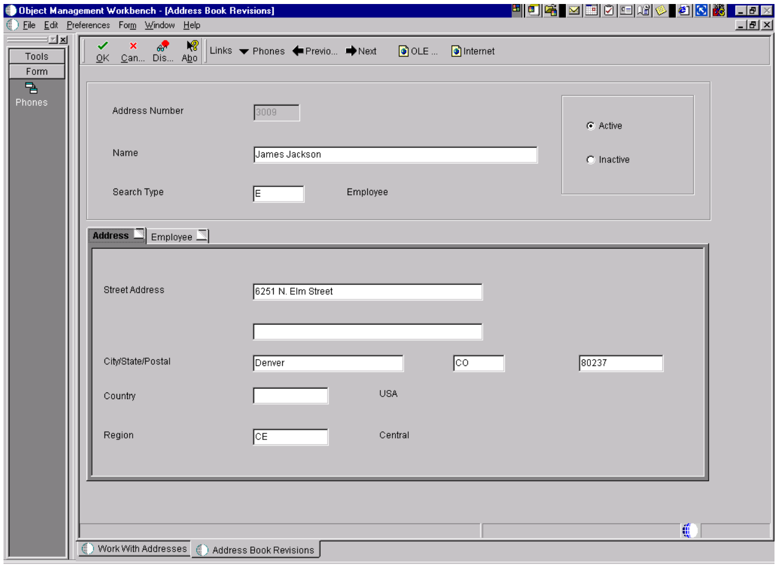
Create radio buttons on the P5501xx Address Book Revisions form to indicate whether an employee, customer, or vendor is considered active or inactive.

## Objective

Upon completion of this exercise, you will be able to do the following:

- Add radio buttons and group boxes
- Associate radio buttons to the appropriate business view item

Radio buttons on the Address Book Revisions Form:



Address Number 3009

Name James Jackson

Search Type E Employee

● Active

○ Inactive

Street Address 6251 N. Elm Street

City/State/Postal Denver CO 80237

Country USA

Region CE Central

## ▶ Steps to complete

Radio buttons are used to indicate choices. The radio buttons in a group box are mutually exclusive, meaning that only one can be selected at a time. Insert two radio buttons on the P5501xx Address Book Revisions (Fix/Inspect) form.

1. On Object Management Workbench, choose the Class Address Book P5501xx application in the Class Address Book Project, and then click Design.

2. On the Design Tools tab, click Start Form Design Aid.

   a. Choose the Address Book Revisions (Fix/Inspect) form.

   b. From the Insert menu, choose Radio Button.

   c. Position the radio button as shown on the form.

   d. Repeat steps a and b for the second radio button.

3. Insert a group box.

    a. Use the pencil tool to draw a rectangle around the two radio buttons, click Insert again, and then click Group Box.

    b. Access Group Box Properties and remove the value from the Group Box Title field.

4. Associate the radio buttons with database item Status Active/Inactive Flag (SASTATxx), and change the title and value for each radio button.

    a. Choose the first radio button and access Properties.

    b. Choose the Data Item tab and click the Data Items tab, find SASTATxx data item and double click. SASTATxx should appear in Data Item Information.

    c. Choose the General tab and check Override Text and type Active.

    d. Choose the Value tab and type A.

    e. Click OK.

    f. Repeat the steps above for the second data item.

        i. Override Text = Inactive

        ii. Value = I

5. From the Form menu choose Create and click Message Form

    a. Go to Properties for the message box.

        i. Title = WARNING:

    b. Type in the message *The selected record is in an inactive state* in the text box.

    c. Select the Fix/Inspect form.

        i. Right-click the form and choose Event Rules

        (a) On the Post Dialog Is Initialized event add the following event rule.

```
If BC Status Active/Inactive (F5501xx)Flag is equal "I"
       Call( App:P5501xx , Form: W5501xD)
End If
```

6. Save and exit the P5501xx application.

7. Run the application to verify that the radio buttons function properly.

# Exercise: Creating Radio Buttons – P5601 (Fix/Inspect)

### Business Purpose

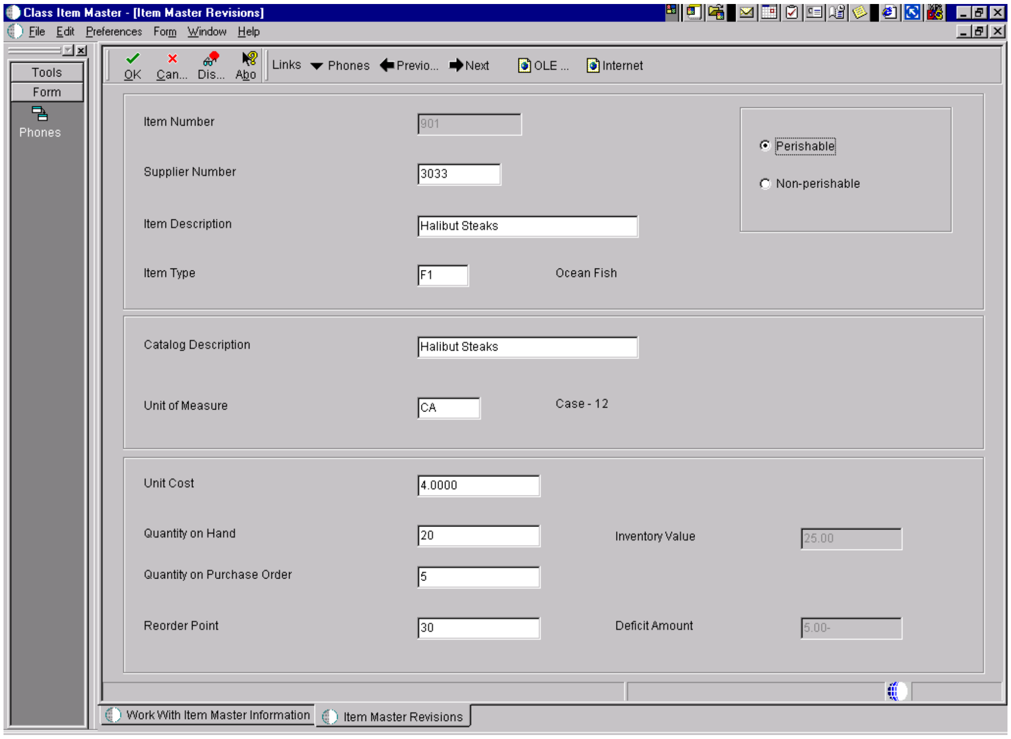
Create radio buttons on the Item Master Revisions form (P5601) that indicate whether an item is perishable or non-perishable.

### Objective

Upon completion of this exercise, you will be able to do the following:

- Add radio buttons and group boxes
- Associate radio buttons to the appropriate business view item

Radio Buttons on the Item Master Revisions Form:



Class Item Master - [Item Master Revisions]
File Edit Preferences Form Window Help
Tools
Form
Phones
OK
Can...
Dis...
Abo
Links
Phones
Previo...
Next
OLE ...
Internet
Work With Item Master Information
Item Master Revisions

| | |
|---|---|
| Item Number | 901 |
| Supplier Number | 3033 |
| Item Description | Halibut Steaks |
| Item Type | F1 Ocean Fish |

● Perishable
○ Non-perishable

| | |
|---|---|
| Catalog Description | Halibut Steaks |
| Unit of Measure | CA Case - 12 |

| | | | |
|---|---|---|---|
| Unit Cost | 4.0000 | | |
| Quantity on Hand | 20 | Inventory Value | 25.00 |
| Quantity on Purchase Order | 5 | | |
| Reorder Point | 30 | Deficit Amount | 5.00- |

### ▶ Steps to complete

Radio buttons are used to indicate choices. The radio buttons in a group box are mutually exclusive, meaning that only one can be selected at a time.

1. Insert two radio buttons on the Item Master Revisions (Fix/Inspect) form (P5601xx).

2. Insert a Group Box around the two radio buttons.

3. Associate the radio buttons with database item Item Storage (SASTRITMxx) and change the title and value for each radio button.

**Note**
Valid values for SASTRITMxx are P and N.

4. Save and exit the P5601xx application.
5. Run the application to verify that the radio buttons function properly.

# Final Lab

## Designing a Purchase Order Application

### Business Purpose

The purpose of this lab is to develop an application from beginning to end to integrate and reinforce the skills that you have learned in the previous exercises. You will analyze the design specifications and formulate a plan to build an integrated application. Be prepared to explain your design decisions and the reasoning behind them.

### Objective

Given the following information, design and build a purchase order application that allows users to find, select, add, and delete purchase orders. Ensure that the forms in the application look like the samples provided, that the application functions as described in the specifications, and that it runs without errors.

### Before You Begin

In the previous exercises, you created projects, data dictionary items, tables, and business views. You designed various application forms that included event rule logic. To complete this lab, you will use this experience to build a simple purchase order application. The Purchasing department has provided you with the following information:

- Database specifications
- Overview of the desired features
- Prototype form formats

Review the specifications before you develop the application. If no specific information is provided, follow documented standards.

# Exercise: Adding a Project

### Business Purpose

All development in the software is done in the context of a project. All of the objects needed for a given development process should be included in a separate project. Because you will be developing an application in the exercises that follow, you need to create a project to hold the objects that make up the Class Purchase Orders application.

### Objective

Given the following information, create a project in Object Management Workbench. Your project should be named according to naming conventions. When creating the project, provide additional details where appropriate.

### ▶ Steps to complete

1. Use Object Management Workbench to add a project for the Class Purchase Orders application.

   a. When naming the project, follow the standard naming conventions.

   b. Provide a meaningful description, in addition to any other helpful information about the project.

2. Add yourself as a Developer to the project.

3. Promote your project to status of 21.

# Exercise: Creating Tables and Business Views

### Business Purpose

The Class Purchase Orders application requires two tables, one for the purchase order header information and one for the detail information about each item ordered.

### ▶ Steps to complete

Become familiar with your data by reviewing the attributes and attached triggers for each data item.

1. List the attributes and triggers for each data item in the Comment column.

All of the data dictionary items that you need are already in the data dictionary. These items are listed below in the table specifications.

| Alias | Data Item | Comment |
|---|---|---|
| SADOCO | SADocumentNumber | |
| SADCTO | SAOrderType | |
| KCOO | CompanyKeyOrderNo | |
| SAAN8Axx | SAAddressNumberAxx | |
| SHAN | AddressNumberShipTo | |
| SASRST | SAPOStatusCode | |
| SAADDJ | SADateShip | |
| SATRDJ | SAOrderDate | |
| SADRQJ | SADateRequested | |
| AG | AmountGross | |
| SADOCO | SADocumentNumber | |
| SADCTO | SAOrderType | |
| KCOO | CompanyKeyOrderNo | |
| SALNID | SALineNumber | |
| SAITM | SAItemNumber | |
| SAUORG | SAOrderUnits | |
| SAPRRC | SAAmountUnit Cost | |
| SATRDJ | SAOrderDate | |
| SAADDJ | SADateShip | |
| SADRQJ | SADateRequested | |

| Alias | Data Item | Comment |
|---|---|---|
| USER | UserId | |
| PID | ProgramId | |
| JOBN | WorkStationId | |
| UPMJ | DateUpdated | |
| UPMT | TimeLastUpdated | |

**Hint**
You might have to override some of the data dictionary attributes or triggers for the data items in your application.

2. Use Object Management Workbench to add the F5701xx and F5711xx tables with the data dictionary items shown below.

### Purchase Order Header Table

| Field | Value |
|---|---|
| Object Name | F5701xx |
| Description | Class Purchase Order Header |
| Product Code | 57xx |
| **Product System Code**<br>**Product System Code** | 57xx |
| Object Use | 210 |
| Column Prefix | PH |

### Data Dictionary Items for the Purchase Order Header Table

| | Alias | Data Item |
|---|---|---|
| 1 | SADOCO | SADocumentNumber |
| 2 | SADCTO | SAOrderType |
| 3 | KCOO | CompanyKeyOrderNo |
| 4 | SAAN8A | SAAddressNumberA |
| 5 | SHAN | AddressNumberShipTo |

|  |  |  |
|---|---|---|
| 6 | SASRST | SAPOStatusCode |
| 7 | SAADDJ | SADateShip |
| 8 | SATRDJ | SAOrderDate |
| 9 | SADRQJ | SADateRequested |
| 10 | AG | AmountGross |
| 11 | USER | UserId |
| 12 | PID | ProgramId |
| 13 | JOBN | WorkStationId |
| 14 | UPMJ | DateUpdated |
| 15 | UPMT | TimeLastUpdated |

Primary Unique Index: Order Number, Order Type, Order Company

### Purchase Order Detail Table

| Field | Value |
|---|---|
| Object Name | F5711xx |
| Description | Class Purchase Order Detail |
| Product Code | 57xx |
| Product System Code | 57xx |
| Object Use | 230 |
| Column Prefix | PD |

### Data Dictionary Items for the Purchase Order Detail Table

|  | Alias | Data Item |
|---|---|---|
| 1 | SADOCO | SADocumentNumber |
| 2 | SADCTO | SAOrderType |
| 3 | KCOO | CompanyKeyOrderNo |
| 4 | SALNID | SALineNumber |
| 5 | SAITM | SAItemNumber |
| 6 | SAUORG | SAOrderUnits |
| 7 | SAPRRC | SAAmountUnit Cost |
| 8 | SATRDJ | SAOrderDate |
| 9 | SAADDJ | SADateShip |
| 10 | SADRQJ | SADateRequested |

| | | |
|---|---|---|
| 11 | USER | UserId |
| 12 | PID | ProgramId |
| 13 | JOBN | WorkStationId |
| 14 | UPMJ | DateUpdated |
| 15 | UPMT | TimeLastUpdated |

Create a Primary unique index for the F5711xx table, using the data items that represent order number, order type, order company, and line number as the primary unique identifier. Use naming standards for the index.

3. Create the three necessary business views over each table. Be sure to follow naming standards. Create a separate business view for the Item Master Search & Select form.

### Questions to Ask

1. Do you need to create a separate business view for each form?

2. Which form will have two business views attached?

3. Can you use the same business view for the Work With Purchase Orders and the Purchase Order Revisions forms?

# Exercise: Creating the Work With Purchase Orders Form

### Business Purpose

The Class Purchase Orders application requires a form that will serve as an entry point to the application. This form allows the user to search for purchase orders, select an individual purchase order, or delete a purchase order. If the user selects a specific purchase order record, the form exits to the Purchase Order Revisions form, where the user can add, update, or view the individual purchase order.

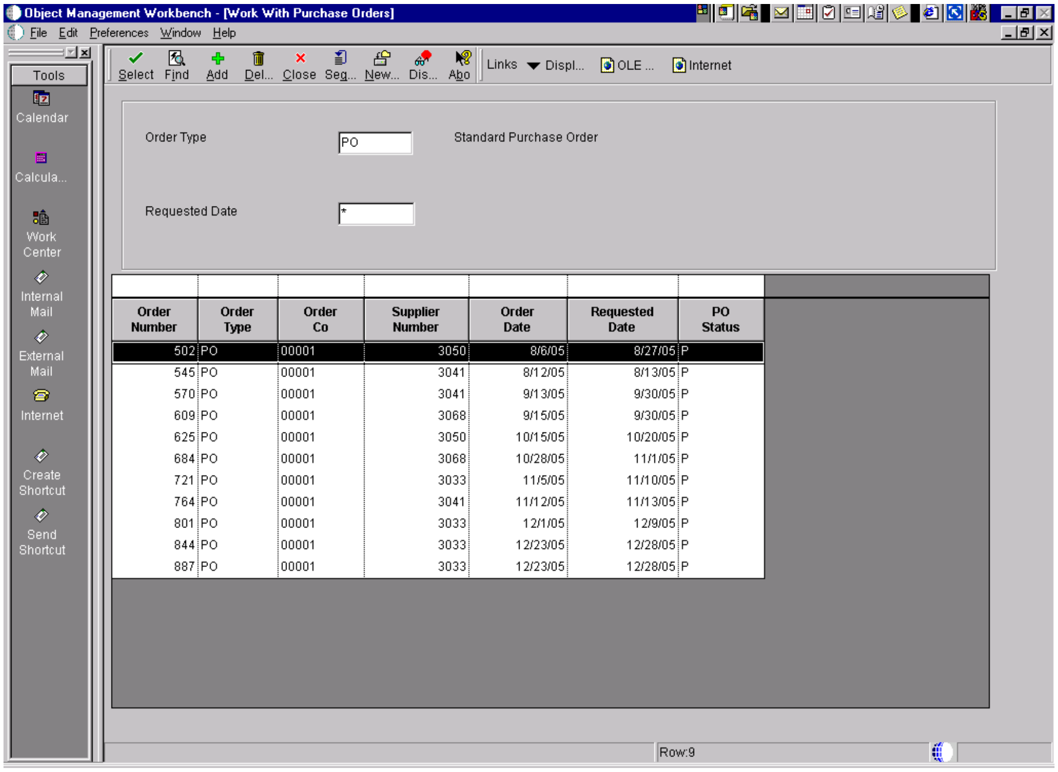
### Objective

Given the following specifications, create a form that looks like the form illustration in this exercise. The form should adhere to all standards for form design.

### Specifications

- This form should be the entry point into the application.
- Users should be able to find, select, add, and delete purchase orders from the form.
- Order Type and Date Requested should be used as filter fields, and the description for an Order Type code should also appear on the form.
- The grid should contain the Order Number, Order Type, Order Company, Address Number, Order Date, Date Requested, and Order Status fields.
- The grid should be sorted by the Order Type and Order Number fields.
- The Add button will call the Purchase Order Revisions form, created in the next exercise and pass no parameters.
- The Select button should make a modeless call to the Purchase Order Revisions form, created in the next exercise and pass the order number, order type, and order company as parameters.
- The form should be designed in accordance with the form illustration, and it should adhere to all form standards.
- When you complete the form, the interactive application P5701xx Class Purchase Orders should appear under the Objects node of the Class Purchase Orders project.

P5701xx Work With Purchase Orders  (Find/Browse):



Order Type PO Standard Purchase Order

Requested Date *

| Order Number | Order Type | Order Co | Supplier Number | Order Date | Requested Date | PO Status |
|---|---|---|---|---|---|---|
| 502 | PO | 00001 | 3050 | 8/6/05 | 8/27/05 | P |
| 545 | PO | 00001 | 3041 | 8/12/05 | 8/13/05 | P |
| 570 | PO | 00001 | 3041 | 9/13/05 | 9/30/05 | P |
| 609 | PO | 00001 | 3068 | 9/15/05 | 9/30/05 | P |
| 625 | PO | 00001 | 3050 | 10/15/05 | 10/20/05 | P |
| 684 | PO | 00001 | 3068 | 10/28/05 | 11/1/05 | P |
| 721 | PO | 00001 | 3033 | 11/5/05 | 11/10/05 | P |
| 764 | PO | 00001 | 3041 | 11/12/05 | 11/13/05 | P |
| 801 | PO | 00001 | 3033 | 12/1/05 | 12/9/05 | P |
| 844 | PO | 00001 | 3033 | 12/23/05 | 12/28/05 | P |
| 887 | PO | 00001 | 3033 | 12/23/05 | 12/28/05 | P |

# Exercise: Creating the Purchase Order Revisions Form

### Business Purpose

In previous exercises, when you wanted to revise a record in an application, you used a Fix/Inspect form. In the Class Purchase Order application, you use the Header Detail form for this purpose. A Header Detail form allows you to work with data from two separate tables. You can use this form to add or update a single header record. You can also add, update, or delete multiple detail records from the same form.

Because the Header Detail form allows you to update or add records from two different tables, you can attach two business views to the form. Attach one business view to the grid and the other to the form itself, thereby updating both tables from a single form. The Header Detail form is often used for one-to-many relationships, which is how you will be using it in the Class Purchase Order application.

### Objective

Given the following specifications, design a Header Detail form that looks like the form illustration in this exercise. The form design should adhere to all Header Detail form standards.

### Specifications

The Header area of the form should do the following:

- Allow users to work with purchase order header information (F5701xx)
- Allow users to add or modify purchase order information and delete individual purchase order lines
- Include the following business view data items:

| | Alias | Data Item |
|---|---|---|
| 1 | SADOCO | SADocumentNumber |
| 2 | SADCTO | SAOrderType |
| 3 | KCOO | CompanyKeyOrderNo |
| 4 | SAAN8Axx | SAAddressNumberAxx |
| 5 | SHAN | AddressNumberShipTo |
| 6 | SASRST | SAPOStatusCode |
| 7 | SAADDJ | SADateShip |
| 8 | SATRDJ | SAOrderDate |
| 9 | SADRQJ | SADateRequested |
| 10 | AG | AmountGross |

- Disable the Order Number field for input
- Display the description for the Order Status code field
- Supply today's date as the default value for the Order Date field on the header portion of the form if the user leaves this field blank (null)

- Have Date Requested as a required entry field
- Disable Next Numbering for Address (Supplier) Number and Ship To Address Number

The Grid area of the form should do the following:

- Allow users to work with the Purchase Order Details table (F5711xx)
- Include the following data items:

| | **Alias** | **Data Item** |
|---|---|---|
| 1 | SADOCO | SADocumentNumber |
| 2 | SADCTO | SAOrderType |
| 3 | KCOO | CompanyKeyOrderNo |
| 4 | SALNID | SALineNumber |
| 5 | SAITM | SAItemNumber |
| 6 | SAUORG | SAOrderUnits |
| 7 | SAPRRC | SAAmountUnit Cost |
| 8 | SATRDJ | SAOrderDate |
| 9 | SAADDJ | SADateShip |
| 10 | SADRQJ | SADateRequested |

- Sort in ascending order by Line Number
- Automatically increment the line number and disable it for input
- Have a visual assist attached to the Item Number field, which calls the Item Master Search & Select form that appears in this exercise

**Note**

You must complete the next exercise, Creating the Item Master Search & Select Form, before you can attach the form as a visual assist.

- Calculate extended cost by multiplying the Quantity Ordered by the Unit Cost

**Hint**

This calculation will need to be done for multiple events.

- Disable the Extended Cost field for input
- Have the Order Number, Order Type, and Order Company fields hidden in the grid after you test the form
- Have an Auto Find on Entry option in the grid because the toolbar does not contain a Find button

### Header/Detail Join

For the Header Detail form to function properly, the header and detail portions need to be joined using the following steps.

**▶ Steps to complete**

1. On the Header form, set up the key fields Order Number, Order Type, and Order Company as filter fields that are set to =.

   Do not make these fields wildcards. Also, these fields must in the grid (they will be hidden grid fields).

   **Note**
   If you do not make these header form fields filter fields, then the grid always selects all of the detail records in the F5711xx table, instead of the detail records that match the key fields in the header.

2. Turn on the grid option Automatically Find On Entry.

3. Use assignments to pass values for the Order Number, Order Type, and Order Company fields from the header area of the form to the same (hidden) fields in the grid.

   a. On the Row Exit & Changed – Inline event, create the following assignments:

```
GC OrderNumber = FC Order Number

GC OrTy = FC Order Type

GC OrderCo = FC Order Company
```

4. Assign the values of the keys located in the header to the business view before the record is added to the database.

   **Caution**
   When you make the following assignments, the fields in the F5711xx table might appear first in the list, followed by the fields in the F5701xx table.

   BC Sample Application - Document Number (F5711) might appear first, followed by BC Sample Application - Document Number (F5701).

   The assignments will not work if (F5711xx) is chosen.

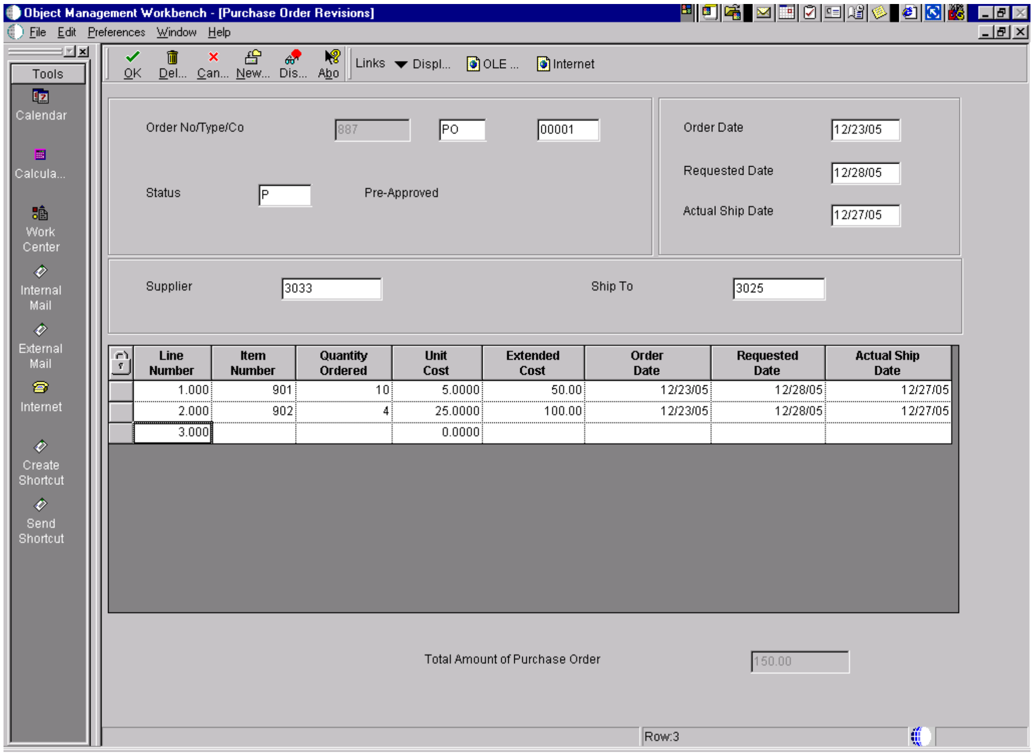
   a. On the Add Record to DB – Before event, create the following assignments:

```
BC Sample Application – Document Number (F5701) = FC Order Number

BC Sample Application  - OrderType (F5701) = FC Order Type

BC OrderCompany (Order Number) (F5701) = FC Order Company
```

P5701  Purchase Order Revisions (Header Detail) in Add Mode:



| | | | |
|---|---|---|---|
| Order No/Type/Co | 887 | PO | 00001 |
| Status | P | Pre-Approved | |

| | |
|---|---|
| Order Date | 12/23/05 |
| Requested Date | 12/28/05 |
| Actual Ship Date | 12/27/05 |

| | | | |
|---|---|---|---|
| Supplier | 3033 | Ship To | 3025 |

| Line Number | Item Number | Quantity Ordered | Unit Cost | Extended Cost | Order Date | Requested Date | Actual Ship Date |
|---|---|---|---|---|---|---|---|
| 1.000 | 901 | 10 | 5.0000 | 50.00 | 12/23/05 | 12/28/05 | 12/27/05 |
| 2.000 | 902 | 4 | 25.0000 | 100.00 | 12/23/05 | 12/28/05 | 12/27/05 |
| 3.000 | | | 0.0000 | | | | |

Total Amount of Purchase Order 150.00

Row:3

# Exercise: Creating the Item Master Search & Select Form

### Business Purpose

In the process of creating a purchase order, you need to enter an item number for each item that you are ordering. Because it is unlikely that you will know all the item numbers in the Item Master application, you can create an Item Master Search & Select form and attach it to the Item Number data item. In this way, any time that you encounter the Item Number field on a form or in a grid, you can use the visual assist, which takes you to the Search & Select form to search for an item. When you choose an item from the Search & Select form, it automatically displays the item number on the calling form.

### Objective

Given the following specifications, create a form that looks like the form illustration in this exercise. The form should adhere to all form design standards.

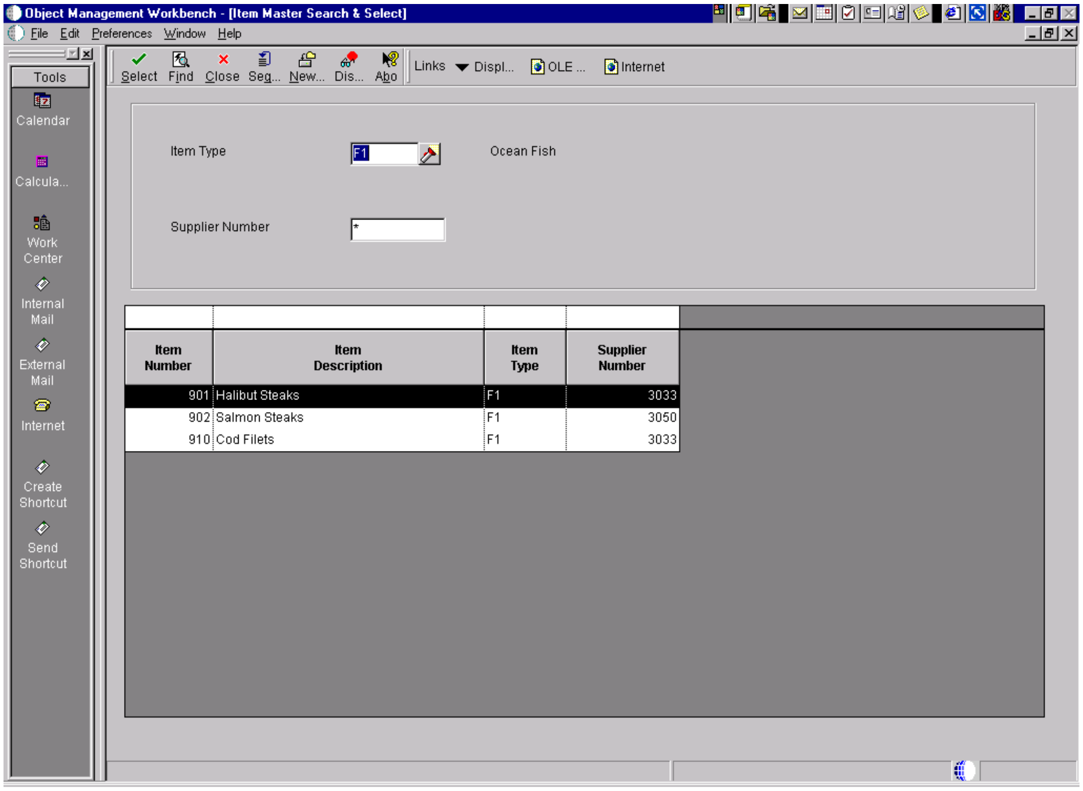
### ▶ Steps to complete

1. Add an interactive application object to the Class Purchase Orders project.

2. Create the Item Master Search & Select form.

3. Attach the Item Master Search & Select form to data dictionary item SAITM.

P5601Sxx Item Master Search and Select:


Object Management Workbench - [Item Master Search & Select]
File Edit Preferences Window Help
Select Find Close Seq... New... Dis... Abo
Links Displ... OLE ... Internet
Tools
Calendar
Calcula...
Work Center
Internal Mail
External Mail
Internet
Create Shortcut
Send Shortcut

| | | |
|---|---|---|
| Item Type | F1 | Ocean Fish |
| Supplier Number | * | |

| Item Number | Item Description | Item Type | Supplier Number |
|---|---|---|---|
| 901 | Halibut Steaks | F1 | 3033 |
| 902 | Salmon Steaks | F1 | 3050 |
| 910 | Cod Filets | F1 | 3033 |

## Exercise: Advanced Functionality

Complete the following additional steps.

► **Steps to complete**

1. Calculate the Total Amount of Purchase Order as the sum of the Extended Cost for all purchase order lines that appear on the form, and then disable the field for input.

   **Hint**
   This calculation must be performed on multiple events. Consider what will happen if a user changes the figures in any of the controls that make up the extended cost for any of the items on the purchase order.

2. Disable the FC Order Type and FC Order Company fields when the form is called in an update mode.

   **Note**
   FC Order Number has already been disabled at the control level.

3. For the Order Date, Requested Date, and Actual Ship Date fields in the grid, use the header dates as the default values if the user leaves these fields blank (null).

4. Add the Class Purchase Orders application to the Class Applications task view.

5. Create a processing option to allow users to choose a default order type to pass to the order type filter on the Work With Purchase Order Find/Browse form (P5701xx).

6. Attach media objects to the grid of the Work With Purchase Orders Find/Browse form.