Meta5

Forms User's Guide

Version 4.3

Note

Before using this information and the product it supports, be sure to read the general information under "Notices".

Third Edition (January 2013)

This edition applies to Version 4 Release 3 Modification Level 3 of Meta5, and to any subsequent releases until otherwise indicated in new editions or technical newsletters. Make sure you are using the correct edition for the level of the product.

Order publications through your Meta5 representative or the Meta5 branch office serving your locality.

Contents

Chapter 19. Enhancing Forms with Design Techniques 145

Chapter 20. Becoming an Expert Forms Designer 155

Chapter 21. Designing Complex Forms 171

Chapter 22. Improving Security and Performance 181

Appendix A. Identifying Keywords and Statements 187

Glossary 193

Meta5 Publications 199

Index 201

Notices

References in this publication to Meta5 products, programs, or services do not imply that Meta5 intends to make these available in all countries in which Meta5 operates. Any reference to a Meta5 product, program, or service is not intended to state or imply that only that Meta5 product, program, or service may be used. Subject to Meta5's valid intellectual property or other legally protectable rights, any functionally equivalent product, program, or service may be used instead of the Meta5 product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by Meta5, are the responsibility of the user.

Meta5 may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

Meta5, Inc.
122 West Main Street
Babylon, NY 11702
U.S.A.

Licensees of this program who wish to have information about it for the purpose of enabling (1) the exchange of information between independently created programs and other programs (including this one) and (2) the mutual use of the information that has been exchanged, should contact:

Meta5, Inc.
122 West Main Street
Babylon, NY 11702
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

Various components of the Meta5 software system require the presence of one or more third party software products, including but not limited to Microsoft Excel for Meta5's Xlaunch, Xtract software product, Microsoft Word for Meta5's WordTool software product, Microsoft Outlook for Meta5's SendOut software product, Brio’s BrioQuery for Meta5’s AutoBrio software product, Business Objects’ BusinessObjects for Meta5’s BOConnect software product and/or IBM's Lotus Notes for SendNts software product. No license is granted by Meta5 to our customers for the use of these third party programs in conjunction with our software products. Furthermore, it is the obligation of our customers to ascertain whether it has sufficient licenses from these third parties, for the third party software in order to utilize the above mentioned Meta5 software programs.

Trademarks

The following terms are trademarks of Meta5, Inc. in the United States or other countries or both:

Meta5

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Microsoft, Windows, Windows2000, Windows 98, Windows ME, Windows XP, Windows NT, Excel, Word, Outlook and the Windows 95 logo are trademarks or registered trademarks of Microsoft Corporation.

Brio and Brio Query are registered trademarks of Brio Software Inc.

BusinessObjects is a trademark of Business Objects SA.

Lotus Notes and Lotus 123 are Trademarks of Lotus Software, a subsidiary of IBM Corporation.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Other company, product, and service names, which may be denoted by a double asterisk (**), may be trademarks or service marks of others.

About This Book

This book contains information about the components, features, and functions of the Data Entry and Forms Design tools.

Who Should Read This Book

This book is intended for users who enter information into a database or a capsule application (Data Entry tool users) and for users who create data entry forms (Forms Design tool users).

For Data Entry tool users, this book assumes that you are familiar with the developer's desktop, and that you are able to perform basic functions within the Meta5 environment.

For Forms Design tool users, this book assumes that you have a working knowledge of:

- The Meta5 desktop environment
- The Data Entry tool
- SQL language

For more information about the developer's desktop, see *Getting Started with the Meta5 Developer's Desktop*. For more information about building and maintaining capsule applications, see the *Capsule User's Guide*.

We'd Like Your Comments

Your feedback is important to us in providing you with the most accurate and high-quality information. If you have any comments about this book or any other Meta5 documentation, please visit our Website at:

```
http://www.meta5.com
```

There you'll find a feedback page where you can enter comments and send them to us.

Part 1. Using the Data Entry Tool

In general, data entry is the process of entering information into a database, and at some point, retrieving the information. Within Meta5, the Data Entry tool is most often used to control a capsule application.

With the Data Entry tool, you enter and retrieve information through a Data Entry form. This section provides examples of Data Entry forms, and gives an overview of how Data Entry forms are used within Meta5.

The Data Entry tool comprises Data Entry forms that allow you to interact with a database or capsule application. You enter, retrieve, and delete information from within Data Entry forms.

Chapter 1. Getting Started

With the Data Entry tool, you enter and retrieve information through a Data Entry form. This chapter describes:

- Using Data Entry forms
- Finding the Data Entry icon
- Viewing the Data Entry window
- Understanding the features of the Data Entry window

Use the Data Entry tool to enter information into a database or capsule application, retrieve this information, and make changes to the data.

Using Data Entry Forms

The Data Entry tool is used to enter, retrieve, delete, and update information in a database table, or in the User Input Control window of a capsule icon. Some Data Entry tools can be used only to enter information, while others can be used to perform all functions. The Data Entry tool can comprise the following forms:

- Entry form

  Use this form to enter information into a database table or into the User Input Control window of a capsule icon.
- Retrieval form

  Use this form to view and to delete specific items.
- Update form

  Use this form to modify and to delete information.

Your Data Entry icon must be specifically enabled by the forms designer to allow you to delete information from a database.

Data Entry icons can contain one, two, or all three forms, depending on the intended use of the Data Entry icon.

Finding a Data Entry Icon

The forms designer, possibly your database administrator or system administrator, creates Data Entry icons using the Forms Design tool. The forms designer creates each Data Entry icon to be used with a specific table within a given database and category. The Data Entry icon is also often created to set @-variables within a capsule application. In these cases, the Data Entry icon might not always access a database.

The Data Entry icon displays a small picture of a form, followed by a name. It is unlikely that any of your Data Entry icons have the name Data Entry. The forms designer or system administrator has probably named the Data Entry icons to describe their subject matter.

Data Entry icons do not necessarily exist in a special file drawer in New Icons. If you have trouble locating a Data Entry icon, consult your forms designer, system administrator, or applications developer for assistance.

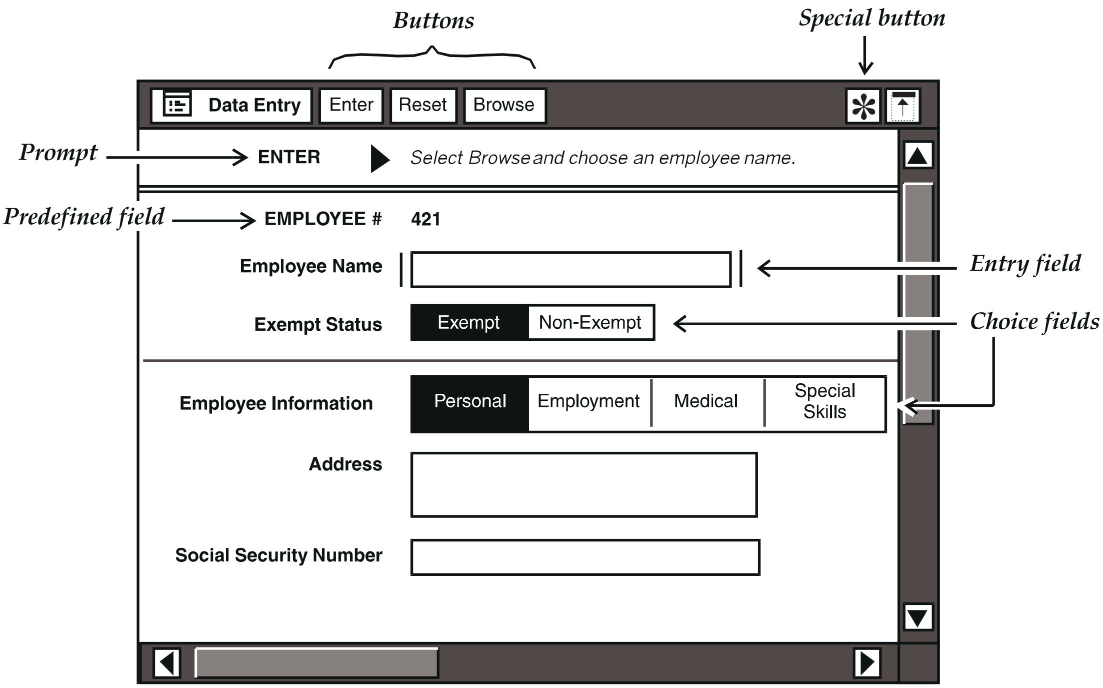
Viewing the Data Entry Window

Each form is displayed individually. View the Data Entry window by opening the Data Entry icon.

When a form is displayed, blinking lines are displayed at each end of the first field where you can enter or change information.

Learning About the Data Entry Window

There are several parts to the Data Entry form that indicate the type of form, allow you to enter information, or enable you to perform distinct actions. Figure 1 shows these parts of the Data Entry form.



ENTER ▶ *Select Browse and choose an employee name.*

EMPLOYEE # 421

Employee Name

Exempt Status: Exempt | Non-Exempt

Employee Information: Personal | Employment | Medical | Special Skills

Address

Social Security Number

Figure 1. A sample Data Entry form

The Data Entry form you use might not include the buttons, prompt, and fields shown in Figure 1.

Looking at the Prompt

Determine which form you are in by looking at the prompt. The prompt for an entry form reads `ENTER`, the prompt for a retrieval form reads `RETRIEVE`, and the prompt for an update form reads `UPDATE`.

The prompt might be followed by a field prompt that contains a message or comment provided to help you with entering data for the selected field. The following example shows a prompt for an `Employee Name` field.

```
ENTER>           Select Browse and choose an employee name.
```

Understanding the Data Entry Fields

The lower part of the window contains the form itself. Figure 1 on page 4 shows three basic types of labeled fields:

- Predefined field

  This information can be database information or comments the forms designer has provided to assist you. You cannot edit the information in this type of field.

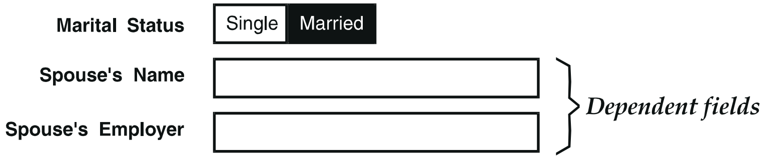
- Entry field

  Type information into this field or select an entry from a *browse list*. A browse list contains valid entries for the field from which you can choose. The `Browse` button is displayed after the field is selected.

- Choice field

  Select one of several choices from this type of field, rather than enter data. Choices can display *dependent fields*, or fields that are displayed only when a particular choice is selected. For example, the `Marital Status` field has two choices.

When `Single` is chosen, there are no dependent fields. However, when `Married` is chosen, more fields are displayed:



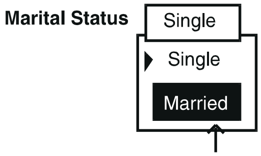
Dependent fields

The `Spouse's Name` and `Spouse's Employer` fields are dependent fields. These fields are displayed only if `Married` is chosen from the `Marital Status` field. If you enter information into these fields, you must click on the `Enter` button to verify the entries while the `Married` choice is selected.

Some fields also have a symbol (§) in front of them. These fields are known as *section fields*.

This symbol indicates that you can enter information into every dependent field of the section field, click on the `Enter` button, and verify all entries after regardless of which choice is currently selected.

Some choice and section fields are displayed as pull-down menus. In other words, when you select the visible choice, a menu is displayed from which you can select another choice.



The arrow to the left of a choice points to whichever choice is the current visible selection within the field. When a different selection is made, the field is displayed again, showing only the new choice. Highlight the choice you want with the pointer and release the mouse button.

Using the Window Header Buttons

The window header buttons change depending upon the Data Entry form you are using. Table 1 gives a description of the buttons for each form.

Table 1. Window header buttons

Form	Buttons	Description
Entry	Enter	Records your entries.
	Reset	Erases all entries from the Data Entry form.
	Browse	Is displayed when you select a field that has a browse list associated with it. Displays a browse window when selected.
Retrieval	Retrieve	Finds the information that you requested.
	Reset	Erases all entries from the Data Entry form.
	Browse	Is displayed when you select a field that has a browse list associated with it. Displays a browse window when selected.
Update	Enter	Records your entries.
	Reset	Erases all entries from the Data Entry form.
	Cancel	Closes the window without performing any changes.

Examining the Special Menu

Your Data Entry icon might contain one, two, or all three of the forms. Select and hold the `Special` button in the window header to see what other forms are currently available.

When you select and hold this button, a menu is displayed.

The `Special` menu offers the following items:

- Enter Data

  Select `Enter Data` to display an entry form. For more information on entering data, see “Chapter 3. Entering Information,” on page 19.

- Retrieve Data

  Select `Retrieve Data` to display a retrieval form. See “Chapter 4. Retrieving and Deleting Information,” on page 27 for more information.

  You also select `Retrieve Data` to access the update form because you must first retrieve data before updating it. For more information on updating information, see “Chapter 5. Updating Information,” on page 31.

- Generate Log

  Select `Generate Log` to record all entries made in the Data Entry form. See “Recording Your Entries” on page 24 for more information.

- Update Browse Lists

  `Update Browse Lists` is displayed only when a form accesses a browse list. See “Selecting Browse List Constraints” on page 11 for more information on selecting entries from a browse list and updating browse lists.

- Show Data Entry Options

  Select `Show Data Entry Options` to change options for the Data Entry tool. See “Chapter 2. Working Within a Data Entry Form,” on page 9 for more information.

- Create Forms Design Icon and Capture SQL

  The last two menu items, `Create Forms Design Icon` and `Capture SQL`, are reserved for use by the forms designer and are displayed only when the form is used on the desktop on which it was created. These items are described in “Chapter 20. Becoming an Expert Forms Designer,” on page 155.

Chapter 2. Working Within a Data Entry Form

This chapter provides you with general information about how to make simple changes to and move around in the data entry form. It includes instructions for:

- Moving around in a form
- Removing entries
- Browsing an entry field
- Changing how a field resets
- Setting data entry options

Moving Around in a Form

When you open a form, you can randomly select any field. When working within a form, you can move back and forth between fields to change previously entered information.

Moving Between Fields

To enter information into a form, you must be able to move between fields. The field in which you are currently entering information is marked by blinking lines outside the field.

To move between fields:

1. Select inside the desired field.
2. Press the Tab key to advance to the next field.
3. Press Enter to validate the entries and move the cursor to the next open field.

   Consider the following points when pressing Enter to move between fields:

   - If input is required in a field, pressing Enter does not advance you to the next field within a form. Instead, the following message is displayed in the message area.

     ```
     Input is required for this field.
     ```

   - If there is a section field in the form, pressing Enter validates the entire section, which includes all dependent fields, and moves to the next open field.

Moving Within Fields

To make a selection from a field, you must be able to move within the field. To move within a field:

1. Select one of the available choices.
2. Press the spacebar to advance to the next choice. If the far-right choice is highlighted and you press the spacebar, the far-left choice is highlighted.

Removing Entries

You can remove information from a single field or from an entire form.

To remove information from a field:

1. Press Backspace to remove the last letter of an entry in an field.
2. Select the text by clicking in the field three times (or four, if the text surrounds a period) to highlight the entire text, and then press Backspace to delete the contents of the field.

To remove information from a form, click on the `Reset` button in the form window header to remove information from the entire form. This action removes all entries you have made. Field entries in the entry and retrieval forms are removed, with the exception of predefined default entries. In the entry and update forms, the original information is displayed in all fields.

Browsing an Entry Field

When you select a field to enter information, you might be given the option to choose information from a browse list. If a field allows you to choose your entry from a browse list, either the `Browse` button is displayed in the form window header or a browse list is automatically displayed when you select in the field, depending on how the form was created by the forms designer.

If the browse list is not displayed automatically, you can click on the `Browse` button to display a browse list for that field, or you can manually enter the data into the field.

The browse list function associated with the Data Entry tool is similar to the independent Browser tool, but is not related. For more information on the Browser tool, see the *Data Access Tools User's Guide*.

To display a browse list for a field:

1. Select a field in your form. The `Browse` button is displayed in the window header.
2. Click on the `Browse` button.

   A browse list, labeled with the name of the selected field, displays a list of all acceptable entries for that field.

   When a browse list contains several columns, a heading is displayed for each column of items. Within each column, the items are listed in alphabetic or numeric order. Items in one column do not necessarily correspond to those items in the same row in another column. You must select values from the column that relates to the field.

3. Select one or more values from the browse list. See "Selecting Browse List Constraints" for information on constraining the browse list.
4. Click on the `Apply` button in the window header, or close the browse window. Your selection is displayed in the appropriate entry field.

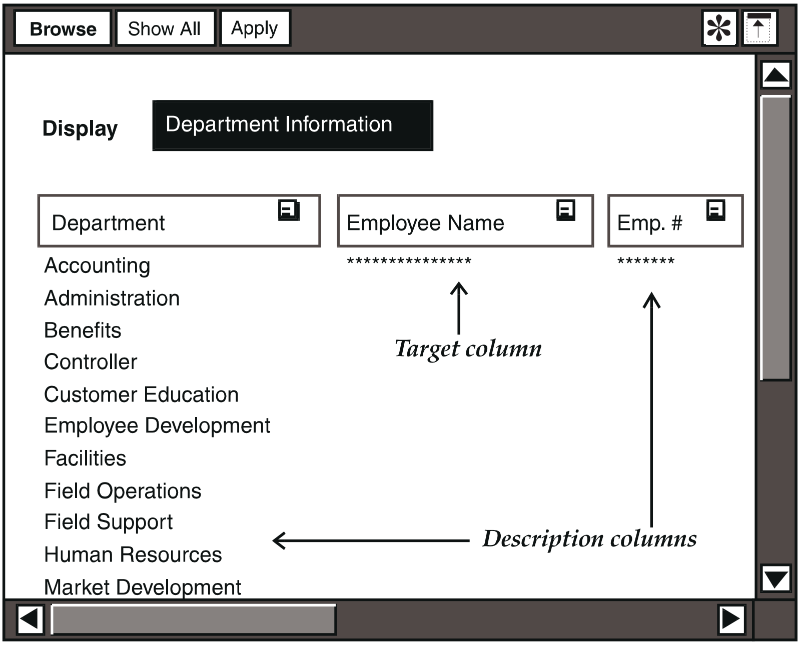
   Closing the browse window enters the selected list item automatically into the current field. Clicking on the `Apply` button adds the item to the field, and keeps the browse window open while you select more items.
5. Enter information into the other fields within the form.
6. Click on the `Enter` button in the window heade to enter the information into the database or capsule application.

Selecting Browse List Constraints

Constraining values with a browse list allows you to further define the type of field entry you want to make. When you select a field that has a browse list, you can *constrain*, or limit, the values.

To constrain browse list values:

1. Click on the `Browse` button in the window header if the browse list is not displayed when you select in the field. The browse window is shown in Figure 2.



Accounting
Administration
Benefits
Controller
Customer Education
Employee Development
Facilities
Field Operations
Field Support
Human Resources
Market Development

Department	Employee Name	Emp. #
Accounting	***************	*******

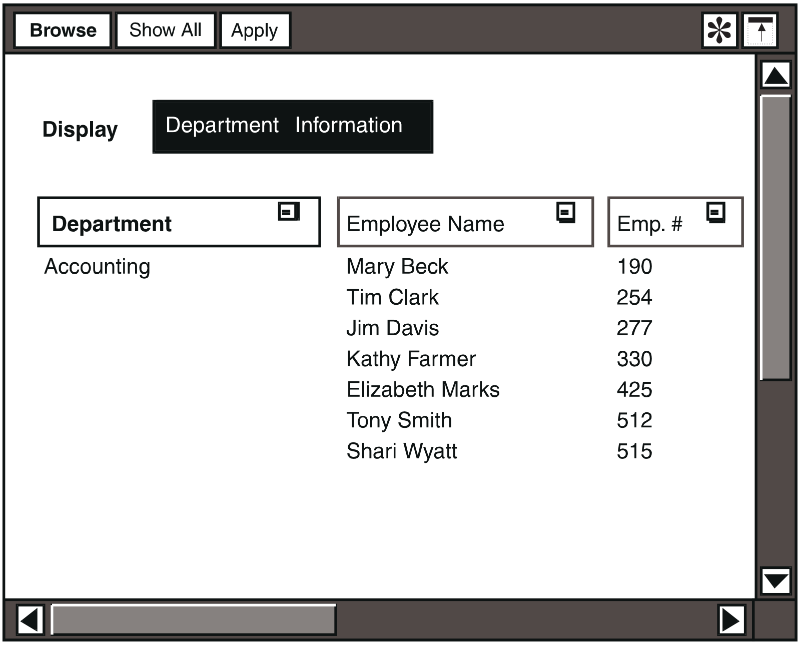
Figure 2. The Browse window

In this example, the Employee Name column is the target column. The target column contains the values that are entered into the field after a selection has been made. The other columns are *description* columns. Description columns are used to limit the values in the target column.

Notice that there are no values in the Emp. # or Employee Name columns. The asterisks (*) mean that there is too much information in these columns to be displayed within the browse window. You must constrain the other columns until you reduce the data enough for these values to be displayed.

2. Select an entry from one of the columns if the browse window contains more than one column. All other choices collapse and disappear from the column, and the item and the heading that you selected are displayed in bold italics. The other browse window columns also collapse to correspond to the selected choice.

   For example, select `Accounting` from the Department column to view values for only the Accounting department. The browse list now contains information about only the Accounting department.

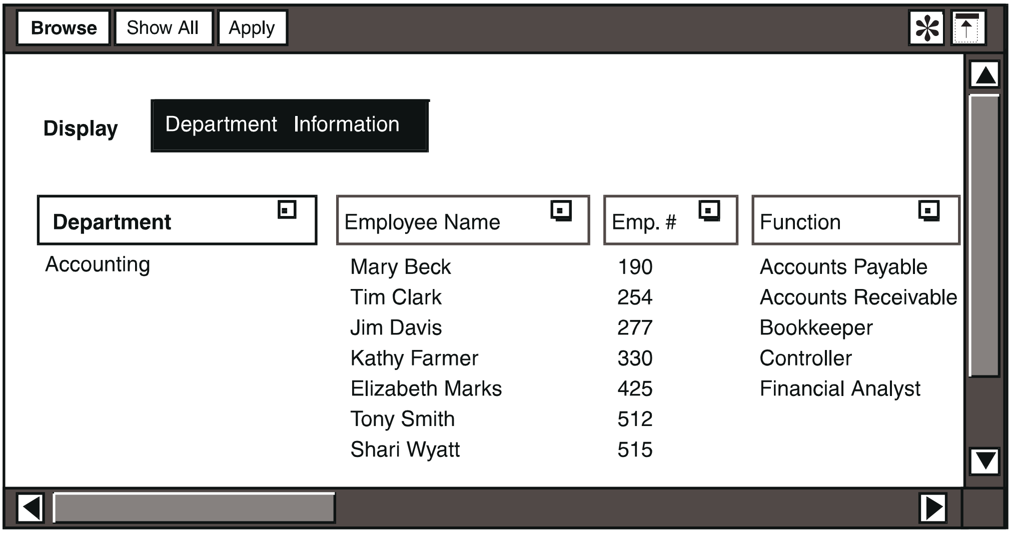


Department Information

Department	Employee Name	Emp. #
Accounting	Mary Beck	190
	Tim Clark	254
	Jim Davis	277
	Kathy Farmer	330
	Elizabeth Marks	425
	Tony Smith	512
	Shari Wyatt	515

Figure 3. Constraining the browse list

Notice that the Emp. # and Employee Name columns now contain values. Suppose the browse window also contains a column that identifies employee functions.



Display: Department Information

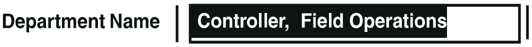
Department	Employee Name	Emp. #	Function
Accounting	Mary Beck	190	Accounts Payable
	Tim Clark	254	Accounts Receivable
	Jim Davis	277	Bookkeeper
	Kathy Farmer	330	Controller
	Elizabeth Marks	425	Financial Analyst
	Tony Smith	512	
	Shari Wyatt	515	

3. Select another entry from one of the remaining columns to further constrain the choices. For example, choose `Financial Analyst` from the Function column. The browse list now shows information only about the financial analysts in the Accounting department.

   If you select and extend on both names and click on the `Apply` button, both names in the Employee Name column are entered into the field. Otherwise, you can constrain the Employee Name column down to one name.

4. Click on the `Apply` button. Your selection is displayed in the field.

   If you already have values in the selected field, your new selection is appended to the original one, separated by a comma and a space. As a rule, the Data Entry tool always reads a comma and a space as the end of an entry, so any information following a comma and a space is considered to be another entry. For example, the following illustration shows two entries separated by a comma and a space: `Controller` and `Field Operations`.



Department Name: Controller, Field Operations

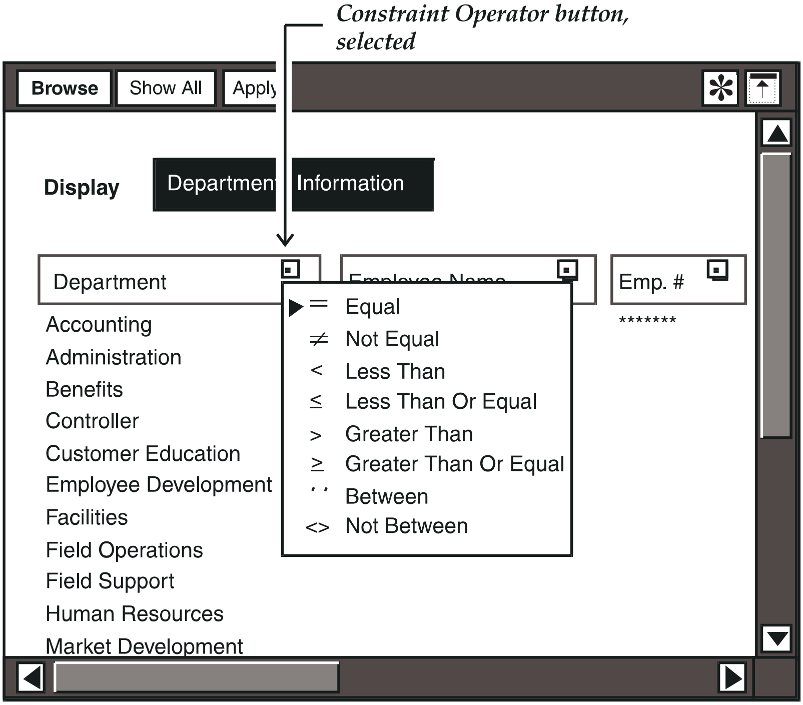
   If you want to separate items that include a comma, such as a date, be sure not to include a space following a comma within the date. For example, in the entry `March 15,1991, March 21,1991`, there is no space before the year, but there is a comma and a space between entries.

If you need to include a comma and a space within an individual entry, enclose the entry within double quotes. For example, `"Lotus, Inc.", Meta5` is read as two separate entries.

Setting Constraints by Selecting an Operator

You can set constraints on a browse list by selecting an operator from the `Constraint Operator` menu. To constrain a column by using an operator:

1. Select an item in a column.
2. Select the `Constraint Operator` button in the column heading. The `Constraint Operator` menu is displayed, as shown in Figure 4.



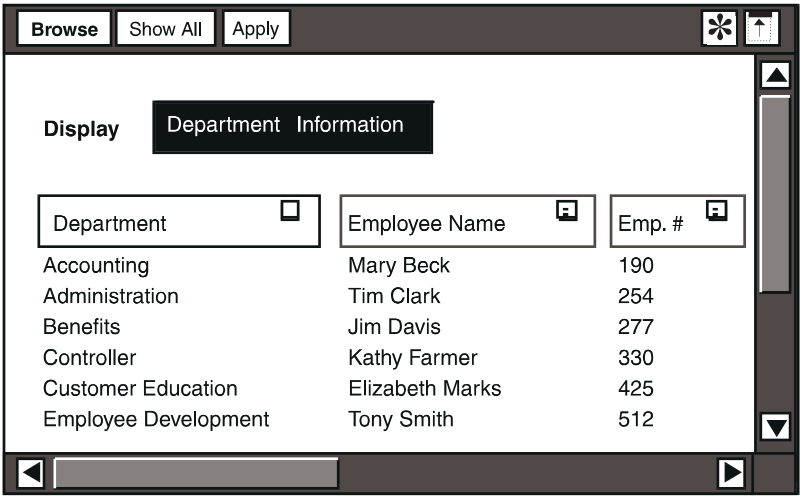
Constraint Operator button, selected

Figure 4. Constraint Operator menu

3. Select an operator in the `Constraint Operator` menu, and the graphic for the `Constraint Operator` button changes (the operator you selected is displayed in the button). The selected column is constrained according to the operator.

For example, suppose you want to view only the departments that are before, or alphabetically less than, F. To display the departments, you select the first department before F, select the `Constraint Operator` button for that column, and then select £ `Less Than Or Equal` in the button menu.

The main browse window now opens as shown in Figure 5. Note the £ sign in the Department column's `Constraint Operator` button.



Display: Department Information

Department	Employee Name	Emp. #
Accounting	Mary Beck	190
Administration	Tim Clark	254
Benefits	Jim Davis	277
Controller	Kathy Farmer	330
Customer Education	Elizabeth Marks	425
Employee Development	Tony Smith	512

Figure 5. A column constrained by an operator

In Figure 5, you see that:

- The Department column displays only the departments alphabetically less than F.
- The Employee Name and Emp. # columns display all items that are alphabetically “less than” the F departments in the Department column.

Selecting an operator gives you a wider range of choices for constraining data. You can further refine your data by constraining the rest of the columns.

Unconstraining a Browse List Selection

If you want to change a constraint to the browse list, you can unconstrain the browse list selection.

To unconstrain a browse list selection, select the bold italicized column item you want to unconstrain. The unconstrained column is then displayed.

Canceling Browse List Constraints

You can choose to not use any of the items you have just constrained. To redisplay all columns, click on the `Show All` button in the browse window header. All columns are displayed, and you can constrain the column by selecting another entry. You can also choose not to make any entries in the form. Close the browse window and delete any entry made in the entry field.

Updating Browse Lists

If information is missing from a browse list, you might need to update the browse list. You can update a browse list from the `Special` menu.

To update a browse list, select `Update Browse Lists` from the `Special` menu.

Changing Field Reset Options

You can set each entry or field of the entry and update forms so they reset in the following ways:

- Automatically

  Whenever you click on the `Enter` button in the window header, or press Enter, to add new information to the database or capsule application, the field automatically resets.

- Manually

  Whenever you click on the `Reset` button in the window header, the form resets.

You cannot change field reset options for a retrieval form. For entry and update forms, you can select the method in which a field is cleared through the Field Options window.

To set the field reset option:

1. Select a field in the entry or update form.
2. Open the Field Options window.
3. Select one of the following choices in the `When Entered` field:
   - `Reset Field`

     This choice clears the field or displays the default value every time you press Enter on your keyboard, or click on the `Enter` button in the form window header.

   - `Retain Field`

     This choice resets the field only when you click on the `Reset` button in the form window header. This feature is useful if you are entering several records in which one field is the same or similar for all records.

Some fields already contain information when you first display the form. These entries are called defaults and were included by the forms designer when that Data Entry tool was designed. You cannot permanently change these entries, and if you try to do so, they are displayed again whenever you click on the `Enter` button or the `Reset` button in the window header of the form.

Setting the Data Entry Options

The Data Entry tool provides options you can set for the Data Entry icon. You can set these options in the Data Entry Options window.

To access the Data Entry Options window, select `Show Data Entry Options` from the form's `Special` menu.

You can set the following options in the Data Entry Options window:

- `Error Signal`

  Choose the type of signal you want to warn you about incorrect entries.

- `Update Browse List`

  Choose a method of updating the browse lists.

Selecting an Error Signal

Your Data Entry icon might have been designed to validate your entries against database tables before changing the database. If the Data Entry tool encounters incorrect information as you work — for example, an invalid field entry — it sends an error signal to notify you about the error. You can specify what type of signal you want to receive.

To set the error signal, select one of the choices in the `Error Signal` field:

- `Flash`

  The window flashes when an error occurs.

- `Beep`

  A beep sounds when an error occurs. You can choose three lengths of sound. When an error occurs, you hear a short, medium, or long beeping noise, depending on the selection you made.

Updating the Browse List

Select a choice in the `Update Browse List` field to control when a browse list is updated. Depending upon how often the database changes, you can specify that the browse list be updated:

- `When Icon Is Opened`

  The browse list is updated every time the Data Entry icon is opened.

- `Upon Request`

  The browse list is updated when `Update Browse Lists` is selected from the Data Entry icon `Special` menu.

- `Upon Validation`

The browse list is updated when it is used for validation.

- `Based on SQL`

  The browse list is updated according to the SQL statements specified in the `SQL for Update Value` field.

When you select `Based on SQL`, an `SQL for Update Value` field is displayed. Enter an SQL statement into this field that gives the Data Entry tool a way to automatically determine when a browse list needs to be updated.

When the database is queried and returns a value, that value is compared against the value that was returned the last time the tool was opened and the SQL statement was issued. If the values are different, the Data Entry tool clears all the values in the browse list. The browse list is then updated when the user clicks on the `Browse` button. You should use any measure appropriate to your browse lists when creating this SQL statement.

If the database changes frequently, then the browse list should be updated frequently. However, updating the browse list is time-consuming, so only update as necessary.

Consult your forms designer or system administrator for more information about this feature of the Data Entry tool.

Chapter 3. Entering Information

There are three basic ways you can enter information into a form. You can enter information manually, select a choice from a browse list, or select one of several predefined choices. The field types are determined by the forms designer, and are discussed in “Chapter 1. Getting Started,” on page 3.

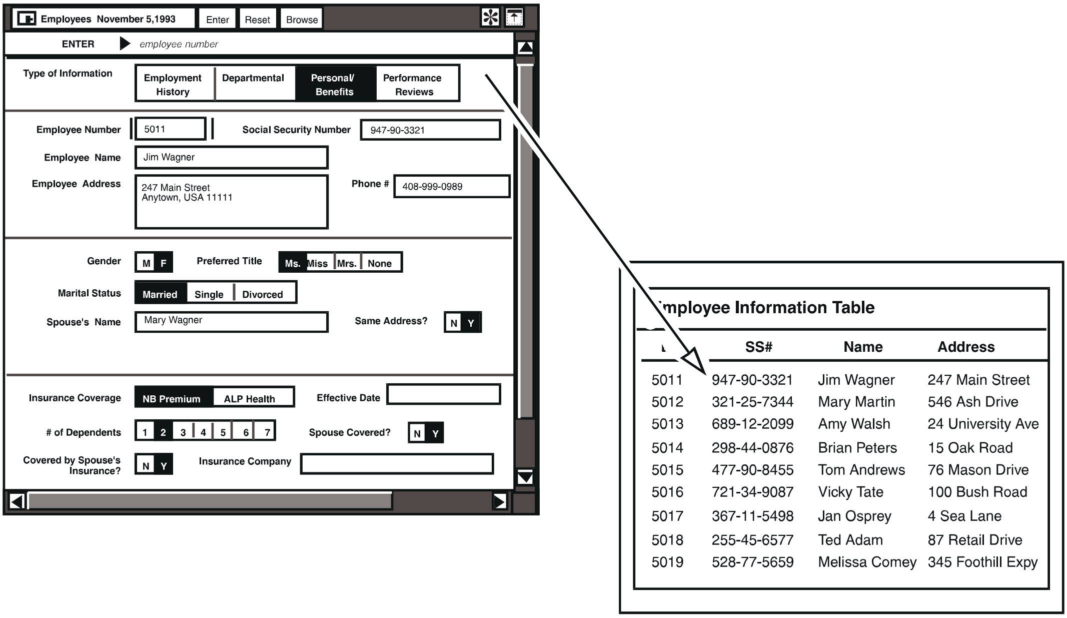
This chapter describes how to:

- Enter information into a database
- Display the entry form
- Enter information
- Record your entries

Entering Information into a Database

Some Data Entry tools are set up to enter information into a database. For example, a Personnel department might use the Data Entry form shown in the previous chapter to update their employee database.

When you enter information into the fields of such a Data Entry form, the Data Entry tool places that information into the appropriate *table* in the database. A database table is a logical structure that contains related information. The relationship between your entries and the structure of the database is shown in Figure 6 on page 20.



Employees November 5,1993

ENTER ▶ *employee number*

Type of Information: Employment History | Departmental | [✓ Personal/ Benefits] | Performance Reviews

Employee Number	5011	Social Security Number	947-90-3321
Employee Name	Jim Wagner		
Employee Address	247 Main Street Anytown, USA 11111	Phone #	408-999-0989
Gender	M [✓ F]	Preferred Title	[✓ Ms.] Miss Mrs. None
Marital Status	[✓ Married] Single Divorced		
Spouse's Name	Mary Wagner	Same Address?	N [✓ Y]
Insurance Coverage	[✓ NB Premium] ALP Health	Effective Date	
# of Dependents	1 [✓ 2] 3 4 5 6 7	Spouse Covered?	N [✓ Y]
Covered by Spouse's Insurance?	N [✓ Y]	Insurance Company	

Employee Information Table

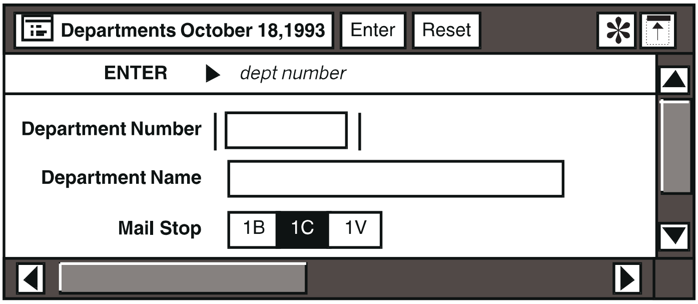
	SS#	Name	Address
5011	947-90-3321	Jim Wagner	247 Main Street
5012	321-25-7344	Mary Martin	546 Ash Drive
5013	689-12-2099	Amy Walsh	24 University Ave
5014	298-44-0876	Brian Peters	15 Oak Road
5015	477-90-8455	Tom Andrews	76 Mason Drive
5016	721-34-9087	Vicky Tate	100 Bush Road
5017	367-11-5498	Jan Osprey	4 Sea Lane
5018	255-45-6577	Ted Adam	87 Retail Drive
5019	528-77-5659	Melissa Comey	345 Foothill Expy

Figure 6. Relationship between a Data Entry form and a table in a database

Figure 6 shows only part of what the Employee Information table in the database would contain if the Data Entry form was constructed as shown. Figure 6 gives you an idea of how information entered into the Data Entry form is structured for storage in the database. In general, each screen of data entered becomes one record in the database table.

Displaying the Entry Form

You use the entry form to enter information into the database or a capsule application. The entry form is usually the first form that is displayed when you select and open a Data Entry icon. If that particular Data Entry icon was not designed with an entry form, the retrieval form is displayed instead. Figure 7 on page 21 shows an example of an entry form.



Departments October 18,1993

ENTER ▶ *dept number*

Department Number

Department Name

Mail Stop 1B [✓ 1C] 1V

Figure 7. Entry form

To redisplay an entry form after retrieving or updating information in the database or capsule application, select `Enter Data` in the `Special` menu of the retrieval form or update form.

Entering Information

Each field type is clearly indicated in the entry form. The form itself prompts you for the information you need to enter, and the message area on your desktop displays warnings that indicate any incorrect procedures.

As an example, if you use an entry form similar to the one in Figure 7, you use two entry fields and a choice field to enter a Department Number, Department Name, and Mail Stop into a Departments table in the database.

To enter information:

1. Open the Data Entry icon. An entry form is displayed.

   Select the first blank field in the form. In Figure 7, the first field is the `Department Number` field, which is already selected. The `Department Number` field is an field with a browse list. In cases like this one, you can use a browse list or manually enter information into the field. Sometimes, the browse list is displayed automatically when you select a field.

2. Click on the `Browse` button to display the browse list for the `Department Number` field.

3. Select one or more items from the browse list. See “Chapter 2. Working Within a Data Entry Form,” on page 9 for more information about using browse lists.

   If the browse list is missing a selection, you need to update the browse list. See “Updating Browse Lists” on page 16 for more information.

4. Close the browse window.

5. Enter and validate the data using one of the following methods:

   - Press Enter on the keyboard.
   - Click on the `Enter` button in the window header.

The data is validated and the pointer automatically moves to the next field that requires data.

6. Make selections or enter data for the rest of the fields in the form.
7. Click on the `Enter` button in the window header to add the information to your database. If you are working in a capsule icon, and @-variables have been used in the Data Entry icon, then some of the new information may be added to the User Input Control window. See “Chapter 6. Using Data Entry in a Capsule Application,” on page 35 for more information on using the Data Entry tool with capsule applications.

   If you receive an error signal, some necessary data is either missing or incorrectly entered. The field pointer returns to the incorrect field.

   If you receive a message in the message area of your desktop stating that you have not entered a valid value, you can use the `Browse` button, when available, to verify the entry. If you do not find your entry in the browse list, update the browse list. See “Updating Browse Lists” on page 16 for more information.
8. Make the necessary changes, and press Enter or click on the `Enter` button in the window header.
9. Click on the `Reset` or `Enter` buttons in the window header before closing the form. You must select one of these buttons any time you make changes or the Data Entry tool will not close the form.

Entering Acceptable Dates

When you have a field that asks you to enter a date, you must enter the date in one of the acceptable formats shown in Table 2 on page 23.

You can also enter `Today` in a date field, and it will enter today's date.

Table 2. Acceptable date formats for direct entry

Date type	Correct formats	Incorrect formats
Day	January 15, 1994	Jan. 15, 1994
	January, 15, 1994	1-15-94
	January.15.94	
	Jan 15, 1994	
	jan 15, 94	
	jan 5 94	
	Jan/15/94	
	15/January/94	
	94.January.15	
	1/15/94	
	15.1.94	
	94-1-15	
	this day	
	today	
	Jan 15, 94 + 2 days	
Week	jan 15, 94 - 2 weeks	
	this week	
	week of Jan 15, 94	
	15 w 94	
	week of 1/15/94	
Quad Week	quad week of Jan 15, 94	

Table 2. Acceptable date formats for direct entry

Date type	Correct formats	Incorrect formats
Month	January, 1994	Jan., 1994
	Jan, 1994	jan, 1994
	january, 94	January
	jan, 94	1-94
	jan 94	
	this month	
	1/94 + 2 months	
Odd BiMonth	JF, 1994	Jan-Feb, 1994
	jf 94	
	1994 JF	
Even BiMonth	JF, 94	Jan-Feb, 94
	jf 1994	
	1994 JF	
Quarter	1q94	First Quarter of 1994
	1Q94	
	this quarter	
Year	1994	'94
	94	
	this year	

Recording Your Entries

You can keep a record of the information you are entering or updating in the database. Keeping a record is helpful if you need to know what changes were made and when they were made. You can record entries for only one session at a time. Each time you open a Data Entry icon, you have to begin a new record. Later entries are not added to previous transcripts. If you are working in a capsule application, any changes made to the User Input Control window are not recorded.

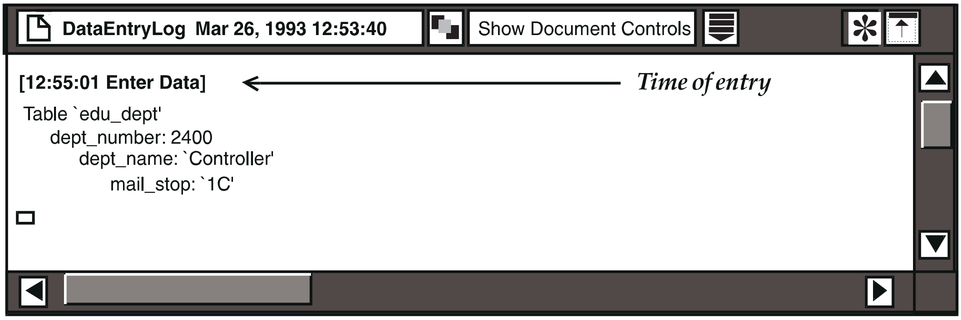
To record entries from an entry form or update form:

1. Select `Generate Log` in the `Special` menu of your form.

The Data Entry Log icon is displayed on your desktop and opens to display the Data Entry Log window.

Both the icon and the window display the date and time at which they were created. The Data Entry Log window records all the information as you enter it into the database as long as the window remains open. The window must remain open on your desktop to continue recording information.

2. Type new data into your form.
3. Click on the `Enter` button, or press Enter. The information is sent to the database, and the entry is displayed in the Data Entry Log window.



```
[12:55:01 Enter Data]
Table `edu_dept'
    dept_number: 2400
        dept_name: `Controller'
            mail_stop: `1C'
```

Time of entry

The Data Entry Log window records each entry in a separate block of text. The time of the entry precedes the text.

4. Close the Data Entry Log window when you have finished entering information into the entry form. The Data Entry Log window closes automatically when you close the form window you are using.

The Data Entry Log icon now contains a record of all entries made for that specific time period. The Data Entry tool creates a new Data Entry Log icon each time you select `Generate Log`. The Data Entry tool does not overwrite existing Data Entry Log icons.

You can save each Data Entry Log icon and use the contents for proofreading your entries. Each log has a unique date and time, reflecting the date and time you created it. However, you may want to rename each icon to better suit your own records.

See “Chapter 20. Becoming an Expert Forms Designer,” on page 155 for more information on capturing the SQL activity generated by the Data Entry tool.

Chapter 4. Retrieving and Deleting Information

In addition to entering data into the database, most Data Entry icons also allow you to retrieve and update information. This chapter explains how to access information currently in the database. It describes how to:

- Display the retrieval form
- Access information in a database
- Retrieve information
- Delete information
- Stop information retrieval

Displaying the Retrieval Form

Use the retrieval form to get specific information from a database and display it on your screen. After you retrieve the information, you can then update or delete it if you have the appropriate privileges. See “Chapter 5. Updating Information,” on page 31 for more information about updating entries in the database.

To display the retrieval form:

1. Open a Data Entry icon.
2. Select `Retrieve Data` in the `Special` menu of the form window header.

Notice that the window header button is labeled `Retrieve` rather than `Enter`, and the prompt is labeled `RETRIEVE` rather than `ENTER`. However, you can still press Enter on the keyboard to retrieve data.

Retrieving Information

You can choose what information to access and retrieve by using the appropriate fields in the retrieval form. You can access and retrieve the following information:

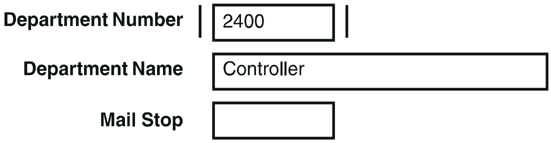
- All data

  To access all data in the database table without any constraints on the data, leave the fields blank. If you have a large database, it can take several minutes to retrieve data that is not constrained. In practice, you usually constrain one or more fields to reduce the amount of data retrieved.

  The Data Entry tool can retrieve up to 200 rows from the database. Any rows exceeding the 200-row limit are not displayed.

- A specific row

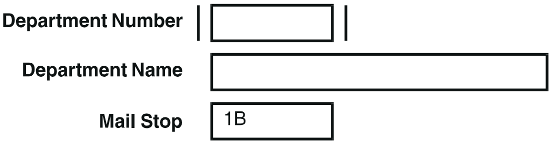
To access a specific row of data in the database, enter field data in all the fields necessary to make the row of data unique. In the next example, the database returns the row of data where the Department Number is equal to 2400, and the Department Name is Controller.



Department Number	2400
Department Name	Controller
Mail Stop	

- Several specific rows

  To access several specific rows of data in the database table, enter field data associated with more than one row. In the next example, the database returns multiple rows of data where the Mail Stop is equal to 1B.



Department Number	
Department Name	
Mail Stop	1B

You can also retrieve data using one of the symbols shown in Table 3 as a prefix for your entry.

Table 3. Retrieval form symbols

Symbol	Returns data that is:
>	greater than the value entered
`<`	less than the value entered
`>=`	greater than or equal to the value entered
`<=`	less than or equal to the value entered
`not=`	not equal to the value entered
`=`	equal to the value entered

You can only enter one value when retrieving data. For example, you cannot retrieve data with the expression `= 5000, 6000`.You can only retrieve data equal to 5000 or 6000. To work around this situation, retrieve data `<= 6000`, and ignore any data retrieved that is less than 5000.

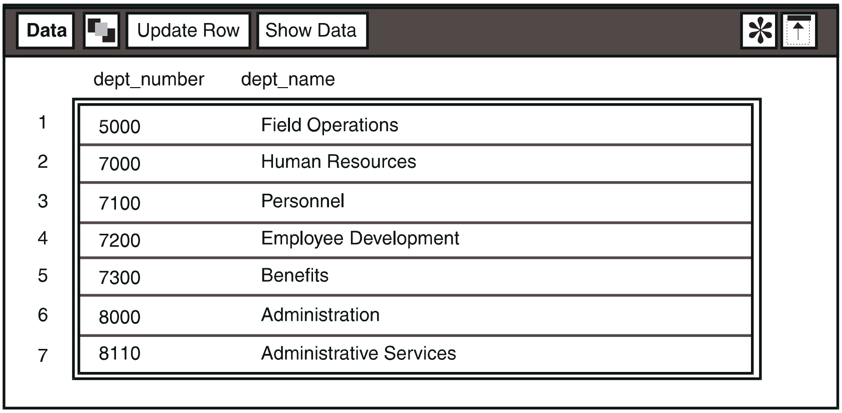
To retrieve information:

1. Select `Retrieve Data` in the `Special` menu of the window header. The retrieval form is displayed.

2. Select a field and enter the data value to retrieve. You do not have to enter data into every field when retrieving information. However, if you only specify data in one field, you could retrieve more data than you need. You can enter data into every field if you want to further constrain the data to be retrieved.

   For example, if you select the `Department Number` field and type `>=5000`, it indicates that you want to constrain the retrieved information to all department numbers that are greater than or equal to the number 5000.

3. Click on the `Retrieve` button in the Data Entry window header. This action causes the Data Entry tool to query the database for information, and the Data window opens displaying a list of all of the information you requested.



	dept_number	dept_name
1	5000	Field Operations
2	7000	Human Resources
3	7100	Personnel
4	7200	Employee Development
5	7300	Benefits
6	8000	Administration
7	8110	Administrative Services

   In this example, you receive a list of all department numbers greater than or equal to 5000, along with their corresponding department names.

   The retrieval data is displayed as `#####` if the numeric precision of the entered data does not match that which was set by the forms designer. In this case, consult the forms designer or system administrator.

   Only data for the fields shown in the retrieval form is displayed in the Data window, which can not be the entire contents of the row. For example, the database might contain a Mail Stop for each item shown in the Data window in the previous illustration. However, the retrieval form shows only the `Department Number` and the `Department Name` fields. Consequently, the Data window lists only that data.

   If multiple rows are returned, then the following prompt is displayed in the message area of your desktop:

   ```
   Select desired row, then delete or update row.
   ```

   If only one row is returned, it is entered automatically into the update form. For a description of how to update information, see “Chapter 5. Updating Information,” on page 31.

4. Make any necessary changes to the information that is displayed, close the Data window, and click on the `Reset` button in the retrieval form window header after you have viewed or updated the information.

Deleting Information

You can only delete information from the database if the forms designer has set up your Data Entry icon to allow you to do so. If you want to delete information, but your Data Entry icon does not permit you to do so, consult your system administrator.

To remove information from a database:

1. In the Data window, select the rows of information you want to remove from the database.
2. Delete the rows. A Confirmation window opens.

   This window acts as a safeguard against accidentally deleting information by giving you another chance to cancel or proceed with the deletion.
3. Select `Proceed` to confirm the deletion. The Confirmation window closes automatically after the data is deleted.

   You can cancel the delete request by clicking on the `Cancel` button in the Confirmation window header, or by closing the Confirmation window before selecting `Proceed`.

If your Data Entry icon does not allow you to delete information from the database, the following message is displayed in the message area of your desktop:

```
You do not have permission to delete rows.
```

Stopping Data Retrieval

You can stop the retrieval process at any time if you decide to retrieve different data, or to be more specific in what you are retrieving.

To stop the retrieval process:

1. Press Stop on your keyboard. The Important Message window opens.
2. The Important Message window contains one of the following messages, depending on when you pressed Stop:

   ```
   Database request was stopped. -- Data can not be complete.
   ```

   ```
   Database request was cancelled.
   ```
3. Close the Important Message window by clicking on the `OK` button.

Chapter 5. Updating Information

This chapter tells you how to update existing information. It describes how to:

- Display the update form
- Change a single row
- Duplicate row changes
- Duplicate field data
- Enter new data into specific fields
- Change all field data

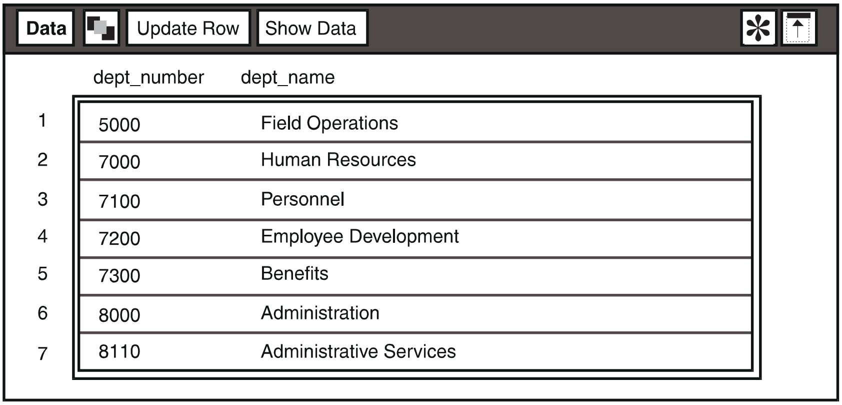
Displaying the Update Form

You access the update form and make changes to the database information by first displaying the retrieval form. Remember, your Data Entry icon must be set up to allow you to make changes. Consult your system administrator if you need assistance. For information about retrieving data, and for an illustration of a retrieval form, see “Chapter 4. Retrieving and Deleting Information,” on page 27.

To display the update form:

1. Open a Data Entry icon.
2. Select `Retrieve Data` in the `Special` menu. The retrieval form is displayed.
3. Select a field, and enter a constraint indicating the data to be retrieved.

   For example, with fields for Department Number and Department Name, select the `Department Number` field and enter `>=5000`. This setting indicates that you want to retrieve information for all department numbers that are greater than or equal to the number 5000.
4. Click on the `Retrieve` button in the window header after entering your constraints. The Data Entry tool queries the database, and the Data window opens as shown in Figure 8 on page 32.



	dept_number	dept_name
1	5000	Field Operations
2	7000	Human Resources
3	7100	Personnel
4	7200	Employee Development
5	7300	Benefits
6	8000	Administration
7	8110	Administrative Services

Figure 8. Data window

The Data window lists all department numbers greater than or equal to 5000 along with their corresponding department names. The following prompt is displayed in the message area of your desktop:

```
Select desired row, then delete or update row.
```

5. Select the row in the Data window that you want to update. For example, suppose you select the first row in Figure 8 on page 32.
6. Click on the `Update Row` button in the Data window header.
7. The retrieval form becomes an update form and each field displays the information you selected in the first row of the Data window.

   You can also click twice on a row to display the data in the update form instead of clicking on the `Update Row` button.

Changing a Single Row of Information

To change a single row of information:

1. Select a row in the Data window. For example, suppose you select the first row of data in Figure 8 on page 32.
2. Click on the `Update Row` button in the Data window header. The update form is displayed and shows the fields that can be updated.
3. Change the field you want to update. For example, change the `Department Number` from `5000` to `5010` in the update form.

   At this point, if you decide not to update the selected row, click on the `Cancel` button in the form window header. When you cancel a change, the update form again becomes a retrieval form. The Data window remains open and you can then select another row if you want to make any further changes.
4. Click on the `Enter` button in the form window header or press Enter on your keyboard to update the database with your changes.

If you entered the data and it is not valid, you receive a warning and a message to correct the invalid data. When the data is valid, the database is updated and the update form becomes a retrieval form.

After you have updated the database, the selected row in the Data window displays an asterisk (`*`) in place of the original data in each column, indicating that the information was changed.

5. Enter a constraint in the retrieval form indicating the data to be retrieved.

   In the current example, you could select the `Department Number` field, then type `>=5000`. This expression ensures that only those department numbers greater than or equal to 5000 are retrieved when you select the `Show Data` button, rather than all department numbers.
6. Click on the Data window header to redisplay the updated information. The new data is displayed with the Department Number as `5010`.
7. Close the Data window.
8. Click on the `Reset` button in the form window header if you did not enter any changes.

   You cannot close the retrieval form window until you have either entered information or reset the fields.

Duplicating Row Changes

You can make the same changes to several rows of database information using a quick duplication method.

To duplicate row changes:

1. Select a row in the Data window.
2. Click on the `Update Row` button in the Data window header. The update form is displayed, showing the fields that can be updated.
3. Select a field in the update form and change the field entry. For example, you could change the department name from Field Operations to Field Operations (HQ).
4. Open the Field Options window.
5. Select Field Options window.

   When you select `Retain Field`, the data that you enter into the update form is retained for duplication, until you click on the `Reset` button in the form window header.
6. Close the Field Options window.

   See “Chapter 2. Working Within a Data Entry Form,” on page 9 for more information about the Field Options window.
7. Click on the `Enter` button in the window header or press Enter on your keyboard to update the database.

The selected row in the Data window changes to asterisks. The update form again becomes a retrieval form.

8. Select another row in the Data window twice, assuming you want to make the same change to this row.

   The update form displays data for the selected row. Each field already contains the updated information and is ready to be entered into the database.

9. Click on the `Enter` button in the window header or press Enter on your keyboard. The information is cleared from the fields in the update form, and the second row in the Data window is changed to asterisks.

   The update form again becomes a retrieval form.

10. Repeat steps 8 and 9 for each additional row in which you want to duplicate the change.

11. Enter a constraint in the retrieval form indicating the data to be retrieved.

    In the previous example, you could select the `Department Number` field, then enter `>=5000`. This expression ensures that only those department numbers greater than or equal to 5000 are retrieved when you select the `Show Data` button, rather than all department numbers.

12. Click on the Data window header after you have made all your changes.

    The updated data is displayed in the Data window.

13. Close the Data window.

14. Click on the `Reset` button in the form window header if you did not enter any changes.

Chapter 6. Using Data Entry in a Capsule Application

You can use a Data Entry icon to set @-variables inside a capsule application. An *@-variable* is a control that passes information to tools within the capsule application. The Data Entry icon must be customized for a specific application to function in this way. The forms designer must customize the icon for specific field data, and prepare to store that data in @-variables.

This chapter describes how to use a Data Entry icon to set and change @-variables inside a capsule application.

See the *Capsule User's Guide* for more information on working with the Capsule tool.

Determining Data Storage

Unless you are the forms designer, it is not immediately apparent to you that an icon has been set to store data in @-variables. You might want to know where data is stored to simplify data entry.

To find out if an icon is set to store data in @-variables, consult with your system administrator or forms designer.

Understanding Data Entry Forms and Capsule Applications

When you enter information into a Data Entry form that interacts with a capsule application, the information is transferred to @-variables within the capsule application.

The contents of the @-variables change based on your entries in the Data Entry form, but the name of the @-variable is constant. Because the name of the @-variable is constant, the contents of the @-variable can be used in other tools within the capsule application.

For example, you want to create a standard report that displays an employee's information. You create the report using a Text tool. You want the information in the report to change based on your entries in the Data Entry form. To do this, place both icons inside a capsule application and define @-variables to use in both tools.

How Information Travels

1. Information from the fields of the Data Entry tool is transferred to the capsule application.

2. The capsule application places the information from the Data Entry tool into the appropriate @-variable inside the User Input Control window.
3. The Text tool reads the information in the @-variables of the User Input Control window and places it in the report.

Using @-Variables for Data Entry

You can set @-variables inside a capsule application by clicking on the `Show Controls` button in the capsule window header to display the User Input Control window. However, the Data Entry icon you are using might have fields already set to specific @-variables, in which case the Data Entry tool automatically enters current input into those User Input Control window.

The following steps explain how to enter values into the @-variables of a capsule application with a Data Entry tool, and view the updated @-variables in the User Input Control window.

If you are using a capsule icon that already contains a Data Entry icon, ignore steps 1 and 2, and begin with step 3. Most of the time you use a customized capsule application, where the Data Entry icon was created inside the capsule icon.

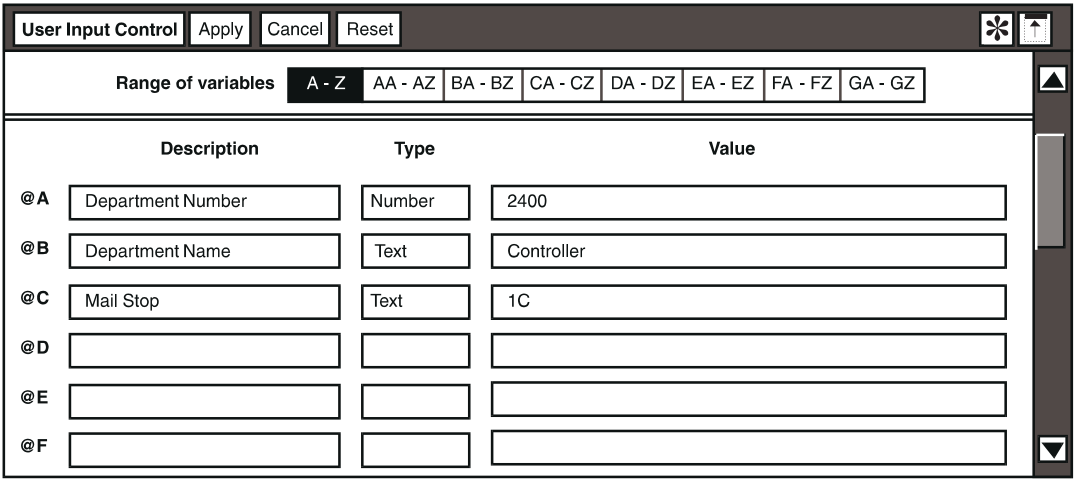
1. Open a capsule icon. The capsule window opens.
2. Select a Data Entry icon that has @-variables defined in it, and move it into the capsule window. The capsule window opens.
3. Open the Data Entry icon.

   The capsule tool has a special feature that automatically opens a Data Entry icon as soon as a capsule icon opens. Name the Data Entry icon `Auto Open`. For more information on this feature, see the *Capsule User's Guide*.
4. Enter the particular information for each field in the form. In this case, enter information into the `Department Number`, `Department Name`, and `Mail Stop` fields.

   When you click on the `Enter` button, the data in the fields is entered into the @-variable fields in the User Input Control window of the capsule application, as shown in Figure 9 on page 37. Other tools within the capsule application then use those values in the @-variable fields for calculations and data retrieval.
5. Enter new data into the current form if you want to run the capsule application using different values. The @-variables are then reset to the new data specifications. In the current example you could enter different department information into the entry form to run the capsule application for another department.
6. Click on the `Show Controls` button in the capsule window header.

   If you want, you can verify in the User Input Control window that the data you entered is reflected in the @-variable fields.

   The User Input Control window opens.



	Description	Type	Value
@A	Department Number	Number	2400
@B	Department Name	Text	Controller
@C	Mail Stop	Text	1C
@D			
@E			
@F			

Figure 9. User Input Control window

The forms designer has already entered information into the `Description` field for each @-variable. The `Value` fields reflect the data you entered into the entry form.

7. Close the User Input Control window. You can now continue working within the capsule application.

   Because the purpose of this Data Entry icon is to set values @-variables in the capsule application, you do not need to reopen the User Input Control window.

   If you are a forms designer, see “Chapter 20. Becoming an Expert Forms Designer,” on page 155 for more information on creating a Data Entry icon to set @-variables inside a capsule application.

Part 2. Using the Forms Design Tool

Have you ever filled out a form, such as an application for employment or a credit card? If that application had been online, it could have been a Data Entry form. Instead of your answers being read by the manager or credit agent, your answers are stored somewhere online, perhaps in a database.

The Forms Design tool is used to create these online forms, or Data Entry forms. These forms are used to:

- Enter data into the database
- Retrieve, delete, and update data in the database
- Set @-variables inside a capsule application

The Forms Design tool and Data Entry tool are separate from one another, but are closely related. Forms designers use the Forms Design tool to create the Data Entry tools that are used by others. This relationship helps to ensure security because forms designers can limit the user's access on a form-by-form basis.

Within Meta5, information can be stored in a database or in @-variables inside a capsule application. But wherever the information resides, you need a method of inserting it and retrieving it later. Data entry forms are the vehicles that allow you to accomplish this information transfer. As the forms designer, you create data entry forms with the Forms Design tool.

Based on your level of understanding, you might be asked to perform the following tasks:

- Design fairly simple forms.
- Design sophisticated forms, using your experience as a database administrator (DBA).
- Design forms to set @-variables in capsule tools, using your experience in application development.

For example, suppose your Human Resources department wants to keep certain information online for the employees at your company. Your system administrator has set up databases to store the information, and you need to create the data entry forms that allow Human Resources to interact with the databases.

As the forms designer, you must know the answers to several important questions before you begin to design a form:

- What type of action needs to be performed?

  You use different data entry forms for entering, retrieving, and updating information, so you must know in advance what action is to be performed. For example, the Human Resources department probably needs all three types of forms so that they can add employees, retrieve their information later, and make changes.

- Does the form use a database or is the form for use with a capsule application?

  You must know whether the form is meant to access a database or interact with a capsule application before designing the form. For example, the forms for the Human Resources department are meant to access a database.

  As the forms designer, you need to understand the structure of the database, and be able to access all database information on which the proposed Data Entry icon operates. You can specify which fields in the database can be accessed, and then design the forms to allow only that access.

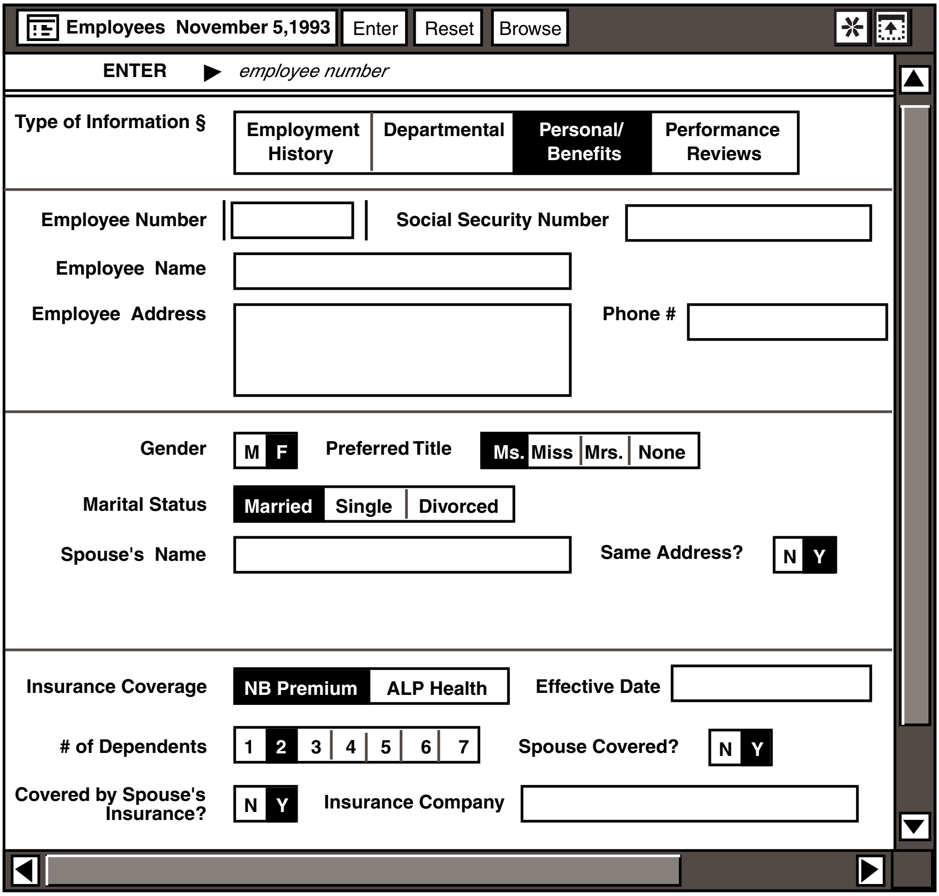
- What restrictions must be in place for the form?

  You need to know whether there is privileged information so that you can set up user access. For example, the forms for the Human Resources department should not be accessed by anyone outside of that department.

- What information needs to be entered into the form?

  You need to know what information will be entered into the form so that you can design your form appropriately. For example, you need to interview the Human Resources department about what information they are planning to put into their databases, and come up with a structure for the data entry forms that makes logical sense to them.

You design forms based on the answers to the previous questions. You can create fairly complex data entry forms by using the various features offered by the Forms Design tool. For example, if you have a form that needs to contain a great deal of information, you can use the features of the Forms Design tool to consolidate information and save space on your form. An example of a complex data entry form is shown in Figure 10 on page 41.



ENTER ▶ *employee number*

Type of Information § Employment History | Departmental | [✓ Personal/ Benefits] | Performance Reviews

Employee Number ▭ Social Security Number ▭

Employee Name ▭

Employee Address ▭ Phone # ▭

Gender M [✓ F] Preferred Title [✓ Ms.] Miss Mrs. None

Marital Status [✓ Married] Single Divorced

Spouse's Name ▭ Same Address? N [✓ Y]

Insurance Coverage [✓ NB Premium] ALP Health Effective Date ▭

# of Dependents 1 [✓ 2] 3 4 5 6 7 Spouse Covered? N [✓ Y]

Covered by Spouse's Insurance? N [✓ Y] Insurance Company ▭

Figure 10. A complex data entry form

Notice how the form consolidates space to show only the pertinent information. For example, in looking at the `Marital Status` field, you notice that because the `Married` choice is selected, the `Spouse's Name` and `Same Address?` fields are displayed. These fields are not necessary for an employee without a spouse, and therefore are not shown. Using the features of the Forms Design tool, you can save space and structure information in many ways.

Chapter 7. Getting Started

Before you design a form, you should understand the basic features of the Forms Design tool. The tasks associated with these basic features include:

- Naming and configuring the Forms Design icon
- Discovering the various windows of the Forms Design tool
- Learning about the main Forms Design window
- Understanding the types of forms you can create

By reading this chapter, you will become familiar with the basic components of the Forms Design tool. You should read this entire chapter before you read the remaining chapters of this guide.

Looking at the Forms Design and Data Entry Icons

The Forms Design and Data Entry icons look very similar. Both show a depiction of a form, but the Forms Design icon also shows a pencil creating a form. Data Entry icons must be created from Forms Design icons, so you will not find a Data Entry icon in the *Blank Icons file drawer.

To understand the Forms Design tool, you must first understand the Data Entry tool because it is the product of the Forms Design tool. Read and work through Part 1, “Using the Data Entry Tool,” on page 1 before using the Forms Design tool.

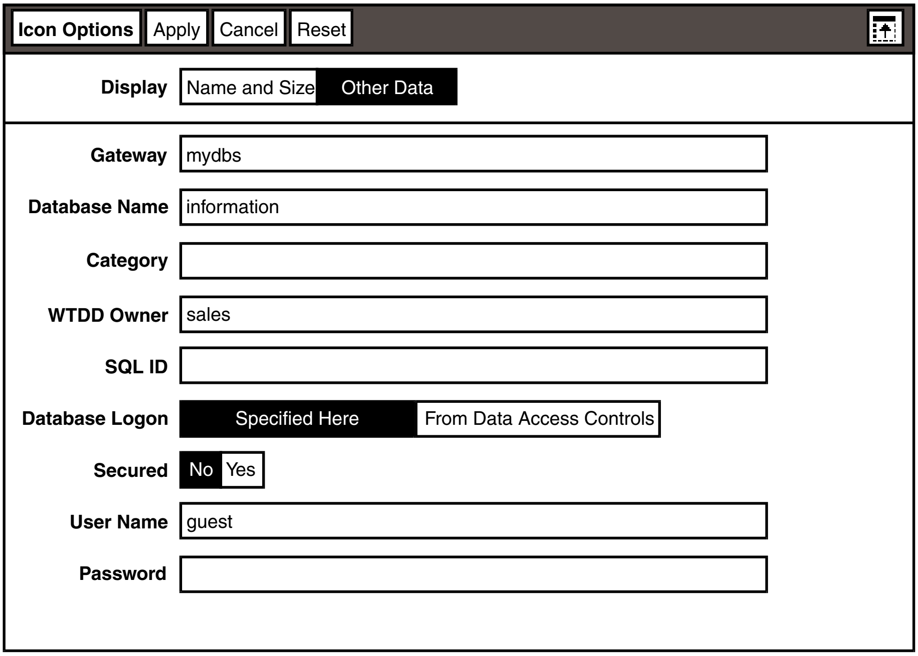
Configuring a Forms Design Icon

After you copy a Forms Design icon from the *Blank Icons file drawer, you must configure it for a specific database before you can use it to access a database. To configure the icon, you must know the password and the name of the database or DB2 subsystem you want to access. If you do not have this information, consult your database administrator.

To configure a Forms Design icon:

1. Select the Forms Design icon and open the Icon Options window.
2. Select `Other Data` in the `Display` field.

   The `Other Data` settings are displayed in the window.



Display	Name and Size [✓ Other Data]
Gateway	mydbs
Database Name	information
Category	
WTDD Owner	sales
SQL ID	
Database Logon	[✓ Specified Here] From Data Access Controls
Secured	[✓ No] Yes
User Name	guest
Password	

Figure 11. Other Data settings in the Icon Options window

3. Using the guidelines provided in Table 4, enter the required information into the text fields and select the appropriate choices in the other fields.
   - When you create a Data Entry icon from a Forms Design icon, the Data Entry icon is configured automatically with the same information as the Forms Design icon. Therefore, it is good practice to configure your Forms Design icon to the name and password of the person who will be using it.
   - When you create a Data Entry icon to be used inside a capsule application that does not access the database, you do not have to configure the icon. See “Chapter 20. Becoming an Expert Forms Designer,” on page 155 for more information on using Data Entry icons with capsule applications.

Table 4. Icon Options window settings

Field	Guidelines
Gateway	Type the name or address of the database gateway that you want the Forms Design and Data Entry icons to access. If the gateway service is in a different realm from the one you are using, immediately follow the gateway name with a colon and then the realm name. For example: `mydbs:HQ`

Table 4. Icon Options window settings

Field	Guidelines

Database Name	Type the name of the physical database you want the Forms Design and Data Entry icons to access. For DB2 databases, type the name of the DB2 subsystem you want to access.
	If you want to connect to a supported remote IBM database using Distributed Relational Database Architecture (DRDA), immediately follow the local database name with a period and then the name of the DRDA server through which the connection is being made. For example:
	`DSN.TD1`
	If you want to connect to a remote Sybase** or Red Brick** database, immediately follow the SQL Server name with a vertical bar and then the name of the database you want to access. For example:
	`SYBSERVER2│SYBPROD`
	If you want to connect to a remote Oracle database, specify the full database name or its alias. For the full database name, specify the standard single-character prefix used for the remote connection (such as **T** for TCP/IP), followed by a colon, the name of the target host followed by another colon, and finally the system ID (SID) of the target Oracle database. For example:
	`T:host1:DBS`

Category	Type either the print name or internal name of the database category needed for this Forms Design icon. The name you specify here is displayed in the window header of the category window when the category is displayed.
WTDD Owner	If the Forms Design icon is configured to a database that supports qualified table names, type the name of the owner of the WTDD tables.
SQL ID	If you are using DB2 Database Manager, type the table qualifier name. The SQL ID overrides the default table qualifier name, which is the name entered in the `User Name` field. If you have questions about the table qualifier name, consult your database administrator.
Database Logon	Select `Specified Here` if you want database access to be based on information you type in the `User Name` and `Password` fields. Select `From Data Access Controls` if you want access to be based on information in the desktop's Data Access Controls window. Consult your database administrator if you have security questions.
Secured	Specify `Yes` if you want the icon's `User Name` field to be set to the user ID of the desktop on which it resides. See “Securing a Forms Design Icon” on page 46 for important information about securing an icon.

Table 4. Icon Options window settings

Field	Guidelines
User ID	Type a user ID the database can identify as valid. This might or might not be the same as your desktop user ID, depending on how your database security is set up. When the Forms Design icon connects to the RDBMS, this is the user ID sent to the RDBMS. If you are using the OS/2 DRDA gateway to connect to a DRDA server, and your DB2 for OS/2 user ID is different from your DRDA server user ID, you must specify both user IDs, separated by a period. For example: `DB2 for OS/2_userid.DRDA_ID`
Password	Type the database user's password. This password might or might not be the same as your desktop password, depending on how your database security is set up. For security, the password is never displayed. If you are using the OS2-DRDA gateway to connect to a DRDA server, and your DB2 for OS/2 password is different from your DRDA server password, you must specify both passwords, separated by a period. For example: `DB2 for OS/2_password.DRDA_password`

Securing a Forms Design Icon

When you secure a Forms Design icon in the Icon Options window, the user ID of the desktop where the icon resides becomes the User Name. When a secured icon is mailed, the User Name changes to the user ID of the recipient. The icon cannot be opened or used by the recipient unless his or her desktop name has the appropriate privileges for the database. You cannot change the User Name field of a secured icon.

If you receive a secured icon from another user and the database does not recognize your user ID, you will not be able to open the icon.

Important:

> Securing an icon is an irreversible operation. Before you secure an icon that you plan to distribute to other users, check with your database administrator to ensure that their desktop names are valid user IDs for the database.

An alternate way to secure a Forms Design icon is to configure it to read the user ID and password setting from the desktop's Data Access Controls window. For more information about the Data Access Controls window, see *Getting Started with the Meta5 Developer's Desktop*, or consult your system administrator.

Learning About the Forms Design Windows

The Forms Design tool operates in several windows. The logical order of the windows in relation to the main Forms Design window is shown in Figure 12 on page 47.

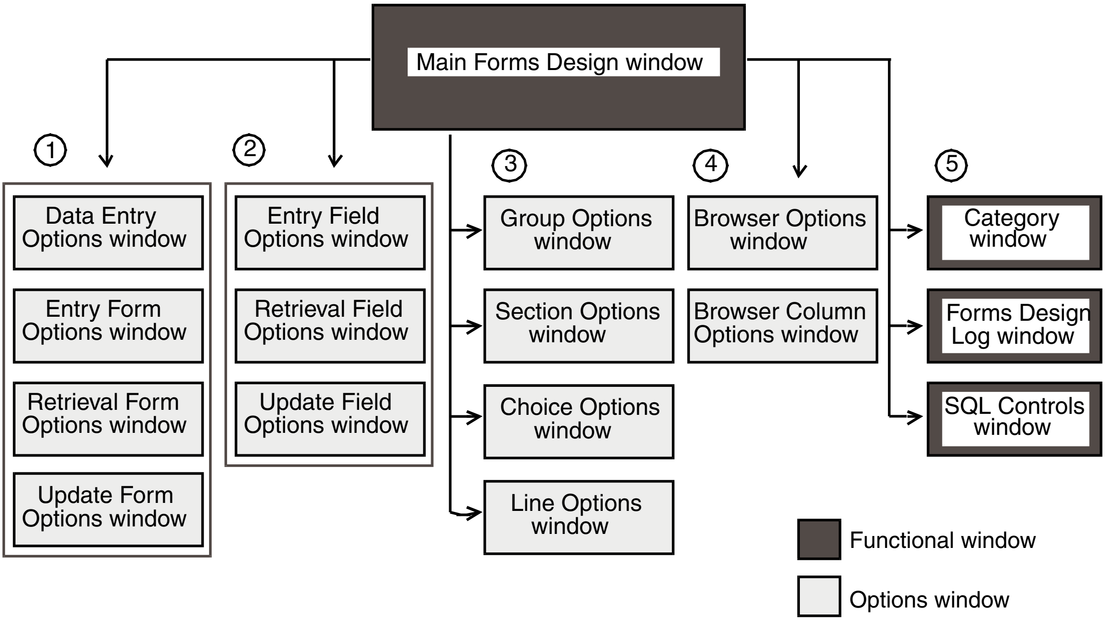


Figure 12. Diagram of the Forms Design windows

Figure 12 shows that there are five branches off the main window:

1. Options windows that apply to the entire data entry form. These options are described in “Chapter 9. Designing the Overall Form,” on page 57.
2. Options windows that apply to specific fields within the form. These options are described in “Chapter 10. Setting Preferences for Fields,” on page 63.
3. Options windows that apply to specific types of fields, such as a group of fields, a section field, a field, or a line field. These types of fields are explained in further detail in “Chapter 11. Entering Data in a Data Entry Form,” on page 67 and “Chapter 19. Enhancing Forms with Design Techniques,” on page 145.
4. Options windows that apply to browse lists. These options are described in “Chapter 18. Working with the Browse Feature,” on page 129.
5. Functional windows that assist with the creation of a data entry form. The category window is described in “Chapter 8. Performing the Basics,” on page 53. The other two windows are explained in “Chapter 22. Improving Security and Performance,” on page 181.

Each window shown in Figure 12 is described in Table 5 on page 48.

Table 5. Forms Design windows

Use this window:	To do this task:
Main Forms Design window	Construct forms and add browse lists.
Category window	View depictions of the tables in the specified database category. The database category is specified in the Icon Options window.
Data Entry Options window	Set options for the Data Entry icon that affect all of its forms.
Entry Form Options, Retrieval Form Options, and Update Form Options windows	Set options for the entry form, retrieval form, and update form that affect all fields in the specified form.
Entry Field Options, Retrieval Field Options, and Update Field Options windows	Set options for individual fields within a form.
Group Options window	Set options for a group of fields within a form.
Browser Options window	Set options for a browse list.
Browser Column Options window	Set options for a customized browse list column.
Section Options window	Set options for sections within a section field.
Choice Options window	Set options for choices within a field.
Line Options window	Set options for lines that divide a form into areas.
Forms Design Log window	View a record of all actions performed while the main Forms Design window is open.
SQL Controls window	Enter SQL statements that are sent to the database.

Understanding the Main Forms Design Window

When you open a Forms Design icon, the main Forms Design window is displayed on your desktop. Enlarge the window until it is displayed as shown in Figure 13.

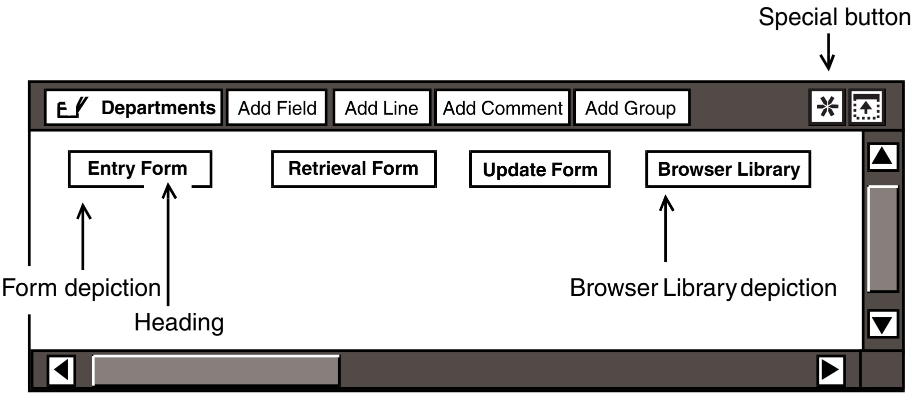


Figure 13. Main window of the Forms Design tool

The main Forms Design window contains three major elements:

- Depictions
- Window-header buttons
- `Special` menu

These elements are discussed in the following sections.

Depictions

The main Forms Design window contains three form depictions, and a browser library depiction. The uses of the four depictions are explained below:

- Entry Form

  Defines the fields into which data can be entered. If you are creating a Data Entry icon to be used only for data entry, you only need the Entry Form depiction.

- Retrieval Form

  Defines the fields that allow retrieved rows to be viewed and deleted. If you are creating a Data Entry icon to be used only for data retrieval, you only need the Retrieval Form depiction.

- Update Form

  Defines the fields that can be updated. If you are creating a Data Entry icon to be used only for retrieving and updating data, and not for entering data, then you use both the Retrieval Form and the Update Form depictions.

- Browser Library

  Defines the browse lists that you specify.

Use all three form depictions if you are creating a form for entering, retrieving, and updating data.

Window Header Buttons

The Forms Design window header contains several buttons that are necessary for constructing forms. The buttons and their uses are described in Table 6.

Table 6. Main Forms Design window header buttons

Button Name	Function
Add Field	Adds a field to a form. Information on defining field options can be found in “Chapter 11. Entering Data in a Data Entry Form,” on page 67.
Add Line	Adds a line to separate fields within a form. This feature is described in “Chapter 19. Enhancing Forms with Design Techniques,” on page 145.
Add Comment	Adds special instructions or comments in a form. This feature is described in “Chapter 19. Enhancing Forms with Design Techniques,” on page 145.
Add Group	Places fields side by side on a form by making them part of a group. This feature is described in “Chapter 19. Enhancing Forms with Design Techniques,” on page 145.

Special Menu Items

When you click on the `Special` button in the Forms Design main window header, the `Special` menu opens.

The `Special` menu contains several items that are useful to the forms designer. Table 7 describes these items.

Table 7. Forms Design Special menu items

Item Name	Function
Show Data Entry Options	Displays the Data Entry Options window from which you can set options for the Data Entry icon that affect all the forms in the icon. See “Chapter 9. Designing the Overall Form,” on page 57 for more information.
Create Data Entry Icon	Creates a Data Entry icon with the name of the Forms Design icon and the date it was created.
Show Category	Displays the category window which contains depictions of the tables in the database category you specified. See “Chapter 8. Performing the Basics,” on page 53 for more information.

Table 7. Forms Design Special menu items

Item Name	Function
Capture SQL	Opens the SQL Controls window which logs the SQL activities. See “Chapter 22. Improving Security and Performance,” on page 181 for more information.

Chapter 8. Performing the Basics

When you have configured a Forms Design icon, you can use it to create the forms that compose the Data Entry icon. This chapter describes the process for creating a simple Data Entry icon. It also covers the following tasks:

- Add fields to a form using the category window
- Copy fields to other forms
- Create a Data Entry icon

The examples in this chapter give you a basic understanding of the Forms Design tool. The remaining chapters of this guide expand on the information in this chapter by explaining the various options that are available to you when you design forms.

Designing an Entry Form

When designing forms, you can copy columns from a table in the category window into the depictions of the forms in the main window. The following example uses a sample database, but the instructions work for any database you decide to use. Keep in mind that the windows shown in this guide will not match exactly those on your screen.

1. Open the Forms Design icon (named *Departments* in this example).
2. Select `Show Category` from the `Special` menu in the main window. The category window opens as shown in Figure 14 on page 54.

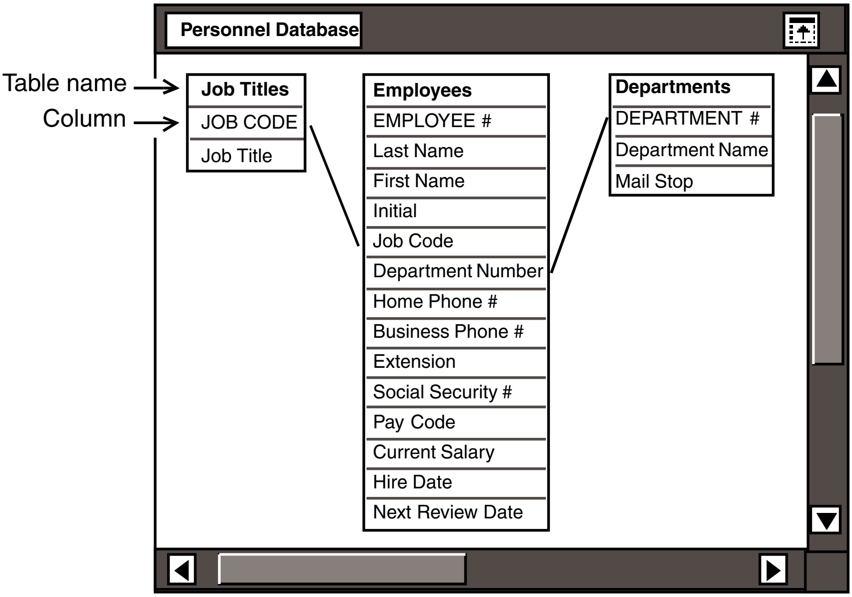


Figure 14. Category window

The category window shows the tables within the database category you specified in the Icon Options window when you configured the icon. In this example, the category window shown contains depictions of the tables in the Personnel Database category.

3. Select one or more columns in the category window.

   For this example, you could select the DEPARTMENT # column in the Departments table, and extend on the Department Name and Mail Stop columns.

4. Copy one or more columns to the heading of the Entry Form depiction in the main Forms Design window.

   The Entry Form depiction now contains the columns that you copied. These columns are displayed under the Entry Form heading in the main Forms Design window. The form depiction expands to accommodate the length of the column names.

5. Close the category window.

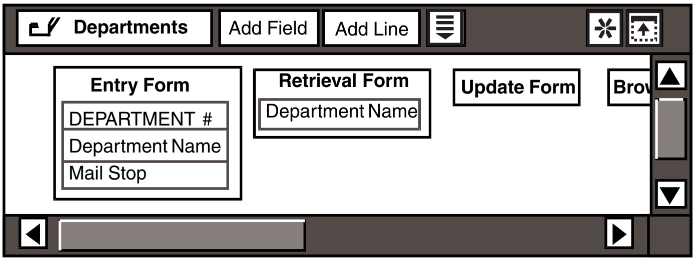
Constructing Retrieval and Update Forms

After you create fields in one form, you can copy and move the fields within the form, or between form depictions in the main window.

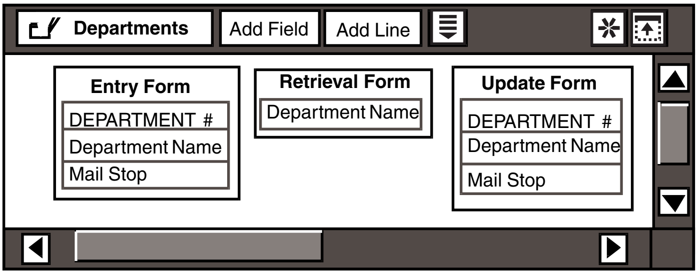
To create retrieval and update forms:

1. Select a field in the Entry Form depiction. In this example, select the `Department Name` field.

2. Copy it to the heading of the Retrieval Form depiction. The field you copied is now displayed in the Retrieval Form depiction.



3. Select another field in the Entry Form depiction. For example, select the `DEPARTMENT #` field, and extend on the `Department Name` and `Mail Stop` fields.
4. Copy these fields to the heading of the Update Form depiction. The fields that you copied are now displayed in the Update Form depiction.



You are now ready to create a Data Entry icon with the forms that you created in the main Forms Design window.

Creating a Data Entry Icon

To create a Data Entry icon:

1. Select `Create Data Entry Icon` in the `Special` menu in the main Forms Design window.

   When you select `Create Data Entry Icon`, the pointer changes, and the following message is displayed in the message area of your desktop:

   ```
   Select location of new Data Entry icon.
   ```
2. Select an empty location on your desktop or in a container, such as a folder.

   A Data Entry icon is now displayed in that position on your desktop. The name of the Data Entry icon is the same as the name of the Forms Design icon that created it, along with the date that it was created. For example, if you create a Data Entry icon from the Forms Design icon used in the previous examples, the name of the Data Entry icon is *Departments*, followed by the date the icon was created.

3. Open the Data Entry icon.

   In this case, the form can now be used to enter, retrieve, and update information in the Departments table.

Learning the Shortcuts

There are several methods of creating data entry forms that can save you time and effort. A few of these methods are described here:

- Know where your data is being stored before you create fields.

  Deciding whether to copy fields from columns in the category window or define them yourself depends on where you plan to store the data from the Data Entry icon. If you are creating a Data Entry icon to store data in the database, copy your fields from columns in the category window. By using this method, you eliminate the need to specify database information because fields copied from the category window already have this information.

  However, if you are creating a Data Entry icon to store data in the User Input Control window of a capsule application, then create fields by clicking on the `Add Field` button in the main Forms Design window header. See “Chapter 17. Storing Data,” on page 125 for more information on storing data in *@-variables*.

- Design the field options before you copy fields between forms.

  In this chapter, you selected fields from the Entry Form depiction and copied them to the Retrieval Form and Update Form depictions without defining any field options. You can save time by first setting the field options for one form, then copying the fields into the other forms that you are creating. By using this method, you do not have to redefine each field as you would if you copy columns repeatedly from the category window. See “Chapter 10. Setting Preferences for Fields,” on page 63 for more information about setting field options.

- Utilize Forms Design's more technical features to reduce form complexity.

  In most cases, you create forms that contain fields from only one database table. However, you can have a form depiction that contains fields from more than one table, or you can specify one or more SQL statements that run after data is inserted or updated in a table. See “Chapter 20. Becoming an Expert Forms Designer,” on page 155 for more information.

Chapter 9. Designing the Overall Form

The first step in designing a Data Entry form is to set options for the Data Entry icon that apply to all the forms in the icon. You can also set options for the individual forms within the icon, such as the entry form, retrieval form, and update form. This chapter describes the options for the:

- Data Entry icon
- Entry form
- Retrieval form
- Update form

The options described in this chapter are *global*, meaning they affect the entire form or icon. You can also use the Forms Design tool to set *local* options; for example, options for specific fields within a form. See “Chapter 10. Setting Preferences for Fields,” on page 63 for information on setting options for specific fields.

Specifying Icon Options

With the Data Entry Options window, you can set several options for the Data Entry icon that affect all of the forms in the icon. The entry form, retrieval form, and update form are all affected by the options set here.

To display the Data Entry icon options, select `Show Data Entry Options` in the Forms Design `Special` menu.

The Data Entry Options window opens.

You can set the following options within the Data Entry Options window:

- Error signal

  Determine the action taken by the system if incorrect information is entered into a field by making a choice in the `Data Entry Error Signal` field. You can specify that either an error message flashes or a beep sounds. If you specify that the error message is an audio signal, you can control its duration.

  If you specify that the error message is an audio signal, a sample beep of the specified duration sounds when you close the Data Entry Options window, or when you click on the `Apply` button.

- Name of the Data Entry icon

  Specify the name that is given to the Data Entry icon by the Forms Design icon within the `Data Entry Icon Name` field. The default name is the name of the Forms Design icon and the date. See “Chapter 20. Becoming an Expert Forms Designer," on page 155 for information about the keywords that you can use to generate the icon's name.

- Precision of real numbers

  Set the default precision used for all real numbers in the Data Entry forms within the `Real Number Precision` field. The default is 10.2, which indicates that the forms can accommodate entries that include 10 numerals with a decimal precision of 2. For example: 98765432.10.

  Be sure that the precision is set correctly for the data that is entered into the forms. If the precision is not set correctly, the user receives `#####` instead of data in a retrieval form.

  The Forms Design tool adheres to precision settings made in the desktop Set Preferences window by default. However, if you set the precision of real numbers with the Data Entry Options window, these settings override those made in Set Preferences window.

Creating Valid Combinations of Forms

Although you can create various combinations of forms for each Data Entry icon, not all combinations of forms are actually valid. For example, if a Data Entry icon contains only an update form, the user cannot update data because a retrieval form is needed to access the update form.

Table 8 shows the valid combinations of forms possible in a Data Entry icon and the types of functions allowed by different forms.

Table 8. Valid combinations of forms

Data entry functions	**Entry form**	**Retrieval form**	**Update form**
Enter data	X		
Enter, retrieve, and delete data	X	X	
Enter, retrieve, delete, and update data	X	X	X
Retrieve, delete, and update data		X	X
Retrieve and delete data		X	

Setting Entry Form Options

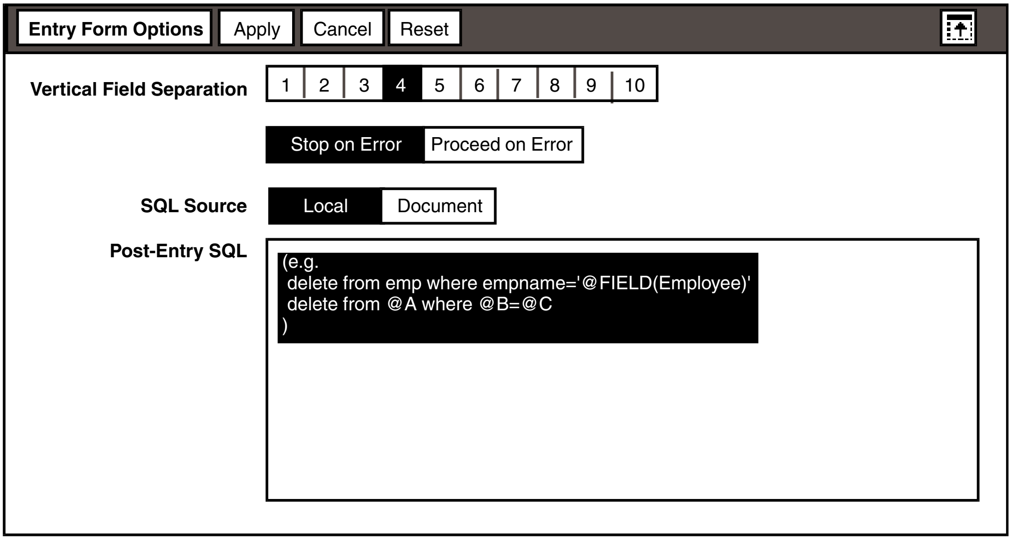
You can set several options that affect both the appearance and operation of the entire entry form.

To display the entry form options:

1. Select the heading of the `Entry Form` depiction in the main Forms Design window.

The depiction becomes highlighted.

2. Open the Entry Form Options window. The default options are shown in Figure 15 on page 59.



Vertical Field Separation 1 2 3 [✓ 4] 5 6 7 8 9 10

[✓ Stop on Error] Proceed on Error

SQL Source [✓ Local] Document

Post-Entry SQL

```
(e.g.
 delete from emp where empname='@FIELD(Employee)'
 delete from @A where @B=@C
)
```

Figure 15. Entry Form Options window

You can set the following options for the entry form:

- Vertical separation between fields

  Set the relative amount of vertical space between fields in the Data Entry window by making a choice within the `Vertical Field Separation` field.

- Behavior on error detection

  Specify whether the Data Entry icon should stop if an error occurs when an SQL statement is run, or continue sending any remaining SQL statements.

  The &ERRNUM statement in post-entry SQL supersedes the `Stop on Error/Proceed on Error` selection. For information about the statements and functions in post-entry SQL, see “Appendix A. Identifying Keywords and Statements,” on page 187.

- Source for SQL statements

  Determine the location of post-entry SQL in the `SQL Source` field. You can use an existing SQL program in another document as the source for post-entry SQL, or enter specific SQL statements into the `Post-Entry SQL` field.

  When you select `Local` as the source, the Entry Form Options window opens, as shown in Figure 15 on page 59. Provide SQL statements to be run each time the `Enter` button is clicked on in the Data Entry window. Enter these statements into the `Post-Entry SQL` field.

When you select Document as the source, a new field is displayed.

Enter the name or path reference to an SQL source document in the Document Name field.

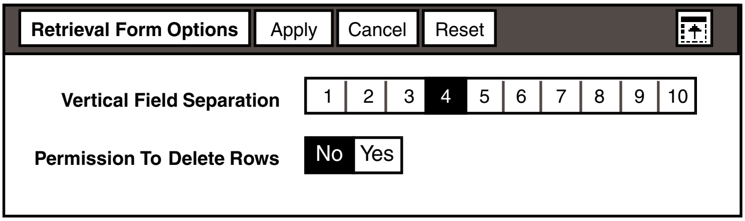
See “Chapter 20. Becoming an Expert Forms Designer,” on page 155 for more information about specifying the SQL Source and post-entry SQL options.

Specifying Retrieval Form Options

You can set several options that affect both the appearance and operation of the entire retrieval form.

To display the retrieval form options:

1. Select the heading of the Retrieval Form depiction in the main Forms Design window. The depiction becomes highlighted.
2. Open the Retrieval Form Options window. The default options are shown in Figure 16.



Vertical Field Separation 1 2 3 [✓ 4] 5 6 7 8 9 10

Permission To Delete Rows [✓ No] Yes

Figure 16. Retrieval Form Options window

You can set the following options for the retrieval form:

- Vertical separation between fields

  Set the relative amount of vertical space between fields in the retrieval form by making a choice within the Vertical Field Separation field.

- Permission to delete rows

  Specify whether or not the user can delete rows from the database by making a choice in the Permission To Delete Rows field. The user name and password entered in the Icon Options window must have appropriate privileges to delete rows from a database.

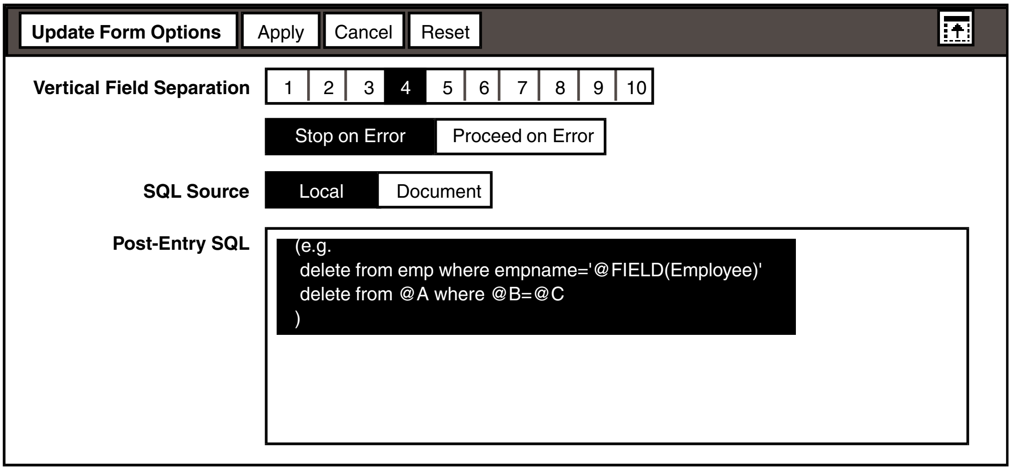
  Use this option with caution because there are limited situations where you want to allow a user this privilege.

Setting Update Form Options

You can set several options that affect both the appearance and operation of the entire update form.

To display the update form options:

1. Select the heading of the Update Form depiction in the main Forms Design window. The depiction becomes highlighted.
2. Open the Update Form Options window. The default options are shown in Figure 17.



Vertical Field Separation: 1 2 3 [✓ 4] 5 6 7 8 9 10

[✓ Stop on Error] Proceed on Error

SQL Source: [✓ Local] Document

Post-Entry SQL:

```
(e.g.
 delete from emp where empname='@FIELD(Employee)'
 delete from @A where @B=@C
)
```

Figure 17. Update Form Options window

You can set the following options for the update form:

- Vertical separation between fields

  Set the relative amount of vertical space between fields in the update form by making a choice within the `Vertical Field Separation` field.

- Behavior on error detection

  Specify whether the Data Entry icon should stop if an error occurs when an SQL statement is run, or continue sending any remaining SQL statements.

  The &ERRNUM statement in post-entry SQL supersedes the `Stop on Error/Proceed on Error` selection. For information about the statements and functions in post-entry SQL, see “Appendix A. Identifying Keywords and Statements,” on page 187.

- `Source for SQL statements`

  Determine the location of post-entry SQL in the `SQL Source` field. You can use an existing SQL program in another document as the source for post-entry SQL, or enter SQL statements into the `Post-Entry SQL` field.

  When you select `Local` as the source, the Update Form Options window opens, as shown in Figure 17 on page 61. Provide SQL statements to be run each time the `Enter` button is clicked on in the Data Entry window. Enter these statements into the `Post-Entry SQL` field.

When you select `Document` as the source, a new field is displayed. Enter the name or path reference to an SQL source document in the `Document Name` field.

See “Chapter 20. Becoming an Expert Forms Designer,” on page 155 for more information about specifying the `SQL Source` and post-entry SQL options.

Chapter 10. Setting Preferences for Fields

You can use the Forms Design tool to design fields by setting local options, or options for specific fields that determine the appearance and functionality of those fields. This chapter provides a brief overview of the types of options that you can set for fields and directions for accessing those options through the field options window.

Although this chapter focuses on the Update Field Options window, other chapters in this book contain more detailed information about using the options described in this chapter.

Accessing Field Options

Each field in a form has unique options associated with it. You can set options to specify the following aspects of a field:

- How the field is displayed in the Data Entry tool
- How data is entered into the field
- Where the data is stored

When you add a field to a form, the Forms Design tool assigns default options to that field. You can change most of these defaults through the field options window for the specific field.

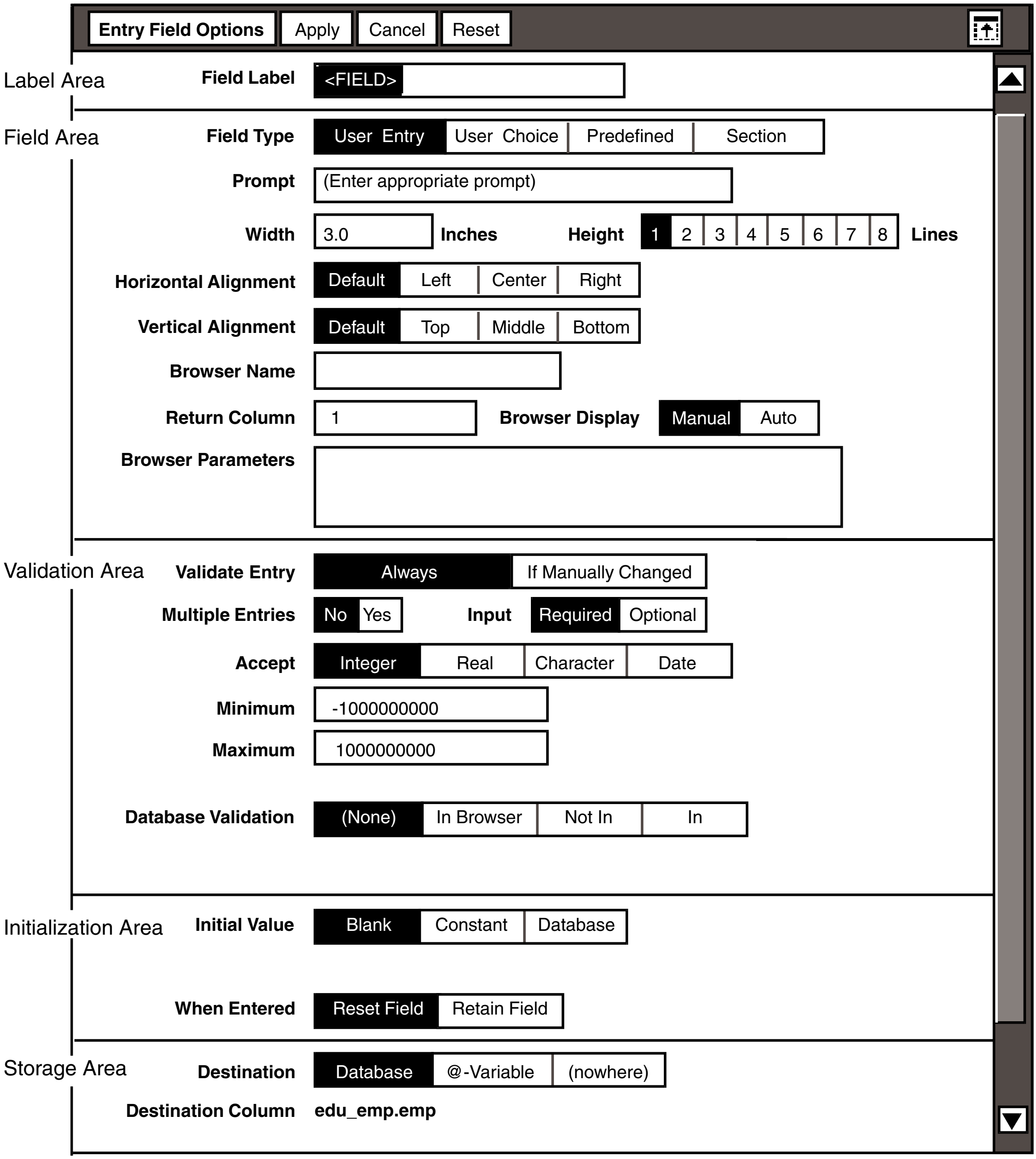
The field options window for a field in the entry form is the Entry Field Options window, and the field options window for a field in the retrieval form is the Retrieval Field Options window.

The field options window differs depending upon the form in which the field resides, the type of data copied from the category window, and the choices selected within the fields of the field options window.

To change field defaults:

1. Select the field for which you want to set options. The field becomes highlighted.
2. Open the field options window. The field options window for a field in the entry form opens, as shown in Figure 18 on page 64.

   Because the form you are creating is unique, the field options window on your desktop might look slightly different than the Entry Field Options window shown in Figure 18 on page 64. Further, Figure 18 on page 64 shows the entire window, but you only see a partial window on your desktop due to the length of the form.



Label Area

Field Label: <FIELD>

Field Area

Field Type: [✓ User Entry] User Choice Predefined Section

Prompt: (Enter appropriate prompt)

Width: 3.0 Inches Height: [✓ 1] 2 3 4 5 6 7 8 Lines

Horizontal Alignment: [✓ Default] Left Center Right

Vertical Alignment: [✓ Default] Top Middle Bottom

Browser Name:

Return Column: 1 Browser Display: [✓ Manual] Auto

Browser Parameters:

Validation Area

Validate Entry: [✓ Always] If Manually Changed

Multiple Entries: [✓ No] Yes Input: [✓ Required] Optional

Accept: [✓ Integer] Real Character Date

Minimum: -1000000000

Maximum: 1000000000

Database Validation: [✓ (None)] In Browser Not In In

Initialization Area

Initial Value: [✓ Blank] Constant Database

When Entered: [✓ Reset Field] Retain Field

Storage Area

Destination: [✓ Database] @-Variable (nowhere)

Destination Column: edu_emp.emp

Figure 18. Entry Field Options window

3. Change the options as needed. “Understanding the Options” on page 65 describes the available options.
4. Close the field options window.

Understanding the Options

The field options window has five areas, as shown in Figure 18 on page 64:

- Label
- Field
- Validation
- Initialization
- Storage

The first two areas, label and field, are always displayed in the field options windows. One or more of the three remaining areas are displayed depending upon the type of field that you select in the `Field Type` field, and the form in which your field resides. However, the general format of the field options windows is the same regardless of which choice you select in the `Field Type` field.

Labeling Fields

When you copy fields from the category window, the Forms Design tool initializes the field label to the print name of the database column to which it corresponds. Change the contents of the `Field Label` field if you want to change the name of the field in the Data Entry window.

For a field within a retrieval form, the field area has an additional field, `Column To Compare`, that contains the name of the database column that corresponds to your field. The Data Entry tool compares the entry in the `Field Label` field to the database column shown in the `Column To Compare` field.

The `Field Label` field is also where you enter a global name. To do this, type `GLOBAL:` followed by the global name that you are assigning to the field. See “Chapter 20. Becoming an Expert Forms Designer,” on page 155 for information on @-keywords and the global function.

Designing Fields

Decide in the field area how data is entered into the field, and how the field is displayed in the Data Entry window.

For a description of the different types of fields available to you, and the options for a predefined field, see “Chapter 11. Entering Data in a Data Entry Form,” on page 67. For information about field options for the other field types, see “Chapter 12. Constructing Entry Fields,” on page 73, “Chapter 13. Designing Choice Fields,” on page 81, and “Chapter 14. Building Flexibility with the Section Field,” on page 95.

Choosing Validation Methods

The validation area is displayed only for fields. This area contains information about how the Data Entry tool validates data entered into a field. It also contains editing information if the field accepts character strings.

See “Chapter 15. Validating Data,” on page 103 for a detailed description of validating field data.

Initializing Fields

Specify how the Data Entry tool *initializes*, or gives value to a field when the Data Entry window opens, by setting options in the initialization area. The initial value can be acquired from the database or by using an @-variable in the User Input Control window of a capsule application, or you can specify a constant in the initialization area.

See “Chapter 16. Initializing Fields,” on page 117 for a detailed description of the various initialization options.

Storing Information

Determine where the Data Entry tool stores the data from this field within the storage area. See “Chapter 17. Storing Data,” on page 125 for a detailed description of this area.

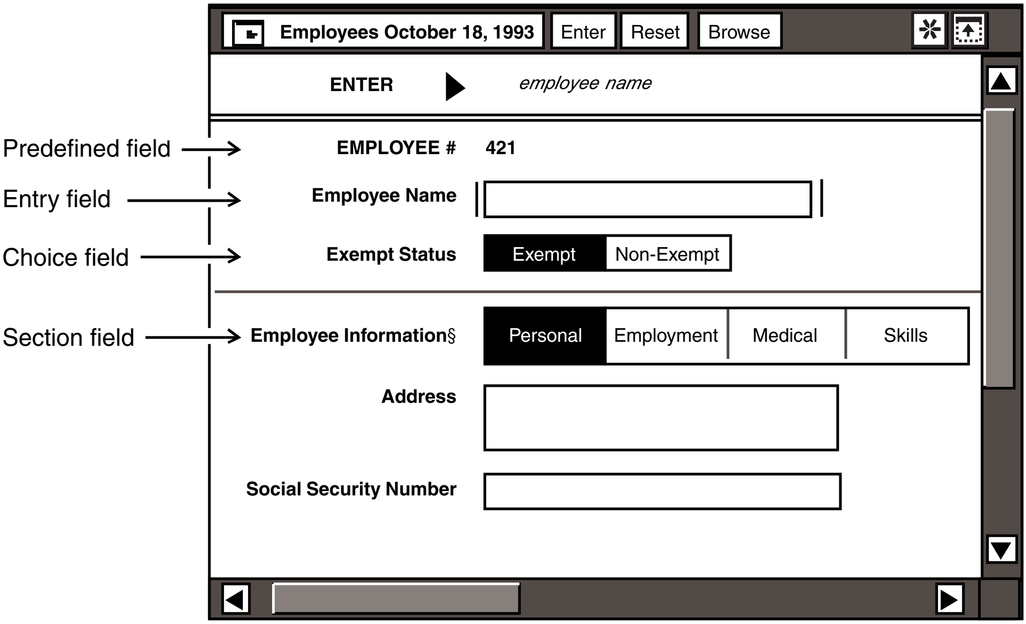
Chapter 11. Entering Data in a Data Entry Form

You can specify several ways that data can be entered into a field in a Data Entry form. This chapter describes the ways that you can define a field, and explains the differences between the methods. The options discussed in this chapter are displayed in the field area of all three field options windows. Other chapters in this book explain the options for the various data entry methods in greater detail.

See “Chapter 10. Setting Preferences for Fields,” on page 63 for an overview of the areas in the field options windows.

Defining Field Types

Data can be entered into a field using four different field types. Figure 19 on page 67 shows how each field type is displayed in an entry form.



Employees October 18, 1993

ENTER ▶ *employee name*

EMPLOYEE # 421

Employee Name

Exempt Status [✓ Exempt] Non-Exempt

Employee Information§ [✓ Personal] Employment Medical Skills

Address

Social Security Number

Figure 19. Predefined, entry, choice, and section fields

A field could be defined as any of the following:

- Predefined field

  The Data Entry tool supplies a value that cannot be changed.
- Entry field

The user enters the data directly into the field, or selects a value from a browse list. The `User Entry` choice is the default of the `Field Type` field.

- Choice field

  The user selects a choice from the field. You can either define the choices yourself, or generate the choices dynamically from the database. Choice fields can also have *dependent fields*, or fields that are displayed only when a certain choice is selected. Choices made from dependent fields of a field must be applied while still viewing that dependent field.

- Section field

  The user selects a choice from the field that you have created. Each choice displays a section that contains dependent fields. When the user enters the information requested in each section and clicks on the `Enter` button, all sections are validated, including all dependent fields. A symbol, §, distinguishes a section field from a field.

Table 9 describes the criteria used to determine the field types for the Data Entry icon shown in Figure 19 on page 67. Use this table in conjunction with Figure 19 on page 67 to help you understand how to choose an appropriate field type.

Table 9. Explanations of field type decisions

Example Field	Field Type	Reason this field type was used:
EMPLOYEE #	Predefined	The user should not be able to change the employee number. The employee number can be determined from the name entered in the `Employee Name` field.
Employee Name	Entry	The user either knows the name to enter, or can find it in a browse list. The data entry form pertains to more than one employee, so the field cannot be predefined. There are too many employees to create a field with all of their names. The choice of names alters due to changes in employment, so a field is not feasible.
Exempt Status	Choice	There are few choices for the field. The choices will not change. The descriptive names associated with each choice are short enough to display in a field. Only one choice needs to be applied at once.

Table 9. Explanations of field type decisions

Example Field	Field Type	Reason this field type was used:
Employee Information	Section	The information to be obtained can be organized into distinct sections which all have dependent fields. There is too much information for a field to be useful. The categories within the section will not change. All choices and their dependent fields are active and can be validated when the `Enter` button is clicked on. Therefore, all of the information for an employee can be entered at once without having to apply choices for each dependent field.
Address	Entry	The address cannot be predefined for new employees.
Social Security Number	Entry	The social security number cannot be predefined for new employees.

Displaying the Field Options Windows

The fields shown in the entry form in Figure 19 on page 67 were created in the `Field Type` field in the Entry Field Options window. The following steps explain how to access the Entry Field Options window. To open the Retrieval Field Options window or the Update Field Options window, select a field under the appropriate form depiction and follow these steps.

To display the Entry Field Options window:

1. Select a field in the `Entry Form` depiction of the main Forms Design window.
2. Open the Entry Field Options window.

   The fields that are displayed in an Entry Field Options window are determined by the type of data copied from the category window and the choices selected within the fields of the Entry Field Options window. Because the form you are creating is unique, the Entry Field Options window on your desktop might look slightly different than the windows shown in this guide.

   When you first copy a field from the category window into a form in the main Forms Design window, the Forms Design tool defines that field as an entry field. As the forms designer, you can change the field type after you have copied the field into a form.
3. Set the options to define your field.
4. Close the Entry Field Options window to apply your settings.

   The options in the field area change depending upon the selection you make in the `Field Type` field. The `Field Type` field and the `Prompt` field remain consistent for all field types but the predefined field type. The `Prompt` field is discussed in “Entering a Prompt Message”. Except for the fields relating to the predefined field type, which are discussed in this chapter, all other fields relating to the field type chosen are explained in subsequent chapters.

Entering a Prompt Message

When you choose to create an entry, choice or section field, you can enter a prompt into the `Prompt` field. Enter information in the `Prompt` field that gives instructions applicable to the form being used.

When you first open the field options window for a field that was copied from the category window, a message already exists in the `Prompt` field. The WTDD (Workstation Tools Data Dictionary) provides an initial prompt based on the description field of the database column copied from the category window.

For example, suppose you copy a `DEPARTMENT #` field from the category window into the main Forms Design window. When you display the field options window for the `DEPARTMENT #` field, the prompt message could be `dept number`, as shown in the following illustration.

Prompt dept number

You can change the prompt message to be more specific, and thus more easily understood. For example, you can change the `dept number` prompt message to `4-digit department number`.

When the `DEPARTMENT #` field is selected, the new prompt message is displayed:

```
ENTER   4-digit department number
```

You can open the field options window and change the prompt message at any time. After each change you must create a new Data Entry icon with the updated prompt.

Creating Predefined Fields

The Data Entry icon supplies values for predefined fields. The value in a predefined field cannot be changed by the user. Examples of predefined fields are those that contain the current date or the name of the user.

To create a predefined field:

1. Copy a field from the category window to the form depiction in the main Forms Design window.
2. Open the field options window.
3. Select `Predefined` in the `Field Type` field.
4. Close the field options window.

When you define a field as predefined, you must specify the predefined value in the initialization area of the field options window.

Using predefined fields with `Blank` selected in the `Initial Value` field is useful for adding blank lines to a form. Blank space can enhance the appearance of a form because it makes fields stand out more clearly.

Using predefined fields with `Constant` selected in the `Initial Value` field is useful for adding comments or instructions to a form.

For more information about the initialization area of the field options window, see “Chapter 16. Initializing Fields,” on page 117.

Chapter 12. Constructing Entry Fields

You can set many options for an entry field that determine its appearance and execution. The options discussed in this chapter are displayed in the field area of the field options windows when you select the `User Entry` choice in the `Field Type` field. This chapter describes:

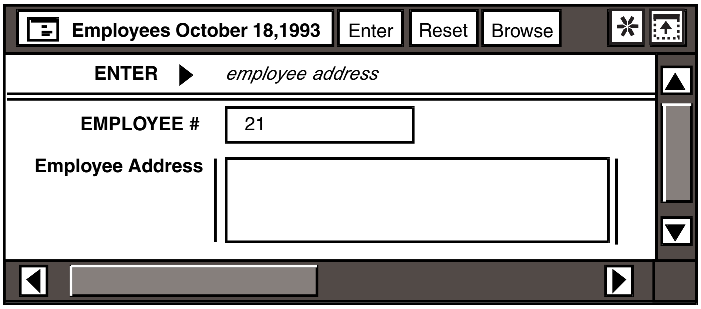
- Setting width and height measurements
- Aligning the field horizontally and vertically
- Associating a browse list
- Specifying the return column for the browse list
- Displaying the browse list
- Setting parameters for the browse list

See “Chapter 11. Entering Data in a Data Entry Form,” on page 67 for an overview of the different field types available to you. For information about options you can set in the other sections of the field options windows, see “Chapter 15. Validating Data,” on page 103, “Chapter 16. Initializing Fields,” on page 117, and “Chapter 17. Storing Data,” on page 125.

Creating Entry Fields

You can specify the type of field that you want to create within a form in the field options windows. Define an entry field as a field when you want the user to either enter a value into a field, or select a value from a browse list that contains valid values.

Examples of entry fields are name and address fields where the user enters a value or selects a value from a browse list. For example, you might want to create an entry form similar to that shown in Figure 20. In this example, the `Employee #` and `Employee Address` fields are entry fields.



ENTER ▶ *employee address*

EMPLOYEE # 21

Employee Address

Figure 20. Entry fields in an entry form

You can create entry fields for the three form types (entry, retrieval, and update) by following the steps given here. Substitute the correct field options window in the instructions.

To create an entry field for an entry form:

1. Select a field in the Entry Form depiction in the main Forms Design window.
2. Open the Entry Field Options window. This window looks similar to the Entry Field Options window shown in Figure 18 on page 64.
3. Select `User Entry` in the `Field Type` field.
4. Select a choice for each field and click on the `Apply` button.

   If you want an entry field to have a browse list, or if you want the entry field data to be validated against the database, you must also set certain options in the validation area of the field options windows. See “Chapter 15. Validating Data,” on page 103 for more information.
5. Close the Entry Field Options window when complete.

All of the options in the field area of the Entry Field Options window apply to entry, retrieval, and update forms.

Table 10. Entry field options

Field	Choices	Function
Field Type	User Entry	Displays the options for a field.
Prompt	N/A	Prompts the user to perform an action, such as enter a value. The `Prompt` field can be used to clarify the format of the entry. The Forms Design tool initializes this field to the WTDD description of the column copied from the category window. For more information on redefining the prompt, see “Chapter 11. Entering Data in a Data Entry Form,” on page 67.
Width	N/A	Sets the field width, in inches or centimeters, where data is entered.

Table 10. Entry field options

Field	Choices	Function
Height (Lines)	1 to 8	Determines the number of lines of text allowed in the field.
Horizontal Alignment	Default	Aligns the field according to the system default.
	Left	Aligns the field flush left in its space.
	Center	Centers the field horizontally in its space.
	Right	Aligns the field flush right in its space.
Vertical Alignment	Default	Aligns the field according to the system default.
	Top	Aligns the field at the top of its space.
	Middle	Centers the field vertically in its space.
	Bottom	Aligns the field at the bottom of its space.
Browser Name	N/A	Specifies the browse list to be associated with the field.
Return Column	N/A	Specifies the column in the browse list whose values are to be returned to the field.
Browser Display	Manual	Displays the browse list when the `Browse` button is clicked on.
	Auto	Displays the browse list automatically when the user selects in the field.
Browser Parameters	N/A	Specifies terms that can be used as constraints for the browse list.

Setting the Dimensions

Specify the dimensions of an entry field by entering information in the `Width` field and selecting a choice in the `Height (Lines)` field.

The characteristics of the `Width` and `Height (Lines)` fields are:

- The width of an entry field is measured in inches or centimeters, based upon selections in the Set Preferences window. See *Getting Started with the Meta5 Developer's Desktop* for more information.
- The height of an entry field is measured relatively by lines of text.

To determine the dimension of an entry field, enter the appropriate number of inches or centimeters in the `Width` field, and select the number of lines from the `Height (Lines)` field.

Aligning Horizontally and Vertically

You can determine the horizontal and vertical alignment of a field in a form by selecting choices in the `Horizontal Alignment` field and `Vertical Alignment` field.

To set the horizontal alignment of your field, place it either to the left, right, or center of its space, or choose the default setting. The horizontal space a field can occupy is determined by the longest field in the group or form. A *group* is a collection of fields that are bound together. For example, if you want to align several fields horizontally, you would put them in a group. See “Grouping Fields” on page 145 for more information about creating groups.

Figure 21 shows the horizontal alignments available for a field, relative to the space in which it resides.

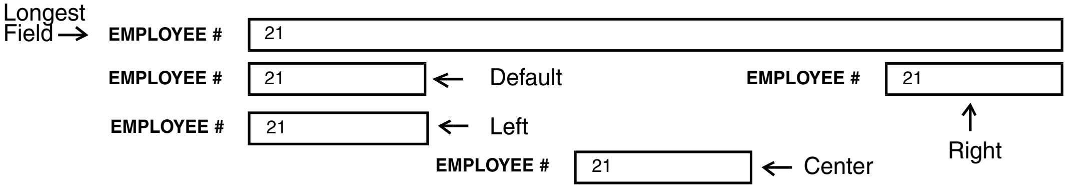


Figure 21. Horizontal alignment within a given space

You must place fields in a group to vertically align them. To set the vertical alignment of your field, place it at the top, middle, or bottom of its space, or choose the default setting. The vertical space a field can occupy is determined by the field with the greatest height in the group. To make this feature work within a group, you must specify that the field size is constant. See “Setting the Field Size” on page 149 for more information about setting the size of fields within a group.

Figure 22 on page 77 shows the vertical alignments available for a field, relative to the space in which it resides. The dotted lines represent the space each field can occupy.

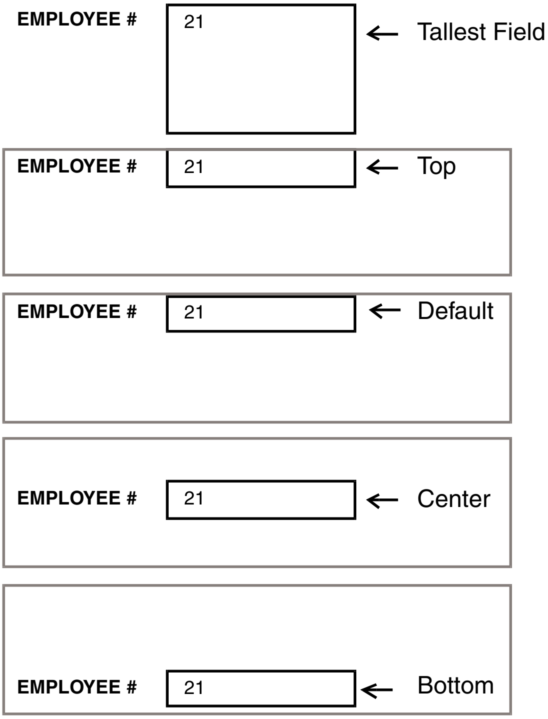
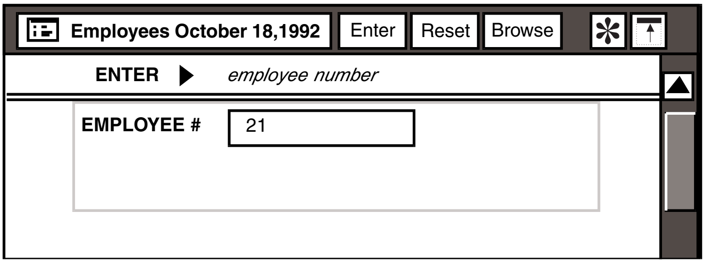


Figure 22. Vertical alignment within a given space

When you specify the alignment of a field, however, you can specify the horizontal and the vertical alignment together. For example, if you set the horizontal alignment to left, and the vertical alignment to top, the field aligns as shown in Figure 23 on page 77. The dotted lines represent the space determined by the longest field and the tallest field. The dotted lines are not displayed on the screen.



ENTER ▶ *employee number*

EMPLOYEE # 21

Figure 23. Field alignment within a given space

Associating an Entry Field with a Browse List

Enter the name of the browse list that you want to be associated with your entry field. To find the available browse lists, look under the Browser Library heading in the main Forms Design window. Figure 24 on page 78 shows a sample Browser Library listing.

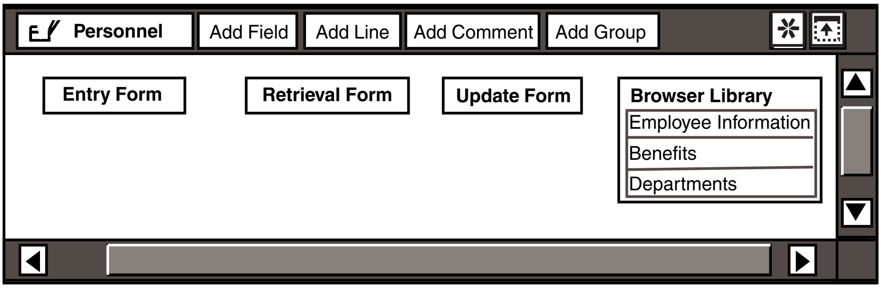


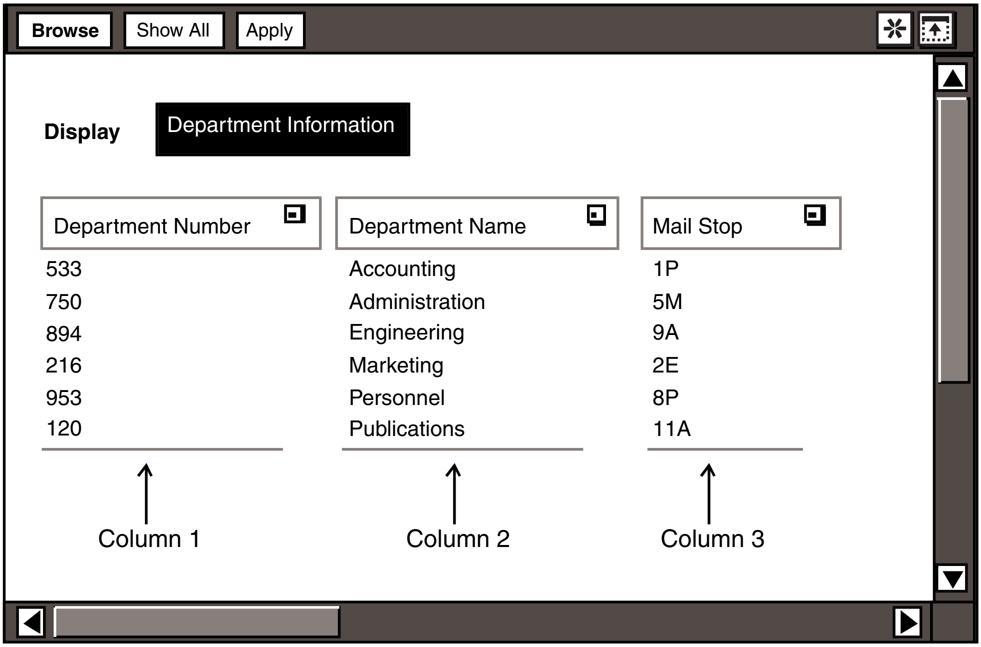
Figure 24. Sample Browser Library listing

To establish a browse list for the `EMPLOYEE #` field shown in Figure 23 on page 77, choose the “Employee Information” browse list. Enter the name of this browse list into the `Browser Name` field.

As a forms designer, you can specify the names and properties of the browse lists. For information about setting up browse lists, see “Chapter 18. Working with the Browse Feature,” on page 129.

Determining the Return Column

Within a browse list, there are usually several columns. For example, in a browse list for department information, there could be columns such as department number, department name, and mail stop. The browse window for such a browse list is shown in Figure 25.



Display Department Information

Department Number	Department Name	Mail Stop
533	Accounting	1P
750	Administration	5M
894	Engineering	9A
216	Marketing	2E
953	Personnel	8P
120	Publications	11A

Figure 25. Browse window for a browse list

The columns in a browse list are numbered consecutively from left to right, beginning with 1. In Figure 25, column 1 is Department Number, column 2 is Department Name, and column 3 is Mail Stop.

You specify which column's contents should be returned to your field by entering the appropriate column number in the `Return Column` field.

For example, if you enter `2` in the `Return Column` field, the browse list returns the contents from column 2, the `Department Name` column, when a selection is made from the browse list. The choices in the browse list can be constrained by using any column in the browse list, but a selection must be made from the return column in order for values to be entered in the field. If you do not specify a column, the `Return Column` field value defaults to 1.

For more information on setting up and using browse lists in Forms Design, see “Chapter 18. Working with the Browse Feature,” on page 129.

Displaying the Browse List

You can determine the behavior of the browse list with the `Browser Display` field.

The display mode for a browse list can be either manual or automatic. If you decide to set the `Browser Display` to `Manual`, the browse list is displayed only if the `Browse` button is clicked on. Use this method when the browse list is large, or when users are familiar with the contents of the browse list. A user who knows what value needs to be entered does not want to wait for the browse list to be displayed.

If you want to encourage the user to select an entry through the browse list, set the `Browser Display` to `Auto` and the browse list is displayed automatically when the user selects in the field. The user can still enter information directly into the field, but it is easier to select the entry from the browse list. Use this method when the accuracy of the entry is imperative, or when users are less familiar with the contents of the browse list.

Entering the Browser Parameters

You can constrain the choices in the browse list by using this field in conjunction with the `General Qualifier` field or the `SQL For Update Value` field. For more information, see “Chapter 18. Working with the Browse Feature,” on page 129.

Chapter 13. Designing Choice Fields

Choice fields offer you flexibility when designing forms by enabling you to create hierarchies of fields. The options discussed in this chapter are displayed in the field area of the field options windows when you select `User Choice` from the `Field Type` field. This chapter describes how to:

- Present choices
- Place dependent fields
- Set the dimensions of choices
- Define choices
- Set dependencies between fields

See “Chapter 11. Entering Data in a Data Entry Form,” on page 67 for a complete overview of the types of fields available to you. For information about options you can set in the other sections of the field options windows, see “Chapter 16. Initializing Fields,” on page 117 and “Chapter 17. Storing Data,” on page 125. For a choice field in a retrieval form, see “Constraining Information Retrieval” on page 114.

Learning About Choice Fields

You define a field as a choice field when you want the user to select one of several possible choices. By creating a choice field, you reduce the number of keystrokes that must be entered, and also ensure that only valid values are entered.

A choice field is similar to an entry field that has a browse list. The main differences are:

- The choices in a choice field are displayed in the Data Entry window.
- The browse list for an entry field has too many entries to display in a choice field.

Therefore, if the list of possible values for a field is relatively small, and the entire list can be displayed in the Data Entry window, then the field can be defined as a choice field.

Creating Choice Fields

To create a choice field:

1. Select a field in one of the form depictions in the main Forms Design window.

2. Open the field options window. This window looks similar to the Entry Field Options window shown in Figure 18 on page 64.
3. Select `User Choice` in the `Field Type` field. The field area options for a choice field in an entry form are displayed.
4. Close the field options window.

The field options for the choice field are discussed in "Setting Preferences for Choice Fields" on page 87.

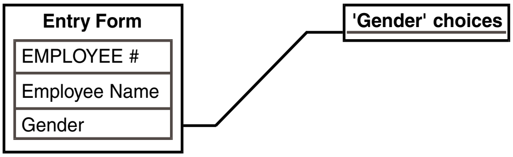
Adding Choice Field Choices

After you have defined a field as a choice field, you can create choices for the field.

To add choices to a choice field:

1. Double-click on the choice field in the form depiction. In this case, double-click on the `Gender` field in the `Entry Form` depiction.

   A heading for the 'Gender' choices is displayed. A line connects the choices heading to the field that it defines.



2. Select the choices heading.

   The choices heading becomes highlighted and the buttons in the main window header are replaced by a single `Add Choice` button.
3. Click on the `Add Choice` button.

   Each time you click on the `Add Choice` button, a field labeled `<CHOICE>` is added under the choices heading.

Naming Choices

After you have added choices to a choice field, you need to give them names that are meaningful to your users.

To name choices:

1. Select a choice within the depiction of your field in the Forms Design window.
2. Open the Choice Options window.
3. Enter a choice name in the `Choice Line 1` field.

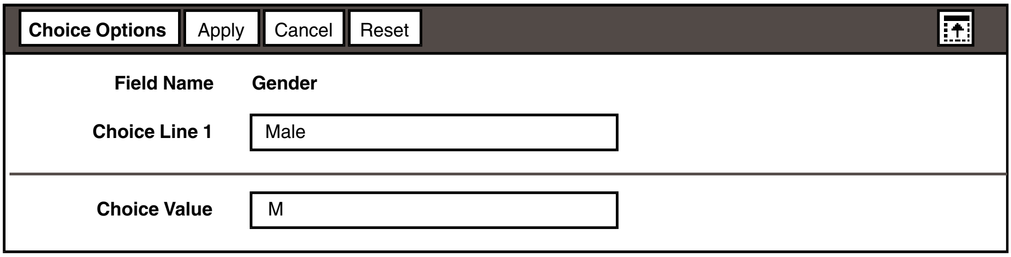
This field contains the actual choice name that is displayed in the form. For example, if you type `Male` in the `Choice Line 1` field, the `Gender` field contains a `Male` choice.

If you set the `Height (Lines)` field to 2 or more, the Choice Options window displays separate fields for each line in each choice.

4. Enter a value in the `Choice Value` field.

   When you predefine choices, you also redefine the choice value; the choice value is assigned a default number according to the order in which it was added. The choice value is a code associated with the choice.

   For example, the choice value for the `Male` choice can be changed from the default value of 1 to the letter M, as shown Figure 26 on page 83.



Field Name Gender

Choice Line 1 Male

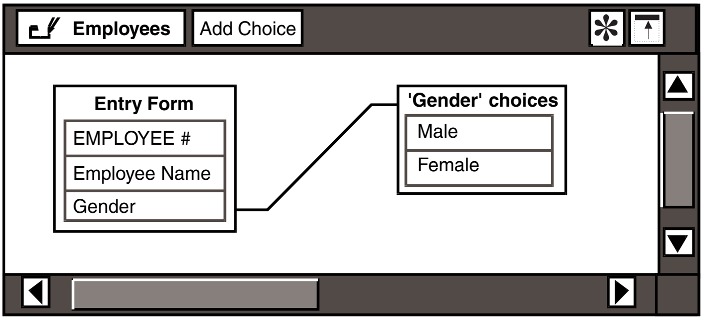
Choice Value M

Figure 26. Setting values for a choice

Although the user selects a choice name from a choice field, you can specify that either the choice name or the choice value be stored in the database. See “Chapter 17. Storing Data,” on page 125 for more information on storing the choice name and choice value. See “Chapter 16. Initializing Fields,” on page 117 for information on initializing a predefined choice from the database.

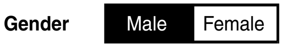
5. Close the Choice Options window. When the window closes, the name of the first field changes to `Male`.

6. Repeat the previous steps to define other choices. These steps were repeated to define the `Female` field.



7. Click on the original field in the form depiction to close the choices heading area. In this example, select the `Gender` field to close the 'Gender' choices heading area.

   The `Gender` field is displayed with the two choices in the Data Entry window:

   Gender Male Female

   Typically, the choice name, rather than the choice value, is stored in the database. However, if you want the choice value to be stored in the database, you have to specify additional information in the storage area of the field options window. See “Chapter 17. Storing Data,” on page 125 for more information on storing the choice name and choice value.

Layering with Choice Fields

If you have a choice field for which you have defined choices, you can create dependent fields by defining fields under each choice. By creating dependent fields, you create hierarchies of fields and choices.

For example, the `Marital Status` field in an Employees table is a choice field that has been defined with several choices, such as Single, Married, Divorced, and Widow.

Each of the choices shown in the `Marital Status` field displays more fields. For example, if `Married` is selected then the `Spouse's Name` field is displayed. The `Spouse's Name` field is dependent on the selection of the `Married` choice.

You can also define choices to show other fields when selected. In other words, choices can have other choices that are dependent on them.

You can layer dependent fields as deep as necessary from the base predefined field to capture the information you need.

With a choice field, you can also provide the option to enter information rather than select one of the available choices. For example, Figure 27 shows an additional dependent field, `# of Dependents`, that is displayed only when `Yes` is chosen from the `Dependents?` field. The `# of Dependents` field has a choice, `Other`, that allows the user to enter a number different than the choices given in the `# of Dependents` field.

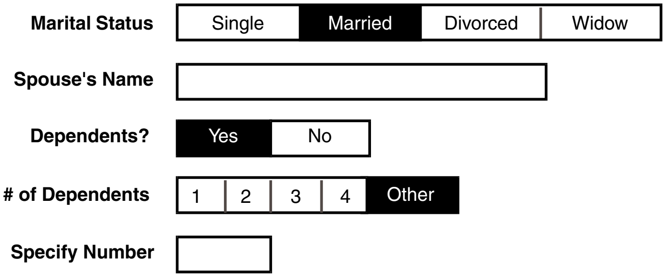


Figure 27. Dependent fields for the Marital Status field

When two fields in an entry form correspond to the same database column, the Data Entry tool stores the data from the field that is at the lowest level. In the following example, both the `# of Dependents` and `Specify Number` fields correspond to the same column in the database table.

If `1` is selected from the `# of Dependents` field, then the Data Entry tool stores the contents of the `# of Dependents` field in the database. However, if `Other` is selected from the `# of Dependents` field, then the Data Entry tool stores the contents of the `Specify Number` field in the database.

Creating Dependent Fields

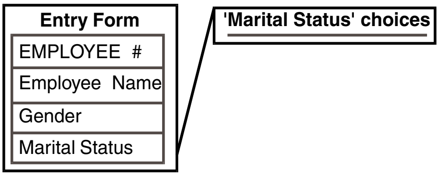
To create dependencies between fields, you follow the same basic procedure no matter what type of dependent field you are creating. Further, you can create as many levels of dependent fields as necessary to gather the needed information. However, only choices that you create can have dependent fields; choices derived from database values cannot have dependent fields.

To create a dependent field:

1. Select a choice field in a form depiction in the main Forms Design window.

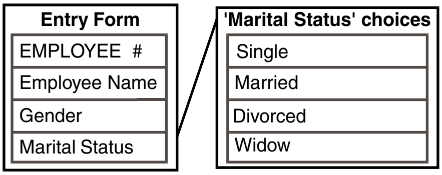
   Use the `Marital Status` field shown in Figure 27 as the example. Select its depiction in the Forms Design window.
2. Open the field options window.
3. Select `Predefined` in the `Choice Definition` field.
4. Close the field options window. You have now specified that you are defining the choices for this field.
5. Double-click on the depiction of the field to display a choices heading.

For example, if you double-click on the `Marital Status` field, the 'Marital Status' choices heading is displayed.



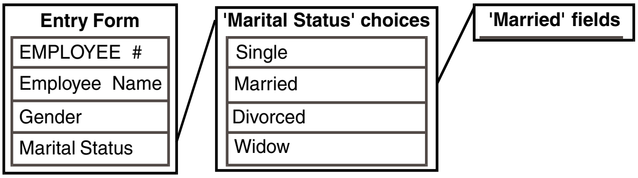
6. Add choices to the choice heading.

   For example, you could define `Single`, `Married`, `Divorced`, and `Widow` as the choices for the 'Marital Status' choices heading. See “Adding Choice Field Choices” on page 82 for more information on adding choices to a field.



7. Double-click on one of the choices that you added to the choices heading.

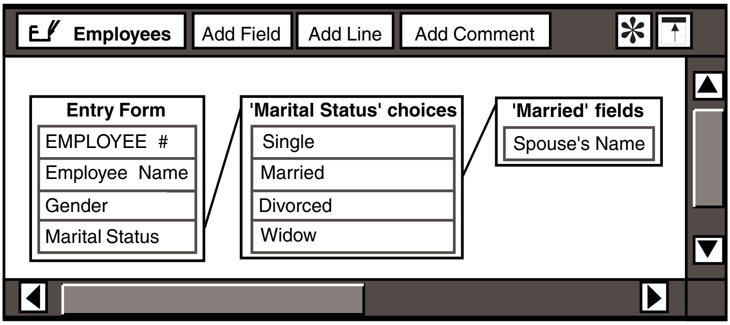
   A heading is displayed for the choice that you selected. For example, if you double-click on `Married`, the 'Married' fields heading is displayed.



   You can now define the dependent fields for the `Married` choice.

8. Add dependent fields to the fields heading.

For example, add a field named `Spouse's Name` to the 'Married' fields heading.



9. Select the dependent field that you added to the heading, if it is not already selected. For example, select the `Spouse's Name` field.
10. Open the field options window.
11. Define the dependent field as necessary.

    For example, to create the `Spouse's Name` field, you select `User Entry` in the `Field Type` field and type `Spouse's Name`.
12. Close the field options window. The field is now displayed as a dependent field when the appropriate choice is selected.

    In the example, the `Spouse's Name` field is now displayed as a field in the Data Entry icon when `Married` is selected from the `Marital Status` field.

Setting Preferences for Choice Fields

You have several options when designing choice fields. You can set options to control how choices are defined and displayed in choice fields.

To set your preferences for a choice field:

1. Select a choice field in one of the form depictions in the main Forms Design window.
2. Open the field options window.
3. Select a choice for each field.
4. Close the form options window when complete.

Table 11. Choice field options

Field	Options	Function
Field Type	User Choice	Displays the options for a field.

Table 11. Choice field options

Field	Options	Function
Prompt	N/A	Prompts the user to perform an action, such as enter a value. The Forms Design tool initializes this field to the WTDD description of the column copied from the category window. For more information about entering a prompt message, see “Chapter 11. Entering Data in a Data Entry Form,” on page 67.
Choice Style	Flat	Displays the choices within the field horizontally.
	Menu	Displays the choices within the field in a pull-down menu.
Choice Width	Constant	Makes each choice large enough to contain the widest choice.
	Variable	Makes each choice width differ according to the length of the choice.
Choice Definition	Predefined	You determine the choices for the field.
	Database	When you select `Predefined` in the `Choice Definition` field, the `Height (Lines)` field and `Placement of Dependent Fields` fields appear. The database generates the choices according to the `SQL For Choices` field. When you select `Database` in the `Choice Definition` field, the `SQL For Choices` field appears.
Height (Lines)	1 to 12	Sets the number of lines of text that appear for each choice.
SQL For Choices	N/A	Enters the SQL select statement that generates the choices.
Placement of Dependent Fields	Below	Places the dependent fields for a choice below the field.
	To the Right	Places the dependent fields for a choice next to the field.

Presenting Choices

You can specify the presentation style of choices within a choice field. The choices can be presented horizontally or in a pull-down menu.

If you select `Flat` as the presentation style, the field is displayed as shown in Figure 28 on page 88.

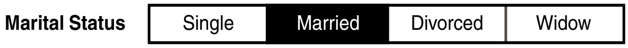


Figure 28. Choices in a flat presentation style

If you select `Menu` as the presentation style, the field is displayed as shown in Figure 29.

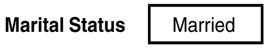


Figure 29. Choices in a menu presentation style

When the user selects the visible choice, a pull-down menu with several other choices is displayed. The menu is displayed only when the user selects on the visible choice. The user can change the visible choice by moving the cursor down to highlight the desired choice as shown in Figure 30.

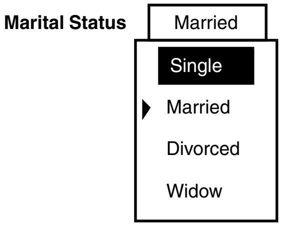


Figure 30. An expanded menu

Notice the arrow to the left of the Married choice. The arrow points to whichever choice is the current visible selection within the field. When a different selection is made, the field is displayed showing only the new choice.

The menu style is useful if you have several choices for your field, and you want to conserve space on your form by having the other choices display only when the visible choice is selected.

Setting the Choice Width

You can specify the width of the choices in your field. You can make each choice a standard width, or you can vary the width to suit the length of each choice.

To set a standard display width for each choice, select `Constant` in the `Choice Width` field. The display width of every choice is the same, and the width is equal to the width required to display the longest choice name in the field. Figure 31 shows a field with a constant choice width.

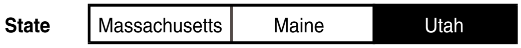


Figure 31. Choice fields set to constant widths

To vary the width of each choice, select `Variable` in the `Choice Width` field. When you have many choices, selecting `Variable` helps save space. The display width of each choice is determined by the length of the particular choice name. Figure 32 on page 90 shows a field with a variable choice width.

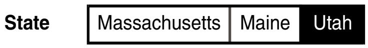
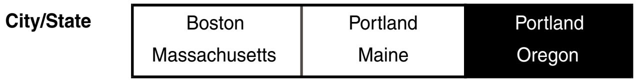


Figure 32. Choice fields set to variable widths

Setting the Choice Height

You can specify the height of the choices in your field. You can determine the height by the number of lines within the `Height (Lines)` field.

For example, if each choice consists of a city and state, you can select `2` in the `Height (Lines)` field so that each choice is displayed in two lines.



Defining Choices

You select a choice in the `Choice Definition` field to determine the data source for a choice field. The choices in the `Choice Definition` field create fields with data that are either:

- Defined by you, the forms designer
- Generated dynamically from the database server

These data sources create two types of choice fields and are described in the following sections.

Defining the Choices Yourself

When creating the form, the `Predefined` choice allows you to specify the exact choices from which a user can select. This method of defining choices is described in “Adding Choice Field Choices” on page 82.

Generating the Choices Dynamically

The `Database` choice in the `Choice Definition` field of the field options window lets you request that the database generate field values. The database then provides the choices from which a user selects.

For choices that are database-defined, the choice value assigned to each choice is its sequence number within the field. Choices are numbered from left to right, or top to bottom, beginning with 1.

To generate choices from the database for a choice field:

1. Copy a field from the category window to a form depiction in the main Forms Design window. As an example, consider the Employees and Departments tables shown in Figure 33 on page 91.

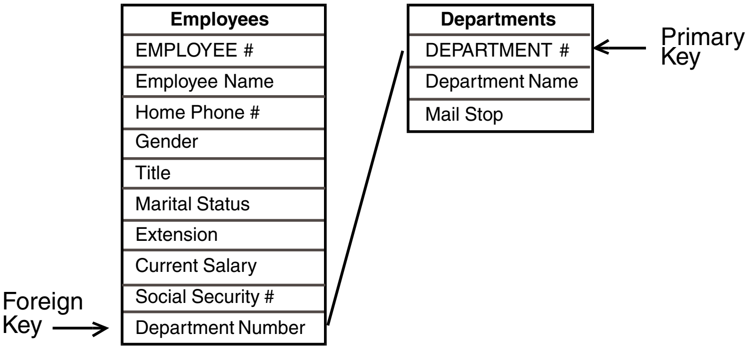


Figure 33. Foreign and primary keys in the category window

   There are two types of fields you can copy from the category window:

   - Foreign key field

     The `Department Number` field in the Employees table is a foreign key; it references the primary key (`DEPARTMENT #`) in the `Departments` table. The choices for this field should be the contents of the DEPARTMENT # column in the Departments table.

   - Primary key field

     The `DEPARTMENT #` field in the Departments table is a primary key for the Departments table.

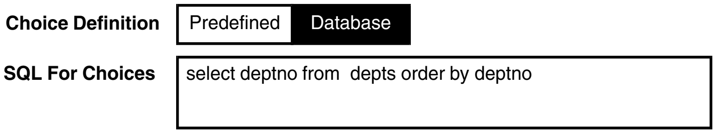
   If the field that you copy from the category window is a foreign key, you might want the choices that are displayed in the field to be generated from information stored in the table that defines the key values. This method ensures a faster response time and generates distinct values for the choice field.

   For example, if you copy the `Department Number` field from the Employees table into an Entry Form depiction, you can define that field as a choice field and specify that the choices for that field should be the primary key values in the Departments table.

2. Open the field options window.
3. Select `UserChoice` in the `Field Type` field.
4. Select `Database` in the `Choice Definition` field. The `SQL For Choices` field is displayed.
5. Enter an SQL statement in the `SQL For Choices` field.

   When a field is database-defined, you must provide an SQL statement that the Data Entry tool executes to generate the choices for the choice field. The SQL should generate distinct choices so that there are no duplicate values. The Data Entry tool executes this statement when the icon is opened.

The choices generated by the example shown in Figure 34 contain all the department numbers that are defined in the dimension table. This information is requested in the `SQL For Choices` field, where the database table to be used is named.



Choice Definition Predefined [✓ Database]

SQL For Choices select deptno from depts order by deptno

Figure 34. SQL SELECT statement used to generate choices

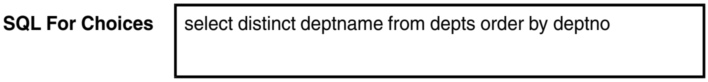
The `SQL For Choices` field is displayed only when `Database` is selected from the `Choice Definition` field. When you select `Predefined`, the `Height (Lines)` field and `Placement of Dependent Fields` fields are displayed instead of the `SQL For Choices` field.

You can use @-keywords in the `SQL For Choices` field. See “Using @-Keywords” on page 187 for information about the @-keywords.

6. Enter a new SQL statement in the `SQL For Choices` field if you want to further define the choices to be used for the choice field.

   Be sure to constrain the SQL statement sufficiently so that an appropriate number of choices are returned from the database.

   In this example, you can further define the choices to contain the department names, instead of the department numbers. Notice that the SQL statement in Figure 35 on page 92 is slightly different from the one shown in Figure 34 on page 92. The term *distinct* is included in the SQL statement to guarantee that no duplicate values are returned from the database.



SQL For Choices select distinct deptname from depts order by deptno

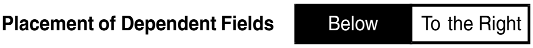
Figure 35. SQL SELECT statement used to return distinct values

   When you define a field to contain department names and you want to store the department numbers in the database, you must specify additional information in the storage area of the field options window. See “Chapter 17. Storing Data,” on page 125 for more information.

7. Close the field options window when complete.

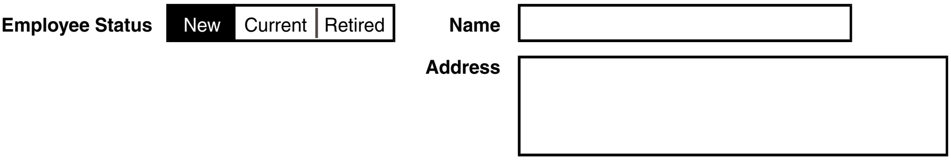
Positioning Dependent Fields

You can determine the placement of dependent fields for a choice in relation to the field. You can display the dependent fields below or to the right of the choice field.



Placement of Dependent Fields [✓ Below] To the Right

To display dependent fields to the right of the field, select `To the Right` in the `Placement of Dependent Fields` field. In this case, the `Name` and `Address` fields are displayed as shown in the following illustration:



Employee Status [✓ New] Current Retired

Name

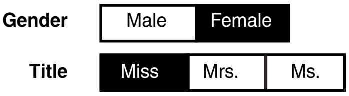
Address

Notice that when you place the dependent fields to the right of the choice field, they are placed as a block. If you want the dependent fields placed horizontal to one another and to the field, you need to specify a group for the dependent fields. See “Chapter 19. Enhancing Forms with Design Techniques,” on page 145 for more information about groups. See “Creating Dependent Fields” on page 85 for more information about defining dependent fields for choices.

Enhancing Retrieval Form Choice Fields

When you create a retrieval form that contains choices, you should be careful to create fields that allow retrieval of the necessary data.

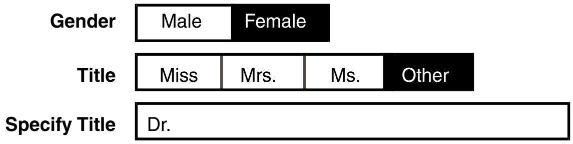
For example, if you use the choices shown in the following illustration for a `Title` field in a retrieval form, the user is only able to retrieve rows for which the `Title` field is set to Miss, Mrs., or Ms.



Gender Male [✓ Female]

Title [✓ Miss] Mrs. Ms.

This construction excludes rows for which the `Title` field might be Dr. To enable users to retrieve other titles, you should include `Other` in the choice field. When the user selects `Other`, a `Specify Title` field is displayed, and the user can then specify a title.



Gender Male [✓ Female]

Title Miss Mrs. Ms. [✓ Other]

Specify Title Dr.

Chapter 14. Building Flexibility with the Section Field

Section fields are useful when you need to break information into distinct segments, and display only one segment of information at a time. The options discussed in this chapter are displayed in the field area of the field options window when you select the `Section` choice in the `Field Type` field. This chapter describes:

- Creating section fields
- Defining dependent fields for section choices
- Presenting section choices
- Placing dependent fields
- Setting the height and width of section choices

See “Chapter 11. Entering Data in a Data Entry Form,” on page 67 for an overview of the types of fields available. For information about options you can set in the other sections of the field options windows, see “Chapter 16. Initializing Fields,” on page 117 and “Chapter 17. Storing Data,” on page 125. For section fields within the retrieval form, see “Constraining Information Retrieval” on page 114.

Learning About Section Fields

Define a section field when the information in the form you are creating can be organized into independent segments. Also, unlike choice fields, a user can make selections across the sections of a section field without losing the information, and without entering the selections until finished. Entries to the entire section field are validated when the user completes data entry and clicks on the `Enter` button.

A section field is similar in structure and operation to a choice field that has been predefined. The main features of the section field are:

- The application process

  Use the section field when your form contains multiple sections that must be completed before validating the data.

- The validation process

  Use the section field when the validation process must involve all dependent fields within the section field, not only those displayed under the current selection.

- The data entry process

  Use the section field when you want the user to complete all sections before clicking on the `Enter` button.

Creating Section Fields

To create a section field:

1. Select a field in one of the form depictions in the main Forms Design window.
2. Open the field options window.
3. Select `Section` in the `Field Type` field. The field options window for the section field opens.

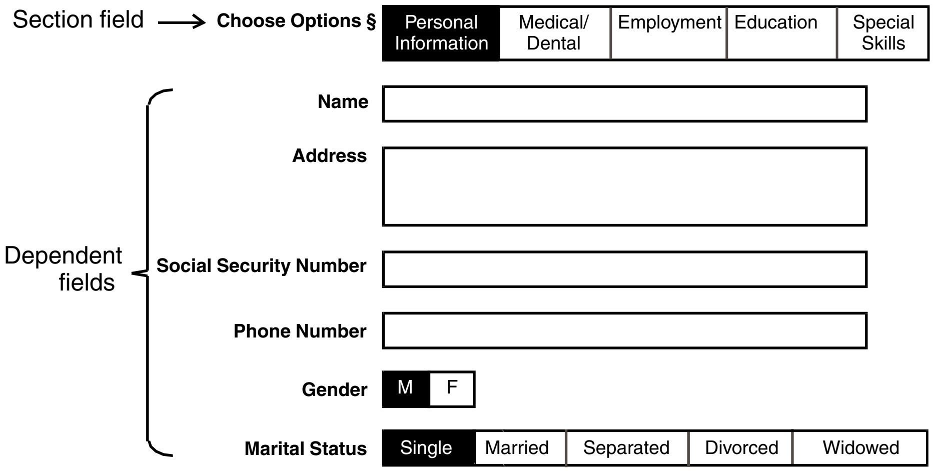


Figure 36. Field area options for a section field

The field options for the section field are discussed in “Setting Preferences for Section Fields” on page 99.

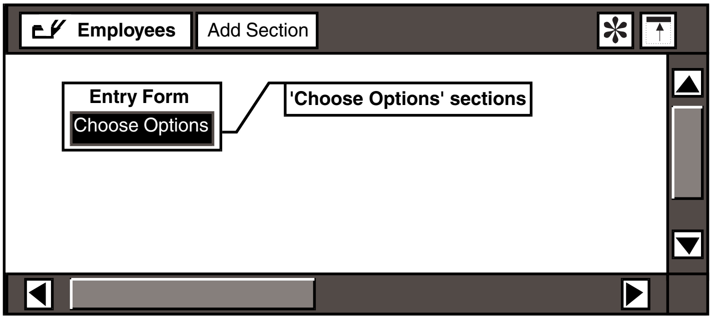
Adding Section Field Choices

After you have defined a field as a section field in the field options window, you can create section choices.

To create choices in a section field:

1. Double-click on the section field in the form depiction in the main Forms Design window.

A section heading is displayed with a line that connects it to the field that it defines. In the case of the `Choose Options` section field, the heading reads `'Choose Options' sections`.



2. Select the section heading. The section heading becomes highlighted and the buttons in the header of the main window are replaced by a single `Add Section` button.
3. Click on the `Add Section` button.

   A field labeled `<SECTION>` is added under the section heading. Each field you create represents a choice within the section field.
4. Select a choice under the section heading.
5. Open the Section Options window.

   If you do not specify the `Height` as two lines in the field options window, then only one `Section Line` field is displayed.
6. Enter a name in the `Section Line` fields.

   The name entered in the `Section Line` fields is the section choice that is displayed in a data entry form. If the section choices contain multiple lines, a `Section Line` field is displayed for each line in the section choice.
7. Enter data in the `Section Value` field.

   The section value is a code associated with the section choice. Although the user selects a section name, you can specify that either the name or the value be stored in the database.
8. Close the Section Options window.
9. Repeat steps 4 through 8 for each section choice that you want to create within the section field.

After you create and define all the section choices for the `Choose Options` section field, the depiction of the section is displayed as shown in Figure 37.

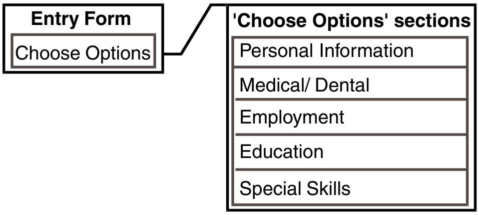


Figure 37. Section choices

Organizing Information into Sections

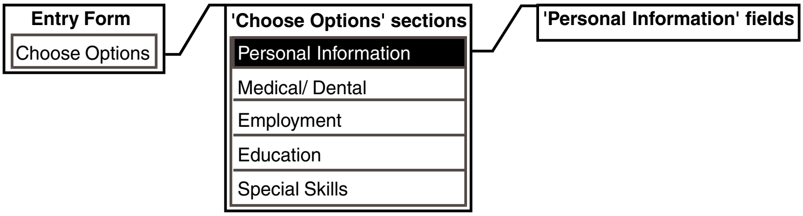
If you have a section field, you can organize information into appropriate sections. Under each section choice, create hierarchies of fields and choices.

In the following example, the `Choose Options` section field is used.

To create a field that depends on a section choice:

1. Double-click on one of the section choices to display a section fields heading.

   For example, if you double-click on the `Personal Information` section choice, the `'Personal Information' fields` heading is displayed as shown in the following illustration.



   You can now define the dependent fields for the `Personal Information` section choice.

2. Select the section fields heading. The `Add Field`, `Add Line` and `Add Comment` buttons are restored in the window header.

3. Copy a field from the category window to the new field's heading, or use the `Add Field` button.

   To create a `Name` field, add a field to the new field's heading with the `Add Field` button. Open the field options window for the new field and select `Entry` in the `Field Type` field. Enter *Name* as the field label.

You can create dependent fields of any field type, and even embed section fields within one another. Experiment with the flexibility of the Forms Design tool. The layering technique was designed to save you form space.

Setting Preferences for Section Fields

To set your preferences for the section field:

1. Select a section field in one of the form depictions in the main Forms Design window.
2. Open the field options window. The field options window for the section field is displayed.
3. Select a choice for each field.

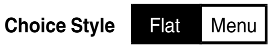
   All of the options in the field type area are described in Table 12 on page 99. A section field is used to enter data in an entry form, update data in an update form, and constrain data in a retrieval form.
4. Close the field options window when complete.

Table 12. Section field options

Field	Options	Function
Field Type	Section	Displays the options for a section field.
Prompt	N/A	Prompts the user to perform an action, such as enter a value. The Forms Design tool initializes this field to the WTDD description of the column copied from the category window. See “Chapter 11. Entering Data in a Data Entry Form,” on page 67 for more information about entering a prompt message.
Choice Style	Flat	Displays the choices within the field horizontally.
	Menu	Displays the choices within the field in a pull-down menu.
Placement of Dependent Fields	Below	Places the dependent fields for a choice below the section field.
	To the Right	Places the dependent fields for a choice to the right of the section field.
Choice Width	Constant	Makes each choice width large enough to contain the longest choice name.
	Variable	Makes each choice width differ according to the length of each choice name.
Height (Lines)	1 to 12	Sets the number of lines of text that appear for each choice.

Presenting Choices

You can specify the presentation style of choices within a section field. The choices can be presented horizontally or in a pull-down menu.



If you select `Flat` as the presentation style, the field is displayed as shown in Figure 38.

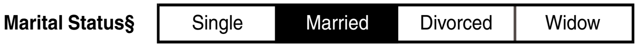


Figure 38. Choices in a flat presentation style

If you select `Menu` as the presentation style, the field is displayed as shown in Figure 39.

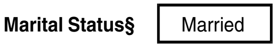


Figure 39. Choices in a menu presentation style

When the user selects the visible choice, a pull-down menu with several other choices is displayed. The menu is displayed only when the user selects the visible choice. The user can change the visible choice by moving the cursor down to highlight the desired choice as shown in Figure 40.

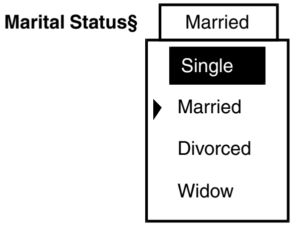
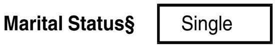


Figure 40. An expanded menu

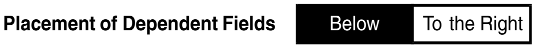
Notice the arrow to the left of the `Married` choice. The arrow points to whichever choice is the current visible selection within the field. When a different selection is made, the field is displayed showing only the new choice.



The menu style is useful if you have several choices for your field, and you want to conserve space on your form by having the other choices display only when the visible choice is selected.

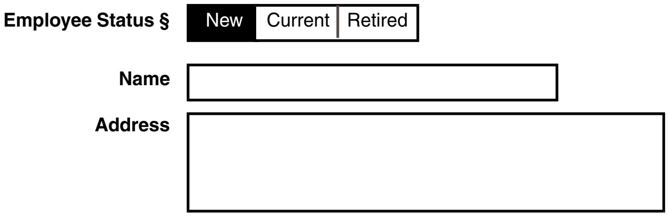
Positioning Dependent Fields

You can determine the placement of dependent fields for a choice in relation to the section field.



Placement of Dependent Fields [✓ Below] To the Right

Figure 41 shows a section field, `Employee Status`, with dependent fields for the `New` choice placed below it.



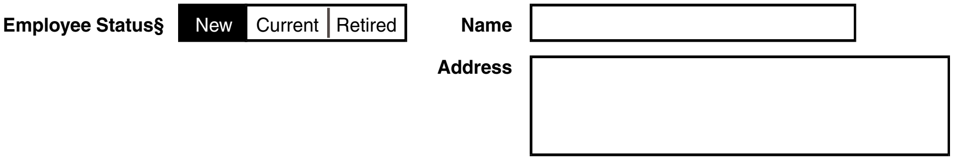
Employee Status § [✓ New] Current Retired

Name

Address

Figure 41. Dependent fields placed below a section field

You can also display the dependent fields to the right of the section field by selecting `To the Right` in the `Placement of Dependent Fields` field. In this case, the `Name` and `Address` fields are displayed as shown in Figure 42.



Employee Status§ [✓ New] Current Retired

Name

Address

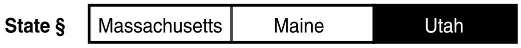
Figure 42. Dependent fields placed to the right of a section field

Notice that when you place the dependent fields to the right of the section field, they are placed as a block. If you want the dependent fields placed horizontal to one another and to the section field, you need to specify a group for the dependent fields. See “Chapter 19. Enhancing Forms with Design Techniques,” on page 145 for information about groups. See “Organizing Information into Sections” on page 98 for information about defining dependent fields.

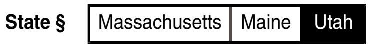
Setting the Choice Width

You can specify the width of the choices in your field. You can make each choice a standard width, or you can vary the width to suit the length of each choice.

To set a standard display width for each choice, select `Constant` in the `Choice Width` field. The display width of every choice is the same, and the width is equal to the width required to display the longest choice name in the section field. The next example shows a section field with a constant choice width.



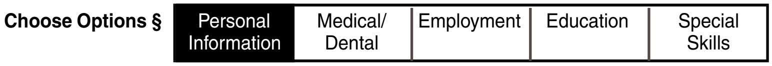
To vary the width of each choice, select `Variable` in the `Choice Width` field. When you have many choices, selecting `Variable` helps save space. The display width of each choice is determined by the length of the particular choice name. The next example shows a section field with a variable choice width.



Setting the Height of a Section Field

You can determine the height of a section field by selecting a choice in the `Height (Lines)` field. The height of a section field is measured relatively by lines of text.

For example, the following section field has two lines specified in the `Height (Lines)` field.



Chapter 15. Validating Data

You can set several options to control how the Data Entry tool validates data in fields. You can also set the editing options that are available for fields containing character strings. The majority of the options described in this chapter are displayed in the validation area of the field options window when `User Entry` is selected. Validation options for choice and section fields in the retrieval form are also discussed. This chapter describes:

- Entry options for the user
- Acceptable data types
- Valid database values
- Validation methods
- Default validation information
- Validation of choice and section field entries for the retrieval form

Choosing Validation Options

You can set the following validation options for fields:

- Entry must be single
- Value must be an acceptable data type
- Value must be within a range of values
- Character string must match a pattern
- Value must be a valid database value
- Value must be in a browse list

To display the field options window:

1. Select a field in a form depiction in the main Forms Design window.
2. Open the field options window. This window looks similar to the Entry Field Options window shown in Figure 18 on page 64.

   The options shown in the validation section of the Entry Field Options window are the same as the validation options in the Retrieval Field Options and Update Field Options windows for a field.

The Data Entry tool applies validation options to entries in the order in which the options are displayed in the field options window. For example, the Data Entry tool first verifies whether the entry should be validated as directed by the `Validate Entry` field. Then, if validation is necessary, the Data Entry tool verifies whether multiple entries are allowed in the field. This procedure continues until all of the specifications set in the validation area are verified against the entry.

Determining the Validation Method

You can specify when an entry within a field is validated by selecting a choice from the `Validate Entry` field. This field defaults to the Always choice.

Always

Entries are validated each time the user clicks on the `Enter` button. Use this method for dynamic databases where information is constantly changing.

If Manually Changed

Entries are validated only when manually changed, then entered. Use this method when the information in the database is fairly static.

A field is considered manually changed when the user types the entry directly from the keyboard or changes an existing entry.

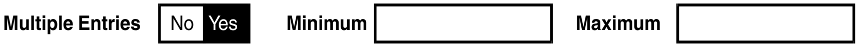
A field is not considered manually changed and, consequently, should not be validated when the entry is deleted completely, the entry is selected from the browse list, or the entry has already been validated.

Specifying Single or Multiple Entries in an Entry Field

You can limit a field to one data item, or allow several data items, by using the `Multiple Entries` field.

When you select `No` in the `Multiple Entries` field, the Data Entry form assumes only one value will be entered into the field. Therefore, commas are interpreted as part of a string rather than as a separator between entries. For example, the entry `Lotus, Meta5` is interpreted as one entry rather than two separate entries. Select `No` when the user is entering strings that include a comma.

To instruct the Data Entry form to look for multiple entries within a field, select `Yes` in the `Multiple Entries` field. When you select `Yes`, the `Minimum` and `Maximum` fields are displayed. These fields let you specify how many items are accepted within the field.



Multiple Entries No [✓ Yes] Minimum Maximum

The `Multiple Entries` field is supported by the entry and update forms. Also, you cannot allow multiple entries when entering data into a database.

The following conditions apply to multiple entries within a field:

- The Minimum parameter must be an integer greater than or equal to 0. If 0 is entered, then input is not required for the field; if a value greater than or equal to 1 is entered, then input is required.
- The Maximum parameter must be an integer greater than or equal to the minimum number.
- The Minimum and Maximum field values can be a maximum of 255.
- The data items must be separated by a comma and a space within the entry field. If a data item already contains a comma and a space, the entire item must be enclosed within single quotes. For example, 'Miami, Florida'.
- Each data item within the field is validated individually.

If you want the Data Entry form to accept only one entry, and to post an error message at multiple entries, then select Yes in the Multiple Entries field. Then set both the Minimum and Maximum fields to 1 to ensure only one entry.

Making Input Required or Optional

In the Input field, you can specify whether input is required in the field.

Select Required if the user must enter a value in the field. If you select Optional, no entries are necessary in the field.

- When Multiple Entries is set to Yes, the Minimum and Maximum fields replace the Input field. You can still make input optional by setting the minimum to 0.
- If you select Optional in the Input field for an integer, real, or date field, and the user leaves a blank field, the Data Entry tool checks the Workstation Tools Data Dictionary (WTDD) to see if an N/A value is defined for that column. If the value is defined, the Data Entry tool enters that value into the database column. If no N/A value is defined, that column is not specified in the insert command. On most database systems, the lack of a defined N/A value results in a null value in the database.

Specifying an Acceptable Data Type Value

You can specify the data type for a field by selecting the appropriate choice in the Accept field.

- Integer

  Whole numbers for which you specify a maximum and a minimum value.
- Real

  Numbers that can contain a decimal point for which you specify a maximum and a minimum value. See “Designing an Entry Form” on page 53 for information about how to set the default precision for fields that contain real numbers.
- Character

Strings of alphanumeric characters for which you specify the maximum length. The maximum length of a character string is 32,000.

- Date

  Dates that you can format by day, month, year, or bimonth. You can also specify that the date be within a maximum and a minimum period.

Creating Integer Value Ranges

Set the range for integer entries by entering high and low values in the `Minimum` and `Maximum` fields. See “Specifying Database Validation” on page 111 for more information on database validation.

To set validation options for a field that contains an integer:

1. Select `Integer` in the `Accept` field. Default `Minimum` and `Maximum` values are displayed.
2. Enter a new value into the `Minimum` field. For example, if you want to create a range of values for an `Employee Number` field, set the field to accept a minimum value of `1`.
3. Enter a new value into the `Maximum` field. You could set a top range for the `Employee Number` field to accept a maximum of `500`.

Creating Real Number Value Ranges

Set the range of real number entries by entering high and low values in the `Minimum` and `Maximum` fields.

To set validation options for a field that contains a real number:

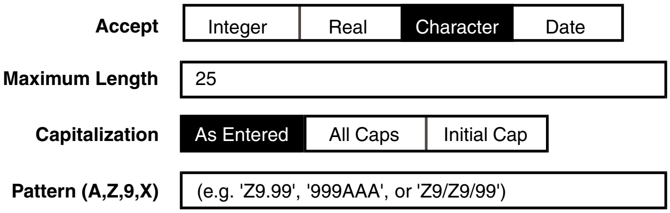
1. Select `Real` in the `Accept` field. Default `Minimum` and `Maximum` values are displayed.
2. Enter a value into the `Minimum` field. For example, if you want to create a range of values for a monthly salary field, set the field to accept a minimum value of `1000.00`.
3. Enter a value into the `Maximum` field. You could set a top range for the monthly salary field to accept a maximum of `7999.99`.

Setting Character String Restrictions

You can specify that a character string entry must be a certain length, capitalization style, and pattern.

To set validation options for a field that contains a character string:

1. Select `Character` in the `Accept` field.



Accept Integer Real [✓ Character] Date

Maximum Length 25

Capitalization [✓ As Entered] All Caps Initial Cap

Pattern (A,Z,9,X) (e.g. 'Z9.99', '999AAA', or 'Z9/Z9/99')

2. Enter a value into the `Maximum Length` field.

   Determine the maximum number of characters that can be entered. For example, to set validation options for a field containing Social Security numbers, type `11` in the `Maximum Length` field. You need a total of 11 spaces because there are 9 digits and 2 hyphens in the Social Security number.

   The maximum length allowed for a character string is 32,000. If the user enters more characters than you specify as the maximum length, an error message is displayed.

3. Select a choice from the `Capitalization` field.

   Decide how the Data Entry tool edits field entries. If you select `As Entered`, no changes are made to the entered character string. A selection of `All Caps` means that all characters are automatically capitalized. Select `Initial Cap` to automatically capitalize only the first character.

4. Enter a pattern into the `Pattern` field.

   The value that is entered into the field must match the pattern that you create. If the pattern entered does not match the pattern you specify, an error message is displayed.

   For example, to create a pattern for Social Security numbers, type `999-99-9999` in the `Pattern` field. The hyphens are imposed in the pattern so that the Social Security number must be entered in the specified format.

Table 13 on page 107 defines the acceptable character formats that you can use in a pattern.

Table 13. Pattern characters

Pattern characters	Acceptable characters
`A`	a-z, A-Z
`\A`	A
`9`	0-9
`\9`	9
`Z`	0-9, space, null

Table 13. Pattern characters

Pattern characters	Acceptable characters
`\Z`	Z
`X`	any character
`\X`	X

Table 14 on page 108 contains several examples of patterns, and shows acceptable and unacceptable data for each pattern.

The following conditions apply when creating patterns for a field:

Table 14. Pattern examples

Pattern	Acceptable data	Unacceptable data
999-99-9999	103-32-8172	2494-44-4321
AA-999	AB-121	A2-352
	aa-502	AA-59a
\AA-999	AZ-352	aC-123
aa-999	aa-123	AA-123
XZZ	@55	AB
	@ 5	B1A
Z9:99	1:23	:23
	12:34	1:2

- A backslash symbol (`\`) indicates that only the character directly following it is acceptable. For example, if the pattern in the `Pattern` field is `\A`, then the only character allowed is `A` (this pattern is case-sensitive, so `a` is unacceptable). However, if the pattern is `A` (without the backslash), then any character from `A` to `Z` is acceptable (including lowercase).
- Other characters, such as the hyphen in a social security number, can be imposed in a specific position within a pattern indicating that only that character can be entered.
- Pattern characters cannot be limited to a specified group, such as `A` to `F`.

Creating Date Value Ranges

You can specify that the values for a date must be within a certain range by setting a precision for the date, and entering values in the `Minimum` and `Maximum` fields.

To set validation options for a field that contains a date:

1. Select `Date` in the `Accept` field. Descriptive date options are displayed in Figure 43 on page 109.

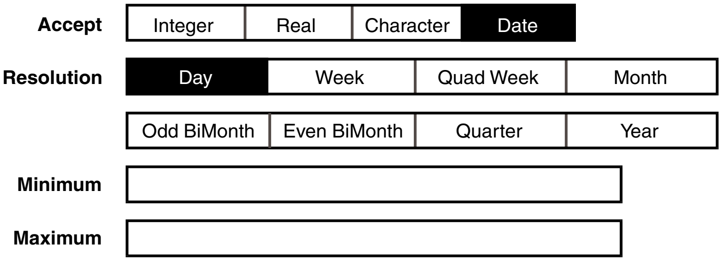


Figure 43. Setting date value ranges

2. Select a choice in the `Resolution` field.

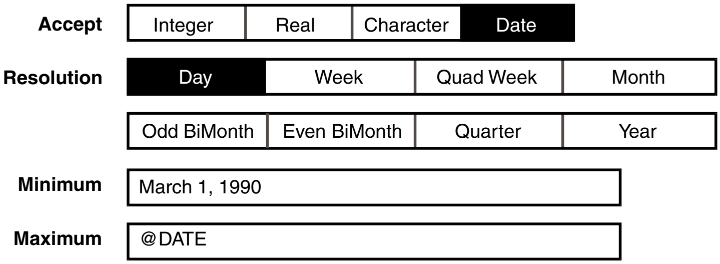
   Break down the date into the nearest day, month, bimonth, or year. For example, select `Day` if you want information specific to certain days.

3. Enter a value into the `Minimum` field that coincides with your choice in the `Resolution` field.

   For example, if you select `Day` in the `Resolution` field, enter a specific day into the `Minimum` field. You enter a date such as `March 1, 1990`.

4. Enter a value into the `Maximum` field.

   Enter a date in the same format as you did for the `Minimum` field. You also have the option of entering `@DATE`. The @-keyword @DATE represents the current date.



Minimum March 1, 1990

Maximum @DATE

This example shows a date field resolved to the nearest day, with a value range between March 1, 1990 and the present date.

Table 15 on page 110 lists the acceptable formats for entering dates.

Table 15. Acceptable date formats for direct entry

Date type	Correct formats	Incorrect formats
Day	January 15, 1994	Jan. 15, 1994
	January, 15, 1994	1-15-94
	January.15.94	
	Jan 15, 1994	
	jan 15, 94	
	jan 5 94	
	Jan/15/94	
	15/January/94	
	94.January.15	
	1/15/94	
	15.1.94	
	94-1-15	
	this day	
	today	
	Jan 15, 94 + 2 days	
Week	jan 15, 94 - 2 weeks	
	this week	
	week of Jan 15, 94	
	15 w 94	
	week of 1/15/94	
Quad Week	quad week of Jan 15, 94	

Table 15. Acceptable date formats for direct entry

Date type	Correct formats	Incorrect formats
Month	January, 1994 Jan, 1994 january, 94 jan, 94 jan 94 this month 1/94 + 2 months	Jan., 1994 jan, 1994 January 1-94
Odd BiMonth	JF, 1994 jf 94 1994 JF	Jan-Feb, 1994
Even BiMonth	JF, 94 jf 1994 1994 JF	Jan-Feb, 94
Quarter	1q94 1Q94 this quarter	First Quarter of 1994
Year	1994 94 this year	`94

Specifying Database Validation

You can specify how the entered value must be validated by using the `Database Validation` field. Use this field with each of the data types that you specify in the `Accept` field.

The following sections of this chapter describe each of the four choices in the `Database Validation` field:

- (None)

  The value entered does not require database validation. Any value entered that meets the minimum and maximum criteria is accepted.

- In Browser

  A list must be generated from an applicable database table. The user browses this list and selects a valid value. For example, display a browse list if the entries consist of lengthy product names. See “Chapter 18. Working with the Browse Feature,” on page 129 for more information.
- Not In

  The value entered should not exist in a database table. Use this choice if the entry must be a unique value; for example, an invoice order number.
- In

  The value entered must exist in a database table. Use this choice if the entry is in a format that is commonly understood, such as a date. Supplying a browse list of dates takes extra time to display, and uses unnecessary desktop disk space and random access memory (RAM).

The Not In and In choices require an SQL statement.

Setting No Database Validation

You can create a field that allows the user to enter any value, as long as the data type is correct. The data is not validated against a database table.

To create a field that allows any value, select `(None)` in the `Database Validation` field.

The `Database Validation` field is displayed.

Validating Data with a Browse List

You can create a field that allows the user to enter or select only those values that exist in a browse list.

To create a field that validates with a browse list, select `In Browser` in the `Database Validation` field.

The `Database Validation` field is displayed.

Browse lists are usually updated only when requested. However, depending upon how frequently the database changes, you can specify that the lists be updated each time the icon is opened, each time the lists are needed for validation, or based on a specified SQL statement. For more information about browse lists, see “Chapter 18. Working with the Browse Feature,” on page 129.

Preventing Duplicated Data

You can ensure that entries are not duplicated by specifying that the value entered is not currently in a database table. This type of validation is useful for fields that contain primary keys.

To ensure that entries do not exist in a database table:

1. Select `Not In` in the `Database Validation` field.

   Two additional fields are displayed along with the `Database Validation` field: `SQL For Values` and `For Database Validation`.
2. Enter an SQL statement in the `SQL For Values` field. See "Entering an SQL Statement" on page 113 for more information about SQL.
3. Select a choice from the `For Database Validation` field.

   Choose `Do Not Quote Field Values` if the entry should be validated as entered, such as for numeric field values.

   Choose `Quote Field Values` if the entry should be validated as a character string and enclosed in single quotation marks. Always use `Quote Field Values` for character string field values.

Validating Data in a Database Table

You can specify that the entry must exist in a database table. This type of validation is useful for fields that contain foreign keys.

To validate entries against a database table:

1. Select `In` in the `Database Validation` field.

   Two additional fields are displayed along with the `Database Validation` field.
2. Enter an SQL statement in the `SQL For Values` field. See "Entering an SQL Statement" on page 113 for information about SQL.
3. Select a choice from the `For Database Validation` field.

   Choose `Do Not Quote Field Values` if the entry should be validated as entered, such as for numeric field values.

   Choose `Quote Field Values` if the entry should be validated as a character string and enclosed in single quotation marks. Always use `Quote Field Values` for character string field values.

Entering an SQL Statement

When you select `Not In` or `In`, the `SQL For Values` field is displayed below the `Database Validation` field.

The statement that you enter in an `SQL For Values` field generates the database values used for validation. For example, `select emp_no from Personnel`.

The &CONSTRAINT function is useful for constructing an SQL constraint for a field. See "Inserting or Updating Data in More Than One Table" on page 161 for more information.

You can also use @-keywords in the SQL statements. See "Using @-Keywords" on page 187 for information.

See the *Data Access Tools User's Guide* for more information about SQL statements in Meta5.

Using the Query Tool to Enter an SQL Statement

You can use the Query tool to create the SQL statement in the `SQL For Values` field.

To obtain an SQL statement using the Query tool:

1. Create a depiction of the query that generates the desired values from the database.
2. Capture the SQL statement for that query.
3. Copy that SQL statement into the field options window of the Forms Design tool.

See the *Data Access Tools User's Guide* for more information about using queries and SQL statements in Meta5.

Learning About the Default Validation

The Forms Design tool creates some default validation information for fields. When you copy a field from the category window into a form in the main Forms Design window, the Forms Design tool accesses information in the WTDD to determine the default validation information for the selected field.

The Forms Design tool:

- Uses WT_COLUMNS to determine the type of data contained in that field
- Selects the appropriate choice in the `Accept` field in the field options window
- Enters the Minimum and Maximum field values to allow for the range of values

Constraining Information Retrieval

The Retrieval Field Options window constrains information received from the database for the choice and section field types (no matter how the choices were defined for these fields). The `Compare` field determines the method of constraint:

- Choice Name

Constrain the information by the choice name given in the Choice Options window. If you choose Choice Name, then the actual title is used as a constraint for information retrieval. For example, in a `Gender` field, the titles Male and Female are used as constraints. See “Chapter 13. Designing Choice Fields,” on page 81 for information about choice names.

- Choice Value

  Constrain the information by the choice value given in the Choice Options window. If no choice value has been defined, the relative sequence number of the choice within its field is used, beginning with one. This method is useful when a field contains descriptive names, but you want to use a code to represent the choice. For example, in a `Gender` field you might want to use the value `M` for Male, and `F` for Female to constrain information retrieval. See “Chapter 13. Designing Choice Fields,” on page 81 for information about choice values.

- Database

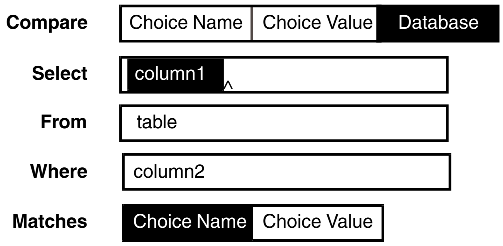
  Constrain the information with the database value generated by executing the SQL select statement.

If you select `Choice Name` or `Choice Value`, no new options are added. When you select `Database`, you can specify the dimension table and the column in the database to use as an information constraint. The entries you specify in these fields relate to the method of database storage you chose in the storage area. See “Chapter 17. Storing Data,” on page 125 for information about storage options.

To specify database information constraints:

1. Select `Database` in the `Compare` field.

   When you select `Database` in the `Compare` field, new fields that are associated with the database table are displayed.



Compare Choice Name Choice Value [✓ Database]

Select column1

From table

Where column2

Matches [✓ Choice Name] Choice Value

2. In the `Select` field, enter the name of the column in the database table that contains the primary key. This name identifies the column that the Data Entry tool uses to constrain the retrieval.
3. Enter the name of the database table in the `From` field.
4. In the `Where` field, enter the name of the column that contains the choices that are displayed in the field.

5. Select a choice in the `Matches` field. Choose `Choice Name` to specify that the `Where` field column contains choice names. Select `Choice Value` to specify that the `Where` field column contains choice values.

Chapter 16. Initializing Fields

Another way the Forms Design tool gives you flexibility when designing your form is by letting you set the initial value of a field. The options described in this chapter are displayed in the initialization area of the field options windows. The initialization options differ for the three forms, and the differences are described within this chapter. This chapter describes:

- Initialization options for fields
- Initialization options for choice and section fields
- Initialization options for predefined fields
- Field reset options
- Default initialization information

Finding Initialization Options

You can set initialization options for all field types to determine the source and description of the initial value. The initialization options vary depending on the choices that you select in the `Field Type` and `Accept` fields, and the form in which your field resides. The initialization options area for the Entry Field Options window is shown in Figure 18 on page 64.

Initializing Entry and Predefined Fields

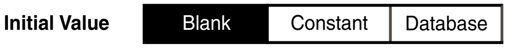
The `Initial Value` field for an entry field and a predefined field are essentially the same. However, the choices within the `Initial Value` field vary for an entry field depending upon the selection made in the `Accept` field of the validation section. The choices in the `Initial Value` field are different for the following data types:

- Integers and real numbers
- Character strings
- Dates

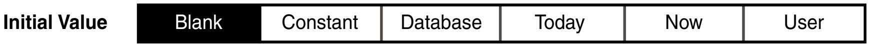
Selecting Initial Values for Entry Fields

The initialization choices for an entry field vary depending upon the data type of the field. The data type is selected in the `Accept` field. The variations in initialization choices are as follows:

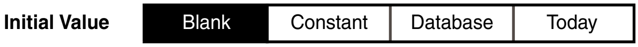
- If the data type is an integer or real number, the `Initial Value` field is displayed.



- If the data type is a character string, the `Initial Value` field is displayed as shown below.



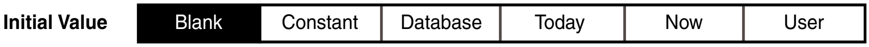
- If the data type is a date, the `Initial Value` field is displayed as shown below.



In all of the previous illustrations, replace the `Blank` choice with `Column` for the update form.

Setting Initial Values for Predefined Fields

For the predefined field, the `Initial Value` field is displayed.



Replace the `Blank` choice with `Column` for the update form.

Discovering Initialization Methods

The choices explained in Table 16 apply to both entry and predefined fields.

Table 16. Initial Value field choices for entry and predefined fields

Choice	Initial value
Blank	A blank field is displayed.
Column	The value from the first row of the column selected to be updated is displayed. (for update forms only)
Constant	The value from the `Constant Value` field is displayed.
Database	A database value is displayed, determined by data from the `Compute` field.
Today	The current date is displayed.
Now	The current date and time are displayed.

Table 16. Initial Value field choices for entry and predefined fields

Choice	Initial value
User	The user's desktop name is displayed.

When you select Blank, Today, Now, or User, no new options are added. When you select Column, Constant, or Database, new options are added. These options are described in the following sections. The following sections pertain to all three forms except where specified.

Initializing a Field to a Column

You can specify that the value in the first row of the selected column is displayed in the field when the update form is opened. This option only applies to the update form.

To set a field to a column name:

1. Select Column in the Initial Value field.
2. Enter a column name into the Column Name field.

   For example, to return the first value in the Employee # column of the Personnel table, enter Personnel.employeenumber into the Column Name field.

Initializing a Field to a Constant

You can specify a constant value to be displayed in a field each time a form is opened. Initializing a field to a constant is helpful when you have a field with data that rarely needs to change.

To set a field to a constant value:

1. Select Constant in the Initial Value field.
2. Enter a value into the Constant Value field.

   For example, if your field relates to tax status, type Exempt as the constant value for an Employee Status field.

   You can use Forms Design @-keywords in the Constant Value field. See “Using @-Keywords” on page 187 for information about @-keywords.

Initializing a Field to a Database Value

You can specify that the database provide the initial value for a field. Initializing a field to a database value is helpful when you have a field that relies on variable data.

To set a field to a database value:

1. Select Database in the Initial Value field.
2. Select a choice in the Compute field.

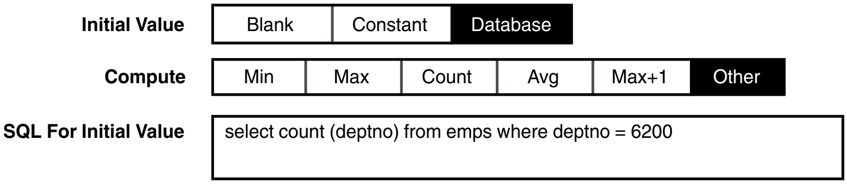
If you select `Min`, `Max`, `Count`, or `Avg`, the initial field value is the aggregate result of the column name.

If you select `Max+1`, the initial field value is the current maximum value stored in the `Column Name` field plus one. This computation method is useful for primary key fields that contain sequence numbers. For example, to add a new employee into an `Employees` table, you use a predefined field with an initial field value of `Max+1`.

If you select `Other`, the initial field value is the result of executing the SQL statement specified in the `SQL For Initial Value` field.

3. Enter a value for the `Column Name` field. Enter the name of the column on which the function is performed for the `Min`, `Max`, `Count`, `Avg`, and `Max+1` choices.

Select `Other` in the `Compute` field and enter an SQL statement in the `SQL For Initial Value` field. The initialization area is displayed:



Initial Value: Database

Compute: Other

SQL For Initial Value: select count (deptno) from emps where deptno = 6200

The `SQL For Initial Value` field in the above illustration contains a statement for a field initialized to the headcount in a specific department.

When you write the SQL statement, you must make sure that the statement returns a single value. If the statement does not return a single value, the Data Entry tool displays an error message, and initializes the field to the first value returned by the database server. See “Chapter 15. Validating Data,” on page 103 for more information on SQL.

You can use Forms Design @-keywords in the `SQL For Initial Value` field. See “Using @-Keywords” on page 187 for information about @-keywords.

Initializing Choice and Section Fields

You can initialize a choice or a section field to one of several possibilities. The options discussed here relate to the entry form and update form. The retrieval form does not provide initialization options for choice or section fields.

The initial value in a choice or a section field can be set to:

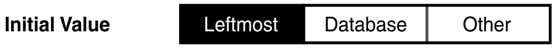
- The first choice in the field for an entry form
- The row selected to be updated in an update form
- A choice name, choice value, or choice number

A field can also be set to a value from the database. The first value returned from the database upon execution of the SQL statement is used to determine which choice is highlighted. Initializing a field to a database value is similar to initializing entry and predefined fields to the database.

To initialize a choice or section field:

1. Select `User Choice` or `Section` in the `Field Type` field.

   The `Initial Value` field for a field in an entry form is displayed in the following illustration. (Substitute `Column` for the `Leftmost` choice for an update form.)



2. Select a choice in the `Initial Value` field.

Learning Initialization Methods

Table 17 on page 121 explains each choice in the `Initial Value` field for choice and section fields.

Table 17. Initial Value field choices for choice and section fields

Choice	Initial value:
Leftmost	The leftmost choice is highlighted.
Column	The first row of the column selected to be updated is highlighted.
Database	The choice that matches the value determined by the SQL For Initial Value statement is highlighted.
Other	The choice name, value, or number specified in the `Initial Choice` field is highlighted.

When you select `Leftmost`, no new options are added. When you select the `Column`, `Database` or `Other` choices, new options are added. These options are described in the following sections.

Initializing a Field to a Column

This option only pertains to the update form. You can specify that the value in the first row of the column selected to be updated is the initial value.

To set a field to the column:

1. Select `Column` in the `Initial Value` field. The initialization area is displayed.
2. Enter a name in the `Column Name` field. For example, to return the first value in the Employee # column of the Personnel table, type `Personnel.employeenumber` in the `Column Name` field.

Initializing a Field to the Database

You can specify that the database choose the initial value for a field.

To set a field to a database value:

1. Select `Database` in the `Initial Value` field. The initialization area for a field in an entry form is displayed.
2. Enter an SQL statement in the `SQL For Initial Value` field.

   You must use the name of the choice in the SQL statement rather than the choice value. Also, when you write the SQL statement, you must make sure that the statement returns a single value. If the statement does not return a single value, the Data Entry tool displays an error message, and initializes the field to the first value returned by the database server.

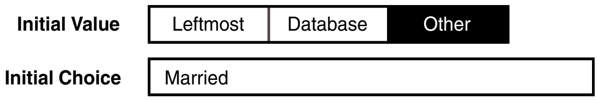
   The value returned by the SQL statement determines which choice is highlighted. If there is no match, the first choice is highlighted. You can also copy an SQL statement from a query. See "Chapter 15. Validating Data," on page 103 for more information on SQL.

Initializing a Field to a Specific Choice

You can specify exactly which choice you want highlighted when a form is displayed.

To specify an initial choice:

1. Select `Other` in the `Initial Value` field. The initialization area is displayed:



Initial Value: Leftmost | Database | Other

Initial Choice: Married

2. Enter a value for the `Initial Choice` field. You can enter a choice name, choice value, or choice number.

   When you use this method to initialize a choice or section field, the Data Entry tool determines how to initialize the field when the icon is opened. The Data Entry tool does not know if the value in the `Initial Choice` field that you entered is a choice name, a choice value, or a choice number. Figure 44 on page 123 illustrates how the Data Entry tool initializes a field to the `Initial Choice` value.

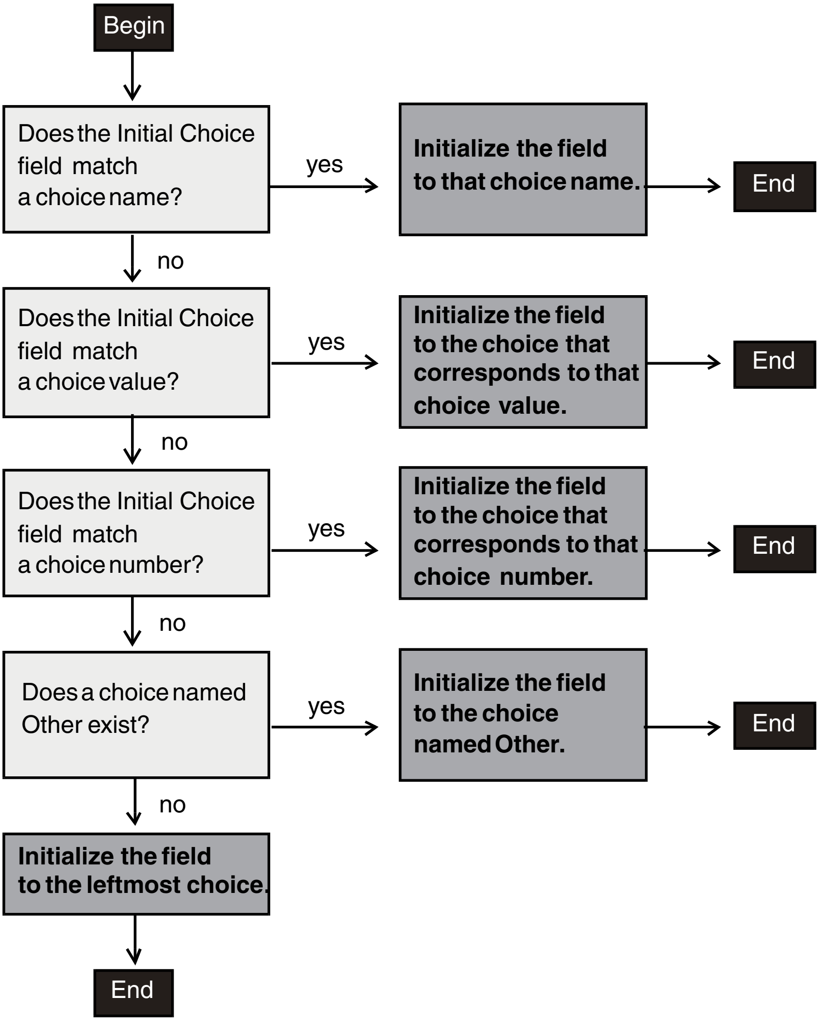

Begin
Does the Initial Choice field match a choice name?
yes
Initialize the field to that choice name.
End
no
Does the Initial Choice field match a choice value?
yes
Initialize the field to the choice that corresponds to that choice value.
End
no
Does the Initial Choice field match a choice number?
yes
Initialize the field to the choice that corresponds to that choice number.
End
no
Does a choice named Other exist?
yes
Initialize the field to the choice named Other.
End
no
Initialize the field to the leftmost choice.
End

Figure 44. Data Entry tool procedure for initializing a choice or section field

Changing Field Reset Options

You can set each entry, choice, predefined, or section field to reset every time the user clicks on the `Enter` button, or you can set the field to retain updated information. This function is available only for fields in the entry form.

For example, if the user wants to enter employee information by department, you can avoid repetition by setting the `Department Number` field to retain the current department number.

To change field reset options:

1. Select a choice in the `Initial Value` field.

Regardless of the choice, you can make a selection in the `When Entered` field.

2. Select `Reset Field` in the `When Entered` field if you want the field to return to the initial value after the user enters any new values into the database.

   Select `Retain Field` in the `When Entered` field if you want the field to retain any updated value after the user clicks on the `Enter` button. This choice clears the field only when the user clicks on the `Reset` button in the form window header. This method is useful if the user is entering several records in which one field is the same or similar for all records.

You can also use this feature to place a timestamp on entries. To do this, you create a predefined field whose `Initial Value` is `Now` and whose `When Entered` field is `Reset Field`. Using this method, the time is updated when the user clicks on the `Enter` button.

Learning About the Default Initialization

When you copy a field from the category window to a form depiction in the main Forms Design window, the initialization information for that field depends on the form in which the field resides.

The default initialization operates as follows:

- If the field is in an Entry Form depiction, the Forms Design tool sets the `Initial Value` field to Blank.
- If the field is in an `Update Form` depiction, the Forms Design tool sets the `Initial Value` field to Column. When the update form is used, the Data Entry tool recognizes the appropriate data column.

Except for predefined fields, fields that are copied from a category window into a Retrieval Form depiction do not have initialization information associated with them.

Chapter 17. Storing Data

You can set options to control where the Data Entry tool stores data for the entry and update forms. The options described in this chapter are displayed in the storage area of the field options windows. This chapter describes:

- Storing data in the database
- Storing data in @*-variables*
- Specifying no data storage
- Storing information according to the default

Locating Storage Options

You can specify where you want data stored for the entry and update forms. The `Destination` field in the field options windows gives you three storage choices:

- Database
- @-variable
- (nowhere)

The dependent fields for these choices vary by the selection made in the `Field Type` field.

To display the field options window:

1. Select a field in a form depiction in the main Forms Design window.
2. Open the field options window. The window looks similar to the Entry Field Options window in Figure 18 on page 64.

Storing Data in the Database

You can store data in the database for all fields that are copied from the category window. However, the procedure for storing entry and predefined field data is different from the procedure for storing choice and section field data. The Forms Design tool determines where field data is stored in the database.

Specifying Entry or Predefined Field Storage

To store entry or predefined field data in the database:

1. Select `User Entry` or `Predefined` in the `Field Type` field.
2. Select `Database` in the `Destination` field.

The Destination Column field is displayed. This field contains the name of the column that stores the data. You cannot change the column name because the Forms Design tool sets this field according to the name of the column that you copied from the category window.

Specifying Choice or Section Field Storage

To store choice or section field data in the database:

1. Select User Choice or Section in the Field Type field.
2. Select Database in the Destination field.

   The Destination Column is displayed as in the entry and predefined fields, as well as the Stored Value field. The Stored Value field allows you to choose which value you want stored in the database: choice name, choice value, or database value.
3. Select a choice in the Stored Value field. When you select Choice Name or Choice Value, no new options are added. When you select Database, new options are added, and are described in the following section.

   The choices in the Stored Value field are explained in Table 18.

Table 18. Database storage options for choice and section fields

Choice	Stored field data
Choice Name	The choice name given in the Choice Options window. For example, in a Gender field the choice names might be Male or Female.
Choice Value	The choice value given in the Choice Options window. For example, in a Gender field you might want to store the value M for Male, and F for Female. If no choice value has been defined, the relative sequence number of the choice in the field is used, beginning with one. Use the choice value when the field contains descriptive names but you want to store a code to represent the choice.
Database	The database value generated by executing the SQL SELECT statement.

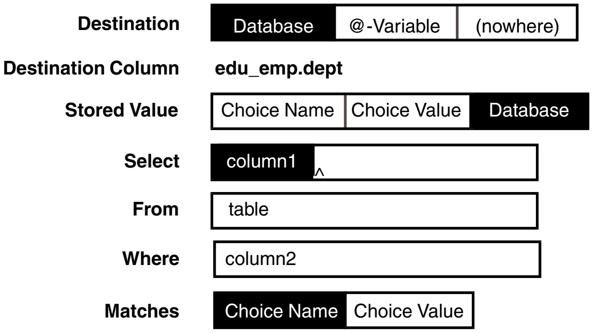
Specifying Database Storage for Choice or Section Fields

You can specify the dimension table and the column in the database that contain the data and choices to be stored.

To specify database storage information:

1. Select Database in the Destination field.
2. Select Database in the Stored Value field.

When you select Database in the Stored Value field, new fields that are associated with the database table are displayed. The choices for your choice or section field must contain descriptive names generated from the database.



Destination [✓ Database] @-Variable (nowhere)

Destination Column edu_emp.dept

Stored Value Choice Name Choice Value [✓ Database]

Select column1

From table

Where column2

Matches [✓ Choice Name] Choice Value

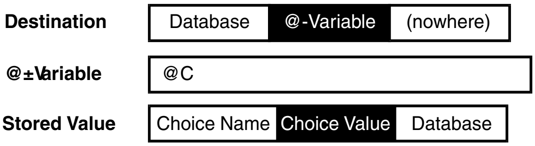
3. In the Select field, enter the name of the column in the database table that contains the primary key. This name identifies the data that the Data Entry tool stores in the field.
4. Enter the name of the database table in the From field.
5. In the Where field, enter the name of the column that contains the choices that are displayed in the field.
6. Select a choice in the Matches field. Choose Choice Name to specify that the Where field column contains choice names. Select Choice Value to specify that the Where field column contains choice values.

Storing Data in an @-Variable

When a Data Entry icon is used in a capsule application to set *@-variables*, you can specify that field data be stored in *@-variables*. When you create a Data Entry icon that has fields preset to specific *@-variables*, the Data Entry icon automatically enters current data into those specified User Input Control window.

To store data in an *@-variable*:

1. Select @-Variable in the Destination field.



Destination Database [✓ @-Variable] (nowhere)

@±Variable @C

Stored Value Choice Name [✓ Choice Value] Database

2. Enter an *@-variable* in the @-Variable field.
3. Select a choice in the Stored Value field. See “Specifying Choice or Section Field Storage” on page 126 for more information about the choices in the Stored Value field.

See “Chapter 20. Becoming an Expert Forms Designer,” on page 155 for more information on using @-*variables* in a Data Entry icon.

Specifying No Data Storage

You can specify that you do not want to store field data anywhere by selecting `(nowhere)` in the `Destination` field.

Specifying no data storage is useful when:

- The Data Entry icon operates inside a capsule application to set @-*variables*, and one or more fields is for display purposes only
- You set up hierarchies of fields in such a way that a field is needed only to group subordinate fields

Learning About the Default Storage

When you copy a field from the category window into an entry or update form depiction in the main Forms Design window, the Forms Design tool:

- Sets the `Destination` field to `Database`
- Sets the `Destination Column` field to the name of the column for the field that you copied from the category window

For example, if you copy a field named `Job Code` from the category window into a form in the main Forms Design window, the storage area is displayed.

The `Destination` field is set to `Database`, and the column name for the `Job Code` field is specified in the `Destination Column` field.

Chapter 18. Working with the Browse Feature

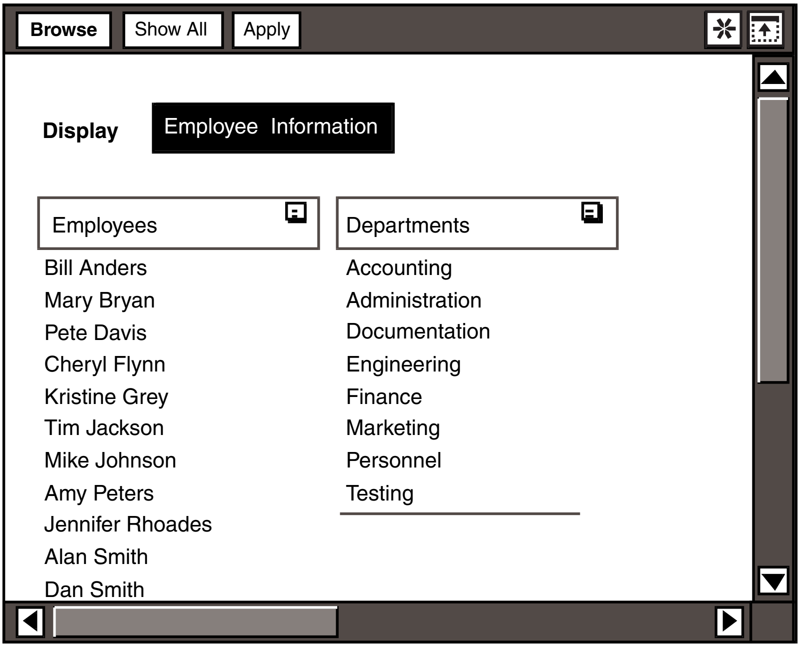
To provide greater flexibility with browse lists, the Forms Design tool enables you to establish a library of browse lists associated with a specific Forms Design icon. You can create and set options for browse lists, use one browse list for several fields, and constrain the information presented in a browse list. This chapter describes:

- Creating browse lists
- Updating browse lists
- Adding columns to a custom browse list
- Setting options for columns
- Using one browse list for several fields
- Converting existing Forms Design browse lists

The browse function associated with the Forms Design and Data Entry icons is similar, but not related to the independent Browser icon. Any browse lists you create in the Forms Design and Data Entry icons apply only to those particular icons. For information on the Browser icon, see the *Data Access Tools User's Guide*.

Setting Up Browse Lists

Browse lists are lists of choices displayed in browse windows. Among other uses, browse lists help users to enter information into fields more accurately. One way the user can display the browse list is by clicking on the `Browse` button in the window header of the Data Entry form. When you have defined a browse list, the browse window is displayed in the Data Entry form as shown in Figure 45.



Display: Employee Information

Employees
Bill Anders
Mary Bryan
Pete Davis
Cheryl Flynn
Kristine Grey
Tim Jackson
Mike Johnson
Amy Peters
Jennifer Rhoades
Alan Smith
Dan Smith

Departments
Accounting
Administration
Documentation
Engineering
Finance
Marketing
Personnel
Testing

Figure 45. Browse window

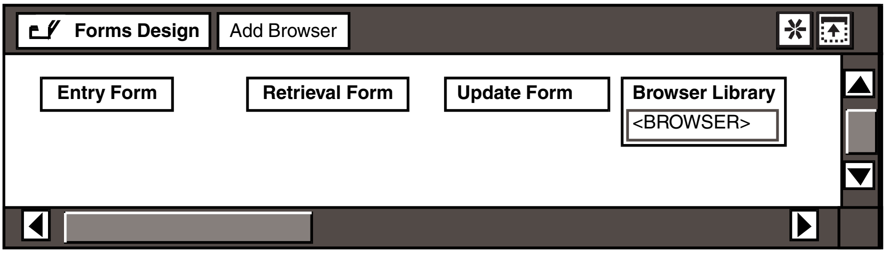
As forms designer, you decide how the browse list is displayed, when the browse list is updated, how the information returned to the browse list is constrained, and many more options. The process for creating and setting options for browse lists has several steps. The following sections describe a part of the process of creating browse lists.

Adding a Browse List to the Browser Library

The main Forms Design window contains a depiction of a browse list library. When you select the Browser Library heading, the buttons in the window header are replaced by a single button, `Add Browser`.

To add a browse list to the Browser Library depiction, click on the `Add Browser` button.

A field labeled `<BROWSER>` is displayed in the `Browser Library` depiction.



Setting Preferences for the Browse List

There are several options that you can specify for individual browse lists including the name of the browse list, when to update the browse list, and the location of the SQL statements that define the browse list.

To set the options for the browse list:

1. Select the `<BROWSER>` field unless it is already highlighted.
2. Open the Browser Options window. The Browser Options window opens as shown in Figure 46 on page 131.

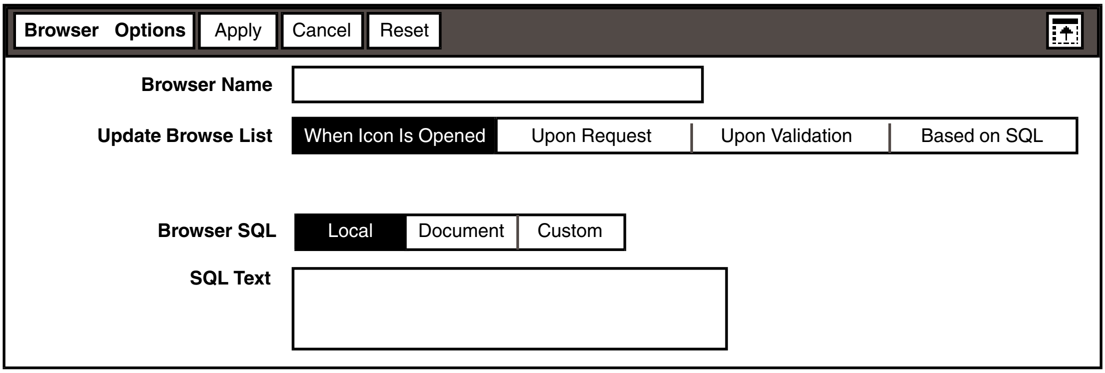


Figure 46. Browser Options window

3. Set the options for the browse list as needed.
4. Close the Browser Options window.

The options within the Browser Options window are described in Table 19 on page 131, and discussed in the following sections.

Table 19. Browse list options

Field	Options	Function
Browser Name	N/A	Specify the name of the browse list.

Table 19. Browse list options

Field	Options	Function
Update Browse List	When Icon Is Opened	Update the browse list every time the icon is opened.
	Upon Request	Update the browse list only when requested by the user.
	Upon Validation	Update the browse list when the browse list is used for validation.
	Based on SQL	Update the browse list according to the SQL statement specified in the `SQL For Update Value` field.
Browser SQL	Local	Enter SQL statements that define the contents of the browse list in the Browser Options window.
	Document	Name the document where the SQL statements that define the contents of the browse list are located.
	Custom	Create a custom browse list. See “Customizing Browse Lists” on page 135 for more information on custom browse lists.
SQL For Update Value	N/A	Enter SQL statements that determine if a browse list needs to be updated.
SQL Text	N/A	Enter SQL statements that define the contents of the browse list.
Document Name	N/A	Enter the location of the SQL statements that define the contents of the browse list.
General Qualifier	N/A	Specify a constraint on the browse list. The entry in this field is equivalent to a WHERE clause in SQL.

Naming the Browse List

The name you enter for the browse list should indicate the contents. When you look at a collection of browse lists in the Browser Library depiction, you need to be able to discern easily the contents of each browse list. For example, if a browse list contains an employee's name, address, and phone number, then name the browse list “Employee Contact Information.” Enter this name into the `Browser Name` field.

To associate the browse list with a field, you enter the browse list name in the `Browser Name` field of the field options window. See “Chapter 12. Constructing Entry Fields,” on page 73 for more information about browse list options in the field options windows.

Updating the Browse List

Select a choice in the `Update Browse List` field to control when a browse list is updated.

Depending upon how often the database changes, you can specify that the browse list be updated:

- When Icon Is Opened

  The browse list is updated every time the Data Entry icon is opened and the `Browse` button is clicked on for the first time.

- Upon Request

  The browse list is updated when `Update Browse Lists` is selected from the Data Entry icon `Special` menu.

- Upon Validation

  The browse list is updated when it is used for validation.

- Based on SQL

  The browse list is updated according to the SQL statements specified in the `SQL for Update Value` field.

When you select Based on SQL, an `SQL for Update Value` field is displayed. You enter an SQL statement into this field that gives the Data Entry tool a way to automatically determine when a browse list needs to be updated.

To update the browse list every time the first value has changed, enter an SQL statement in the `SQL For Update Value` field. For example:

```
select count (*) from employees
```

When the database is queried and returns a value, that value is compared against the value that was returned the last time the tool was opened and the SQL statement was issued. If the values are different, the Data Entry tool clears all the values in the browse list. The browse list is then updated when the user clicks on the `Browse` button. You should use any measure appropriate to your browse lists when creating this SQL statement.

If the database changes frequently, then the browse list should be updated frequently. However, updating the browse list is time consuming, so only update as necessary. See “Determining Data Entry SQL Performance” on page 183 for more information on performance issues related to browse lists.

Determining the Browser SQL

When deciding on the SQL with which to define browse lists, you should consider consolidating the fields that can share the same WHERE clause. See “Using One Browse List for Several Fields” on page 142 for more information. Select a choice in the `Browser SQL` field to control how the relevant information is passed from the database into the browse list.

There are three choices in the `Browser SQL` field:

- Local

  You specify SQL statements in the `SQL Text` field that provide the table name and the names of specific columns to access for the browse list. If you want the browse list to contain the Department # column from the Personnel database table, enter a statement into the `SQL Text` field. For example:

  ```
  select dept_no from Personnel
  ```

- Document

  You specify the name of the document in the `Document Name` field that contains the SQL statement. By storing the SQL in a separate document, you can use the SQL statement for many different Forms Design icons. This method also helps with maintenance of the browse lists in case of changes.

  For example, if 'Personnel' is the name of a document in the Browsers File Drawer, enter the document name and path in the `Document Name` field:

  ```
  system|File Drawers|Browsers|Personnel
  ```

  You can store the document on the desktop, in a container such as a Folder icon or File Drawer, in your capsule application, or any other location within Meta5. Be sure to provide the correct path to the document. Meta5 searches for the document in the current container or capsule icon first, then the desktop, and finally in the System|File Drawers.

- Custom

  You create your own columns for the browse list. This method is especially useful when you want to specify a formula which alters the information that is displayed in the browse list.

  When you specify Custom, you can enter a WHERE clause in the `General Qualifier` field that constrains the data allowed in the browse list (you performed the SELECT by adding columns to the browse list). For example, to show a browse list constrained to only return data for October 1, 1993 with quantities greater than 100, type `quantity > 100 and period > '10/01/93'` in the `General Qualifier` field.

  You can set up the browse list so that it can be used by different fields to access different parts of the browse list. The @-keywords @PARM1, @PARM2, through @PARM10 allow you to set up a constraint without specifying the conditions in the `General Qualifier` field. The specific conditions for each field are then given in the `Browser Parameters` field of the field options window. See “Setting Up Conditional Expressions with @PARMs” on page 166 for more information.

If you select either Local or Document, the browse list contains the columns specified by the SQL statements. If you select Custom, then you decide to create your own columns for the browse list. See “Customizing Browse Lists” for information about creating your own columns.

Customizing Browse Lists

Customizing browse list columns is useful for several reasons:

- You can combine columns from different tables in the same browse list. By copying columns directly from the category window to the 'Browser' columns heading, you can combine columns from different tables without becoming confused by complicated SQL statements.
- You can tailor the data type of the column according to the WTDD definition. This feature helps you show data in the most appropriate form. For example, you can show real numbers as integers.
- You can perform calculations on the data retrieved for the browse list, and display the altered information.

Choosing a Method of Customizing

You can customize a browse list in two ways:

- Using the category window

  Copy columns from the category window to the 'Browser' columns heading. Use this method when you want specific columns from database tables, but you do not want to write SQL statements. This method does not allow you to alter the values displayed by the browse list, or determine the display type for the information. The Browser Column Options window specifies the table and column heading for you as well.
- Using the `Create Column` button

  Add columns to the 'Browser' columns heading by clicking on the `Create Column` button. Use this method when you want to create a column that is not defined in the WTDD. You can also use this method if you want to specify a formula which alters the information you retrieve from the database for the browse list. For example, if the table you are accessing contains prices for items without sales tax figured in, you can display the price with the sales tax in the browse list.

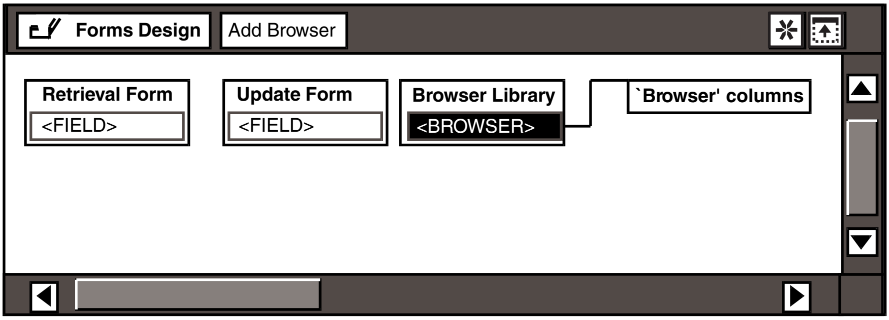
For both of these methods, you must specify `Custom` in the `Browser SQL` field of the Browser Options window. The `Custom` choice informs the Forms Design tool that you are creating your own columns using one of the methods outlined in this section. If you specify `Local` or `Document` in the `Browser SQL` field of the Browser Options window, the 'Browser' columns heading is not displayed.

Adding Columns to the Browse List

As described in “Choosing a Method of Customizing” on page 135, there are two ways to add columns to the browse list. With either method, you follow the same steps to reach the **'Browser' columns** heading. The following example shows how to add columns using the `Create Column` button.

To add columns to the browse list:

1. Be sure to set the `Browser SQL` field in the Browser Options window to `Custom` so that you can add columns to the browse list.
2. Double-click on the browse list field in the Browser Library depiction. The **'Browser' columns** heading is displayed as shown in the following illustration.



The **'Browser' columns** heading takes on the name of your browse list.

3. Select the **'Browser' columns** heading. The `Add Browser` button in the window header is replaced by the `Create Column` button.
4. Click on the `Create Column` button. A field labeled `<COLUMN>` is displayed under the **'Browser' columns** heading.

Designing Browse Columns

When you have added a column to the browse list, you can set several options that affect the display, sorting method, and contents of the column.

To set options for the browse column:

1. Select the column under the **'Browser' columns** heading.
2. Open the Browser Column Options window.

   The Browser Column Options window, for a column created with the `Create Column` button, opens as shown in Figure 47 on page 137.

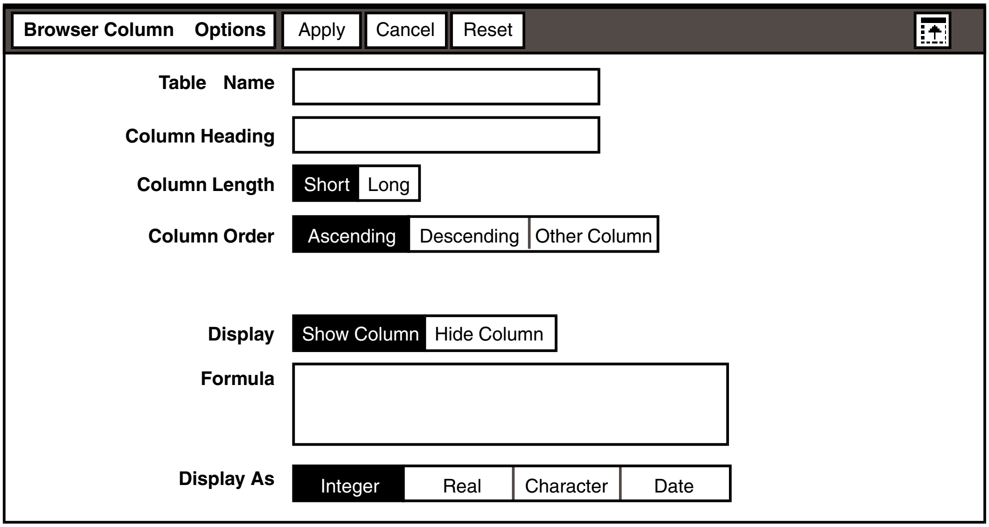


Table Name

Column Heading

Column Length: [✓ Short] Long

Column Order: [✓ Ascending] Descending Other Column

Display: [✓ Show Column] Hide Column

Formula

Display As: [✓ Integer] Real Character Date

Figure 47. Browser Column Options window for a column created with the Create Column button

3. Set the options as needed.
4. Close the Browser Column Options window.

If you copy the column from the category window, the Browser Column Options window looks slightly different.

Notice that the information for the `Table Name` and `Column Heading` fields are supplied by the Forms Design tool. Also, the `Formula` and `Display As` fields are missing from this window. When you copy fields from the category window, you cannot alter the information retrieved for the browse list or decide what data type the browse list uses to display the information. However, you can change the column name given in the `Column Heading` field.

The various options available in the Browser Column Options window are described in Table 20 on page 137 and discussed in the following sections.

Table 20. Browse list column options

Field	Options	Function
Table Name	N/A	Specify the name of the database table to be accessed by the browse list column. The information for this field is supplied by the Forms Design tool if you copy the column from the category window.
Column Heading	N/A	Specify the display name for the browse list column.
Column Length	Short	Estimate that the length of the column does not exceed the 100 item display limit for the browse list window.

Table 20. Browse list column options

Field	Options	Function
	Long	Estimate that the length of the column is likely to exceed the 100 item display limit for the browse list window.
Column Order	Ascending	Sort the column into ascending order.
	Descending	Sort the column into descending order.
	Other Column	Sort the column according to the column you specify in the `Order by Column` field.
Order by Column	N/A	Specify the name of the column to be used as a basis when sorting the current column.
Display	Show Column	Display the column in the browse list.
	Hide Column	Conceal the column in the browse list.
Formula	N/A	Enter SQL statements as a formula to generate customized columns to retrieve information into the browse list.
Display As	Integer	Display numeric data as whole numbers.
	Real	Display numeric data as numbers with decimal points for which you can specify a precision in the `Precision` field.
	Character	Display data as strings of alphanumeric characters.
	Date	Display data as dates for which you can specify a precision in the `Resolution` field.
Precision	0000.1 - .00005	Determine the placement of the decimal point for data that is defined as a real number.
Resolution	Day	Display the date in days.
	Week	Display the date in weeks.
	Quad Week	Display the date in four week periods.
	Month	Display the date in months.
	Odd BiMonth	Display the date as 1st and 3rd month periods.
	Even BiMonth	Display the date as 2nd and 4th month periods.
	Quarter	Display the date in quarter years.
	Year	Display the date in years.

Predicting Column Length

The browse list can store up to 100 items in workstation memory and display them in a single column. If a column exceeds this 100-item limit, the browse list does not display any items for the column until enough constraints have been placed on the data so that the number of items returned in the column does not exceed the capacity of workstation memory.

As forms designer, you must predict the number of items that will be returned in the columns that you create, and decide whether the column exceeds display limits. In the `Column Length` field, you choose the selection that is most appropriate.

Select `Short` if you think that the column will return less than 100 items, and Meta5 will store and display the column's contents from memory. Select `Long` if you think the column will return more than 100 items. For long columns, Meta5 does not store the column's contents in memory. Rather, Meta5 retrieves and displays the column's contents when the column has been constrained by choices in other columns, and the number of items to be retrieved in the column has been constrained to below 100.

Always begin with the `Short` choice in the `Column Length` field. If you later discover that some columns are using too much memory, or if you experience performance problems, you can specify that the columns with the most items are `Long`. Also remember that the width of a column can affect the amount of memory used. For example, a column that is 250 characters wide requires a lot of memory, and slows down performance both while defining the browse list and during later use.

Ordering Columns

You can specify how each column in the browse list is sorted by selecting a choice in the `Column Order` field.

You can sort the column by three ordering methods:

- Ascending

  The items in the column are displayed in ascending order – for example, Apple, Banana, Cherry; and 1,2,3.

- Descending

  The items in the column are displayed in descending order – for example, Cherry, Banana, Apple; and 3,2,1.

- Other Column

  The items in the column are displayed in the order of another column. This method keeps items that are related in the same order, but not necessarily in the same row. If an item is displayed in two rows of the column, it is only listed once in the proper order. For example, assume that the employees and departments actually correlate as shown in Table 21.

Table 21. Actual correlation of employees and departments

Employee	Department
Jan Adams	Marketing
Dave Jones	Personnel
Sharon Korth	Sales
Ben Marks	Marketing
Mary Smith	Engineering

To correlate the department to the employee, you define column two with the `Column Order` field set to `Other Column`, and specify column one in the `Order by Column` field. The table shown in Table 22 is displayed.

Table 22. Column two sorted by column one

Employee	Department
Jan Adams	Marketing
Dave Jones	Personnel
Sharon Korth	Sales
Ben Marks	Engineering
Mary Smith	

There are no duplicate instances of a department, but the departments are now displayed in the order of the employee names. The only limitation on the use of the column order feature is that the column listed in the `Other Column` field must be in the same browse list as the column you are defining.

Determining Column Display

You can specify whether or not to show a column in the browse list by selecting a choice in the `Display` field.

This feature is useful when you need to sort a column according to another column which you wish to hide. For example, if you want to sort employee names according to salary groups, but do not want to show the salaries.

Entering a Formula

By using the `Formula` field, you can perform calculations on the data retrieved by the browse list, and display the updated information. For example, if you have a browse list column that returns the salary of each employee, and you want to display the salary plus bonus, you enter a calculation into the `Formula` field. You could enter a calculation that returns the sum of salary plus bonus in the `Formula` field:

```
salary + bonus
```

The `Formula` field is useful when you want to place different constraints on individual browse list columns. This option is available only when you create columns with the `Create Column` button.

Setting Data Type

When you customize your columns with the `Create Column` button, you can specify the data type that the browse list uses to display information.

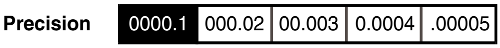
The choices in the `Display As` field are:

- Integer

  Displays numeric data as a number with no decimals – for example, 1, 20, or 323.

- Real

  Displays numeric data as a number with a decimal point – for example, 5.21. When you select this choice, the `Precision` field is displayed as shown in the following illustration.

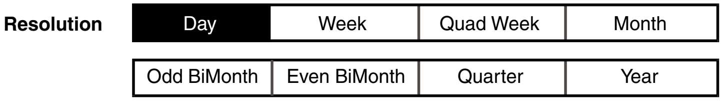
  Select a choice based on the number of decimal points you want.

- Character

  Displays data as a string of alphanumeric characters.

- Date

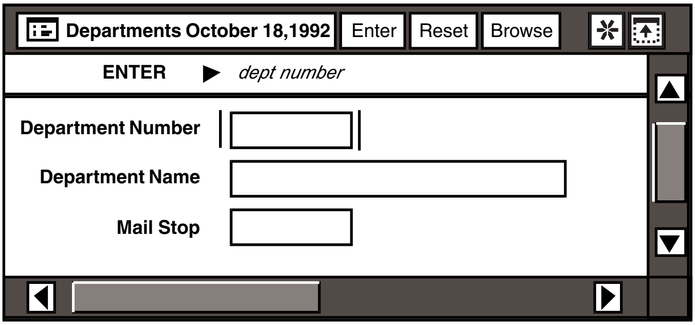
  Displays data as a date. When you select this choice, the `Resolution` field is displayed as shown in the following illustration.

  Choose the precision with which you want the browse list to display the date. See “Creating Date Value Ranges” on page 108 for more information about the choices offered by the `Resolution` field.

Using One Browse List for Several Fields

With the enhanced browse list functionality, you can use one browse list for several fields. For example, suppose you have an entry form that looks similar to the one shown in Figure 48 on page 142.



ENTER ▶ *dept number*

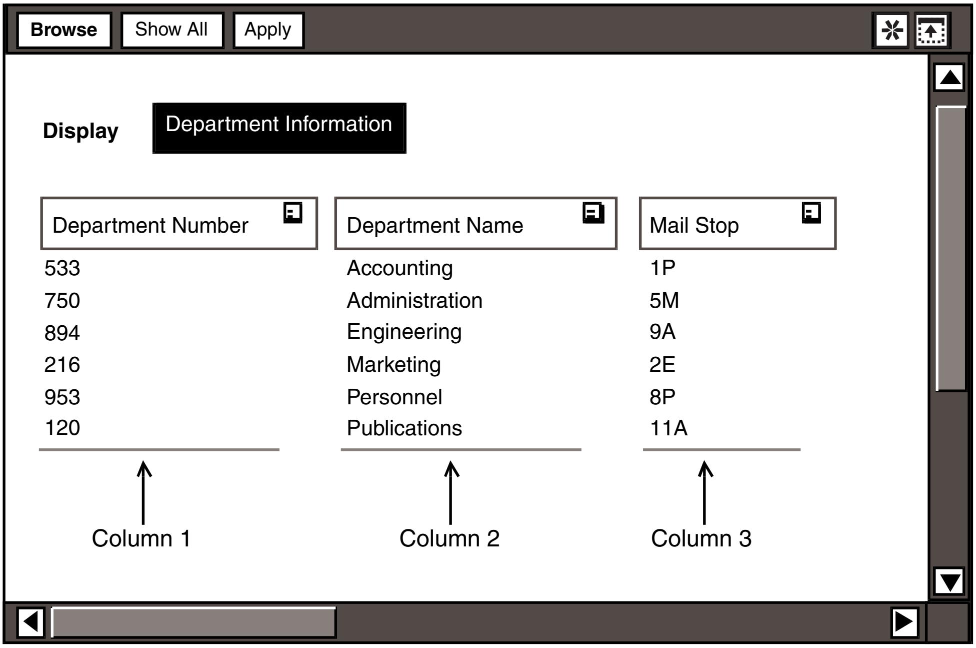
Department Number

Department Name

Mail Stop

Figure 48. An entry form

You can set up one browse list to accommodate each field within this form by creating a browse list with columns for the department number, department name, and mail stop, as shown in Figure 49 on page 142.



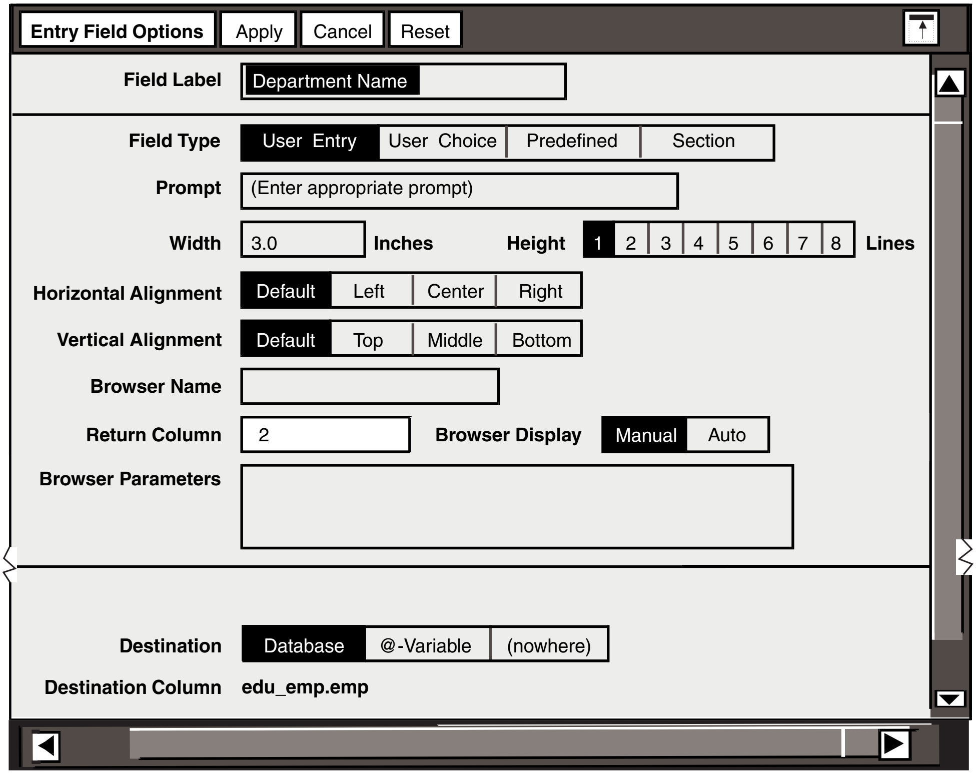
Display: Department Information

Department Number	Department Name	Mail Stop
533	Accounting	1P
750	Administration	5M
894	Engineering	9A
216	Marketing	2E
953	Personnel	8P
120	Publications	11A

Figure 49. Browse list

As shown in Figure 49 on page 142, Department Number is column 1, Department Name is column 2, and Mail Stop is column 3. To use this browse list for all three fields, you must specify the correct column to return information to the specific fields. Enter the column number in the `Return Column` field in the field options window.

For example, to specify that the item in the Department Name column is returned to the field, type `2` in the `Return Column` field as shown in Figure 50 on page 143.



Field Label: Department Name

Field Type: [✓ User Entry] User Choice Predefined Section

Prompt: (Enter appropriate prompt)

Width: 3.0 Inches Height: [✓ 1] 2 3 4 5 6 7 8 Lines

Horizontal Alignment: [✓ Default] Left Center Right

Vertical Alignment: [✓ Default] Top Middle Bottom

Browser Name:

Return Column: 2 Browser Display: [✓ Manual] Auto

Browser Parameters:

Destination: [✓ Database] @-Variable (nowhere)

Destination Column: edu_emp.emp

Figure 50. Field Options window

See “Chapter 12. Constructing Entry Fields,” on page 73 for more information on browse list options in the field options window.

Converting Existing Forms Design Browse Lists

When a Forms Design browse list from a previous DIS release is encountered, a new browse list is created that complies with Meta5 Version 4. The name of the browse list remains the same, but the `Return Column` field in the field options window is set to `1`, and the `Browser SQL` field is set to Local.

Returning Information from the Right Column

You can access all columns in the browse list by specifying the column number in the `Return Column` field. Thus, when converting an existing Forms Design browse list, the `Return Column` field needs to be set to `1` because the existing browse lists are already set up with the valid information in column one.

Understanding the Browser SQL Choice

The `Browser SQL` field is set to Local because the SQL text is taken from the old Forms Design browse list and placed in the `SQL Text` field. This method sends the SQL statements as they are, and the Data Entry tool is not affected.

Chapter 19. Enhancing Forms with Design Techniques

Placing fields side-by-side, separating areas of a form, and adding special instructions are several design techniques Forms Design provides. Use these techniques to design effective and efficient forms. This chapter describes how to:

- Create groups to place fields side-by-side
- Use lines to separate fields in a form
- Embed special instructions in a form

Grouping Fields

The group feature gives you greater flexibility when creating forms. Grouping fields together saves form space by enabling you to place fields, or groups of fields, side-by-side.

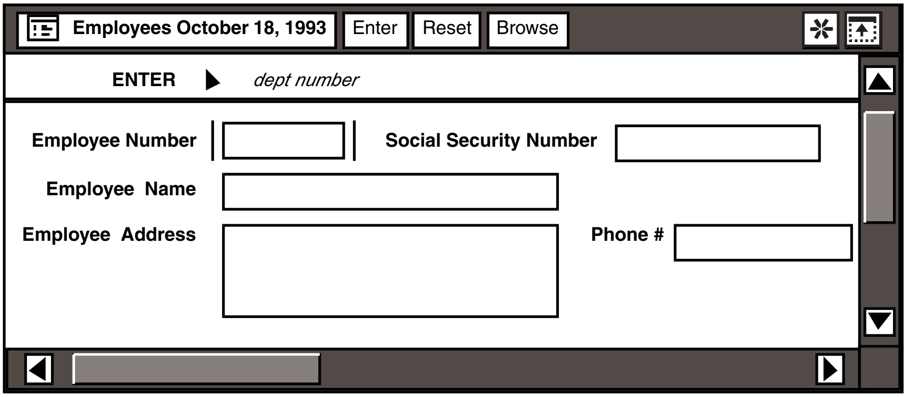


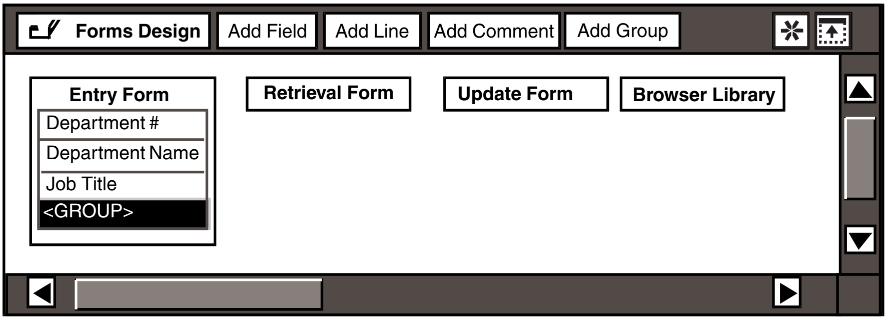
Figure 51. Fields placed side-by-side

In Figure 51, the Data Entry form saves space by placing the `Employee Number` and `Social Security Number` fields next to each other. These two fields are part of a group.

Creating Groups

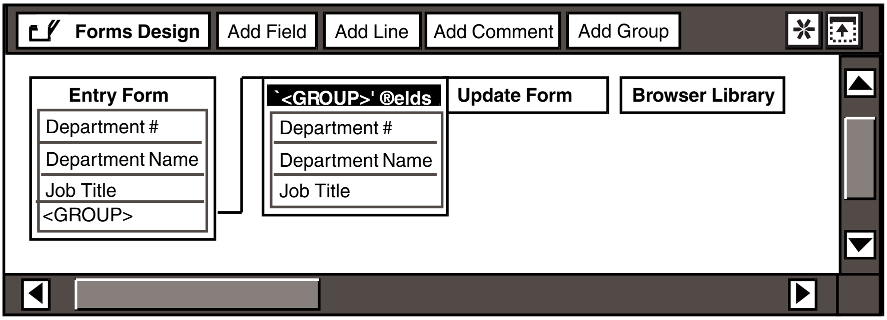
To create a group:

1. Click on the `Add Group` button in the main Forms Design window header. A field labeled `<GROUP>` is added to the last form you selected.



2. Select the highlighted `<GROUP>` field. A heading, **'<GROUP>' fields**, is displayed.
3. Select the **'<GROUP>' fields** heading. Add fields to the group by copying fields to the **'<GROUP>' fields** heading, or by clicking on the `Add Field` button in the main Forms Design window header.

   For example, copy the `Department #`, `Department Name`, and `Job Title` fields to the **'<GROUP>' fields** heading.



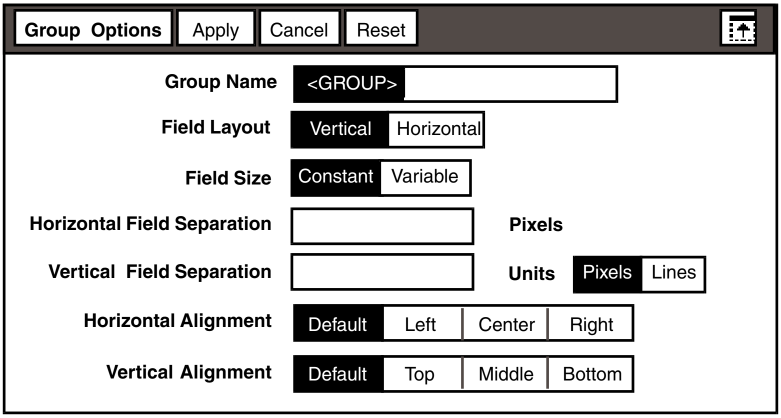
4. Double-click on the `<GROUP>` field to close it.

Designing Groups

Now that you have added fields to the group, you can set options for the group that determine how the fields are placed on the form.

To set options for a group:

1. Unless it is already highlighted, select the group field for which you want to set options.
2. Open the Group Options window. The Group Options window opens as shown in Figure 52 on page 147.



Group Name <GROUP>

Field Layout [✓ Vertical] Horizontal

Field Size [✓ Constant] Variable

Horizontal Field Separation □ Pixels

Vertical Field Separation □ Units [✓ Pixels] Lines

Horizontal Alignment [✓ Default] Left Center Right

Vertical Alignment [✓ Default] Top Middle Bottom

Figure 52. Group Options window

3. Change the options as needed.
4. Close the Group Options window.

With the group feature, you can create groups within groups. By using this technique, you can create several sub-groups within one group. Experiment with the functionality of groups to enhance your forms.

Table 23 on page 147 explains the options available within the Group Options window.

Table 23. Group field options

Field	Options	Function
Group Name	N/A	Enter the name of your group.
Field Layout	Vertical	Stack the fields within the group.
	Horizontal	Place the fields side-by-side within the group.
Field Size	Constant	Make each field space as big as the space occupied by the longest and widest fields.
	Variable	Make each field space differ according to the space needed by each field.
Horizontal Field Separation	N/A	Determine the relative amount of space between fields that are placed side-by-side.
Vertical Field Separation	N/A	Determine the relative amount of space between fields that are stacked.
Units	Pixels	Measure the vertical field separation in pixels.
	Lines	Measure the vertical field separation in lines.

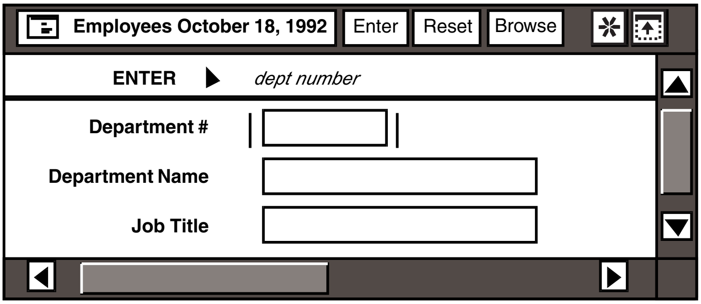
Table 23. Group field options

Field	Options	Function
Horizontal Alignment	Default	Align the group according to the system default.
	Left	Align the group flush left in its space.
	Center	Center the group horizontally in its space.
	Right	Align the group flush right in its space.
Vertical Alignment	Default	Align the group according to the system default.
	Top	Align the group at the top of its space.
	Middle	Center the group vertically in its space.
	Bottom	Align the group at the bottom of its space.

Placing Fields

You decide whether to stack the fields or place them side-by-side within a group.

If you define the `Field Layout` as `Vertical`, the fields are displayed stacked in the entry form as shown in Figure 53.



Department #

Department Name

Job Title

Figure 53. Stacked fields

If you select the `Horizontal` choice, the fields are displayed side-by-side. Figure 54 on page 149 shows fields placed horizontally within their group.

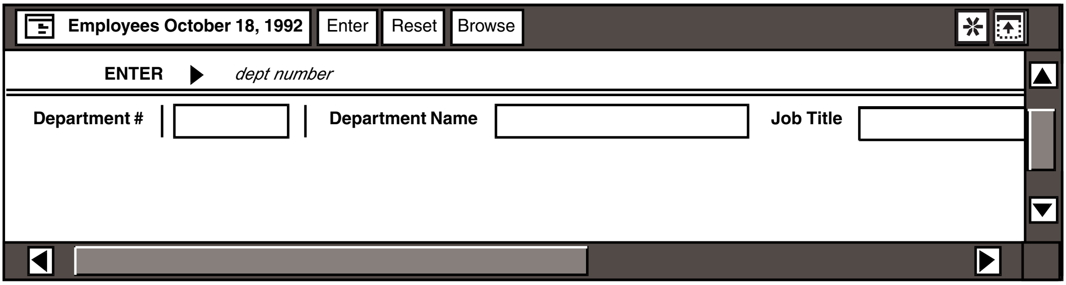


Figure 54. Fields placed side-by-side

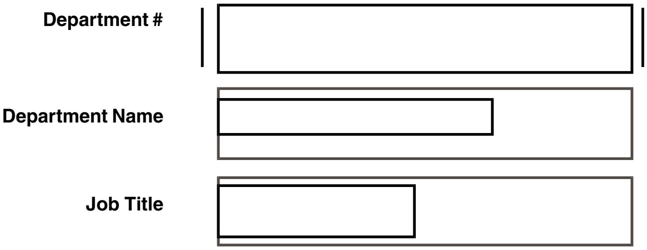
The Horizontal choice is especially useful if you have a form with many fields that have small lengths. Instead of scrolling down a very long form, you can condense information by grouping fields and specifying horizontal layout.

For an example of how you can design a form by placing fields horizontally or vertically, see “Chapter 21. Designing Complex Forms,” on page 171.

Setting the Field Size

You can specify the space occupied by the fields in your group. You can allocate space for fields according to the space occupied by the tallest and widest fields, or you can allocate space based on the needs of individual fields.

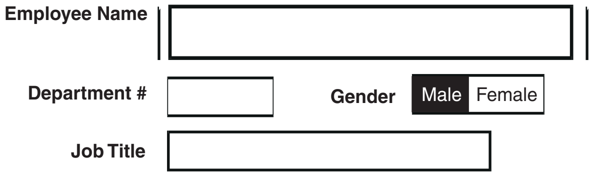
To allocate space based on the space occupied by the tallest and widest fields, select Constant in the Field Size field. The space taken by every field is the same, and is equal to the space occupied by the tallest and widest fields in the group. The next example shows a group in a form with a constant field size.



In the above illustration, the Department # field is the tallest and the widest field; therefore its size determines the space the rest of the fields in the group can occupy. The dotted lines represent the space the Department Name and Job Title fields could occupy.

If you want to align fields horizontally or vertically within a group, you must specify Constant in the Field Size field. See “Aligning Horizontally and Vertically” on page 76 for more information.

To vary the space allocated to each field, select `Variable` in the `Field Size` field. When you have many fields, selecting `Variable` helps save space; especially if you place the fields side-by-side. The space for each field is determined by the width and height required for the particular field. The next illustration shows a group with a variable field size.



Employee Name

Department #

Gender

Male Female

Job Title

Setting Field Separation

You can specify how much space separates fields within a group.

The `Horizontal Field Separation` field specifies the amount of space, in pixels, between fields that are placed side-by-side. The `Vertical Field Separation` field specifies the amount of space between fields that are stacked. You set the unit of measurement for vertical field separation in the `Units` field.

Aligning Horizontally and Vertically

You can specify the horizontal and vertical alignment of a group by selecting choices in the `Horizontal Alignment` field and `Vertical Alignment` field. See “Aligning Horizontally and Vertically” on page 76 for more information.

Using Lines to Separate Fields

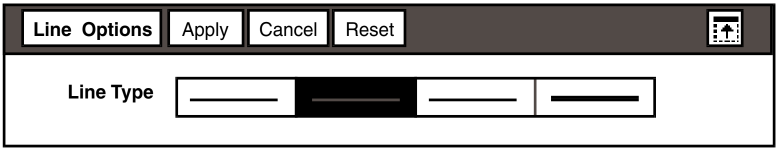
When you create a form with many fields, you can give the form structure with lines.

To separate fields using a line, click on the `Add Line` button in the window header. A `----LINE----` field is displayed at the bottom of the form.

If the line is in the wrong position, you can move it below the field that you want it to follow.

To change a line type:

1. Select the `----LINE----` field.
2. Open the Line Options window shown in Figure 55 on page 151.



Line Type

Figure 55. Line Options window

3. Select a choice from the `Line Type` field.

   By default, a line that you add with `Add Line` is the second choice in the `Line Type` field shown in Figure 55 on page 151. This choice is displayed as a faint solid line.

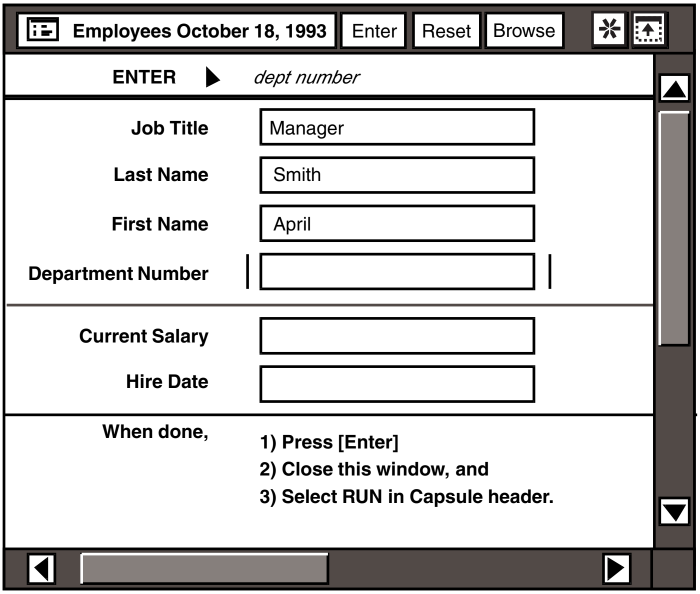
4. Close the Line Options window.

   If you want to add a blank space following the line you just added, use a comment field. You can create a comment field with the `Add Comment` button.

Embedding Special Instructions

There might be times when you want to embed special instructions in a form. For example, if you have a form that sets @-*variables*, you can include instructions to tell the user how to run the capsule application.

Figure 56 on page 151 shows an entry form that contains special instructions.



ENTER ▶ *dept number*

Job Title	Manager
Last Name	Smith
First Name	April
Department Number	
Current Salary	
Hire Date	

When done,

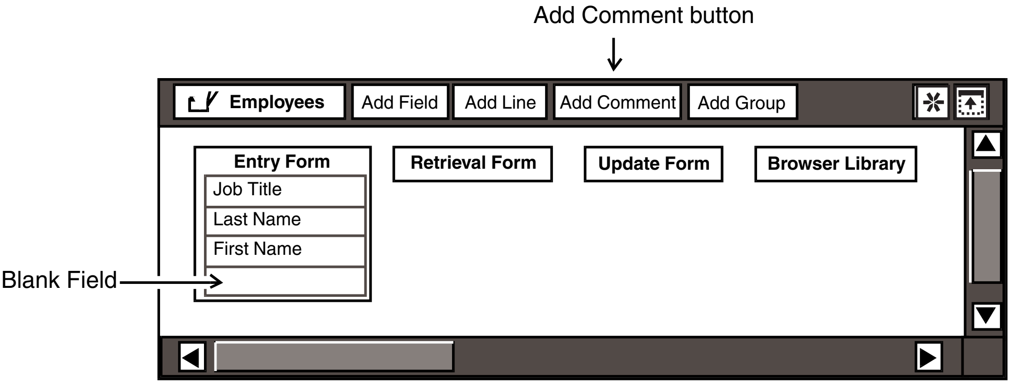
1) Press [Enter]
2) Close this window, and
3) Select RUN in Capsule header.

Figure 56. Special instructions embedded in a form

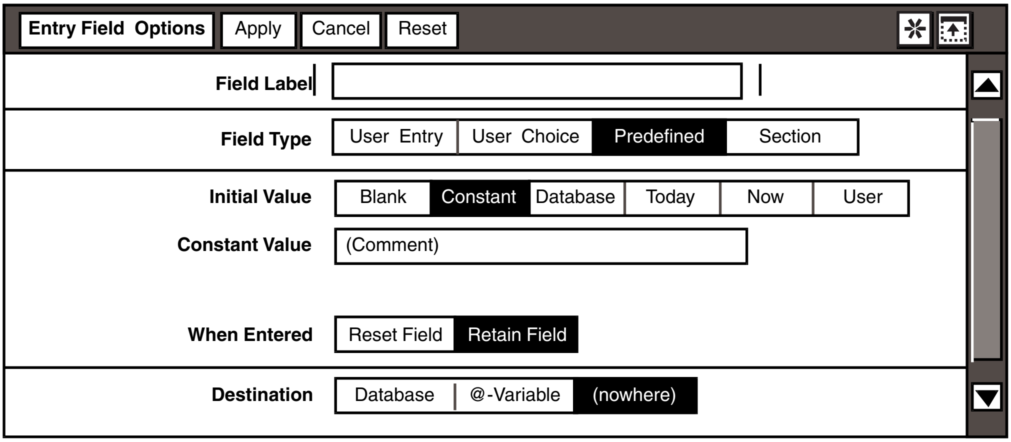
Each step in the instructions corresponds to a field in the Entry Form depiction in the main Forms Design window.

To add comments to a form:

1. Click on the `Add Comment` button. A blank field is displayed at the bottom of the form depiction.



2. Select the blank comment field, if it is not already highlighted.
3. Open the field options window.



Field Label

Field Type: User Entry / User Choice / [✓ Predefined] / Section

Initial Value: Blank / [✓ Constant] / Database / Today / Now / User

Constant Value: (Comment)

When Entered: Reset Field / [✓ Retain Field]

Destination: Database / @-Variable / [✓ (nowhere)]

4. Enter a label in the `Field Label` field if you want your instructions to align with the other fields in the form.

   For example, to create the instructions shown at the bottom of Figure 56 on page 151, `When done,` was entered into the `Field Label` field.

5. Enter an instruction into the `Constant Value` field.

   In this example, `1) Press Enter` was entered.

6. Select `Retain Field` in the `When Entered` field.

7. Select (nowhere) in the Destination field.
8. Close the field options window when complete.
9. Repeat these steps to add any additional comments or instructions.

   In this case, the Field Label field was left blank for the additional instructions so that all instructions are left aligned.

Chapter 20. Becoming an Expert Forms Designer

As you gain experience with the Forms Design tool, you might consider using its more technical features to set @-variables within a capsule application, or use @-keywords to determine the name of your Data Entry icon. This chapter describes the following tasks:

- Using primary key fields in forms
- Displaying database information for fields
- Creating a form to set @-variables
- Generating post-entry SQL
- Capturing SQL
- Specifying Forms Design @-keywords
- Enhancing fields in retrieval forms
- Modifying Forms Design tools with scripting
- Using column and table name indirection

General Form Considerations

There are several items you should consider when you initially design a form.

Using Primary Key Fields in Forms

Forms that allow a user to insert, delete, or update rows in a database table must be carefully designed to specify the primary key for the table. Otherwise, the form might not include the appropriate constraints, and the user could alter more rows than intended. The primary key fields need not be in the form itself; the values can be specified in post-entry SQL. See “Generating Post-Entry SQL” on page 159 for more information.

See “Generating the Choices Dynamically” on page 90 for information on primary keys.

Limiting the Number of Tables Each Form Represents

Although you can design an entry form with columns from more than one table, do so with caution. Each entry form should refer to columns from only one table unless you are inserting the data with post-entry SQL.

Retrieval and update forms only access one table. If you create a retrieval form or update form with columns from more than one table, the user receives an error message when trying to use the form.

If a data entry task involves multiple tables, you can:

- Create one Data Entry icon for each table, and place all the Data Entry icons in a folder for distribution.
- Use post-entry SQL to update multiple tables. See “Generating Post-Entry SQL” on page 159 for more information.

Using Views in Forms

Be very cautious when you design forms based on views. The view might contain columns from more than one table, or it might not be possible to use the view to insert or update data. See *Administering Databases with Meta5* for information on views.

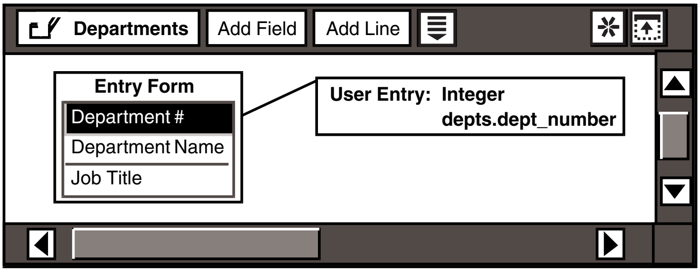
Displaying Database Information for Fields

You can display database information specific to each entry or predefined field, or for multiple fields within a form.

Displaying Information for Single Fields

To display database information for a single field (entry or predefined):

1. Double-click on an entry or predefined field in the Forms Design window. An information box is displayed as shown in Figure 57.



User Entry: Integer
depts.dept_number

Figure 57. Information box in the Forms Design window

The information box contains the field type, the acceptable data type, and the name of the database column to which the selected field corresponds. In this example, Department # is defined as a `User Entry` field, with an acceptable data type of Integer, and database column depts.dept_number.

You can obtain the same information by double-clicking on the desired column in the table shown in the category window.

2. Select the field again. The information box closes.

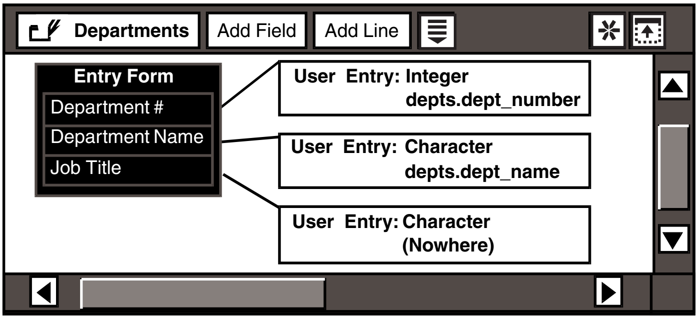
If you double-click on a choice or section field, the heading for choices or sections is displayed. See “Chapter 13. Designing Choice Fields,” on page 81 for information about adding choices to a field, and “Chapter 14. Building Flexibility with the Section Field,” on page 95 for information about adding sections to a section field.

Displaying Information for All Fields in a Form

To display database information for all fields in a form:

1. Double-click on the form heading in the main Forms Design window.

   For example, if you double-click on the heading of the Entry Form depiction, the information boxes are displayed for all the fields in that form.



2. Select the `Entry Form` heading again. The information boxes close.

Working with Capsule Applications

When you create a Data Entry icon for use inside a capsule application:

- You can automatically enter values into the `@-variable` fields in the User Input Control window of the capsule tool.
- You can verify that the Data Entry icon is set to store data in @-variables.
- You can create a Data Entry icon to automatically open as soon as a capsule window opens.

However you decide to use your Data Entry icon with your capsule application, it is a good idea to leave the Forms Design icon in the capsule icon along with the Data Entry icon. You can move the Forms Design icon way to the bottom or the right so that it is not visible. Leaving the Forms Design icon in the capsule icon makes it easy to update the Data Entry icon later.

The following sections describe these tasks in greater detail. For information about using a Data Entry icon inside a capsule application, see “Chapter 6. Using Data Entry in a Capsule Application,” on page 35.

Creating a Form to Set @-Variables

You can customize a Data Entry icon to set @-variables inside a capsule application. To do so, create a Data Entry icon with fields whose destinations are specific @-variables. The Data Entry icon then automatically enters the information into the `@-variable` fields in the capsule tool's User Input Control window.

To create a field with data stored in @-variables:

1. Select a field in the `Entry Form` depiction.
2. Open the Entry Field Options window.

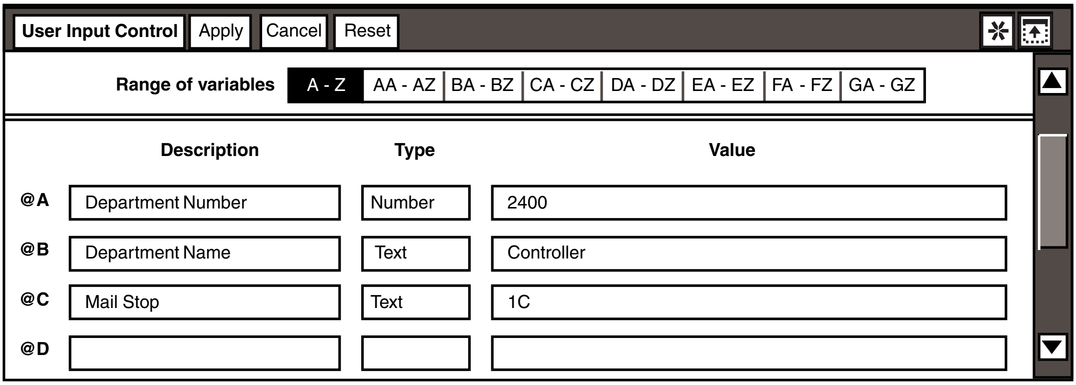
   The Entry Field Options window, similar to the one shown in Figure 18 on page 64, is displayed.
3. Select `@-Variable` in the `Destination` field at the bottom of the window. Because `@-Variable` is the default, it might already be selected.
4. Enter a value in the `@-Variable` field. For example, enter `@A`.
5. Close the Entry Field Options window.
6. Repeat this procedure for each field that you want to store in an @-variable, assigning a different @-variable for each field.

Verifying @-Variable Entries

Verify your @-variable declarations by testing with a Data Entry icon. Enter values into the Data Entry form, then check inside the User Input Control window to see if the values have been updated in the `@-variable` fields.

To open the Capsule tool's User Input Control window, click on the `Show Controls` button in the Capsule window header.

The User Input Control window is shown in Figure 58.



	Description	Type	Value
@A	Department Number	Number	2400
@B	Department Name	Text	Controller
@C	Mail Stop	Text	1C
@D			

Figure 58. User Input Control window

Opening a Data Entry Icon Automatically

You might want to ensure that the Data Entry icon opens before running the capsule application. To rename a Data Entry icon so that it opens automatically:

1. Select the Data Entry icon.
2. Open the Icon Options window.
3. Delete the old name, and type `Auto Open` in the `Name` field.
4. Close the Icon Options window and deselect the Auto Open icon before you run the capsule application.

The location of the Auto Open icon is not important. It can be placed anywhere inside the capsule window. However, only one Auto Open icon can automatically open in a capsule window. If you insert two such icons, the first one you inserted automatically opens when you open the capsule window.

Creating a Forms Design Icon from a Data Entry Icon

You might want to use the same Forms Design icon to create other similar Data Entry icons. Therefore, you should retain the Forms Design icon associated with each Data Entry icon that you create. However, if you inadvertently delete a Forms Design icon, there is a way to recreate it from its Data Entry icon. With the new Forms Design icon, you can:

- Replace the original Forms Design icon
- Redesign the Data Entry icon that you are presently using

  This procedure works only if you are logged on to the same desktop that you used to create the Data Entry icon.

To recreate a Forms Design icon:

1. Log on to the desktop where the Data Entry icon was created.
2. Open the Data Entry icon.
3. Select `Create Forms Design Icon` in the Data Entry window's `Special` menu. The pointer changes, and the following message is displayed on your desktop:

   ```
   Select location of new Forms Design icon.
   ```
4. Select an empty location on your desktop or in a container. The Forms Design icon is displayed on your desktop.

Generating Post-Entry SQL

Generate post-entry SQL when you want to run one or more SQL statements after each transaction. Post-entry SQL enables you to:

- Simultaneously insert or update data in more than one table

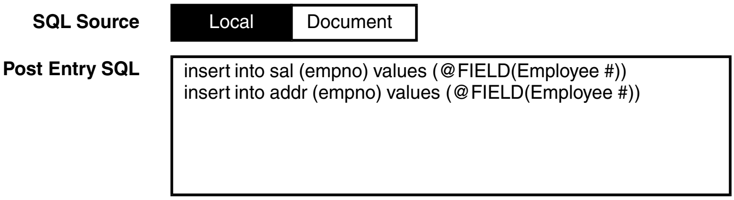
- Run an SQL program that exists in another document or in the Forms Design tool
- Produce an audit trail
- Construct constraints on data

You can specify SQL statements in either the Update Form Options windows. Because the SQL statement is associated with the form, you can specify different SQL commands for each form.

Choose from two methods of specifying post-entry SQL:

- Locally

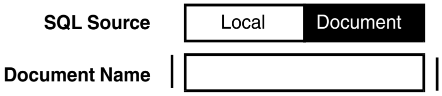
  Select `Local` in the `SQL Source` field and enter statements in the `Post-Entry SQL` field.



```
insert into sal (empno) values (@FIELD(Employee #))
insert into addr (empno) values (@FIELD(Employee #))
```

- Remotely

  Select `Document` in the `SQL Source` field and name the location of the statements in the `Document Name` field.



If you are entering the SQL statements locally, be sure to separate each command line by pressing Shift+Enter on your keyboard. This procedure ensures that all the commands are run before access of the tables is allowed.

If you are inserting or updating data in multiple tables, see “Inserting or Updating Data in More Than One Table” on page 161.

If you are entering the SQL statements remotely, see “Accessing SQL Statements in a Source Document” on page 161 for more information.

As with the SQL Entry tool, you can incorporate flow control and conditional processing by including &statements, functions, and function statements in the post-entry SQL. See “Appendix A. Identifying Keywords and Statements,” on page 187 for a list of the &statements and functions that are supported by the Forms Design tool. See “Constructing an SQL Constraint” on page 163 for information about the &CONSTRAINT function.

Inserting or Updating Data in More Than One Table

You can insert or update data in more than one table by specifying post-entry SQL statements. You write these statements in the `Post-Entry SQL` field of the form options window, or refer to the document that contains the SQL statements.

For example, you can write statements that insert rows into the Salary and Name and Address tables for a specific employee. The statements are similar to the following examples:

```
insert into sal (empno) values (@FIELD(Employee#))
insert into addr (empno) values (@FIELD(Employee #))
```

The @FIELD keywords in the above example reference the `Employee #` field in the Entry form.

Set the destination of the column to `None` or to `@-Variables`.

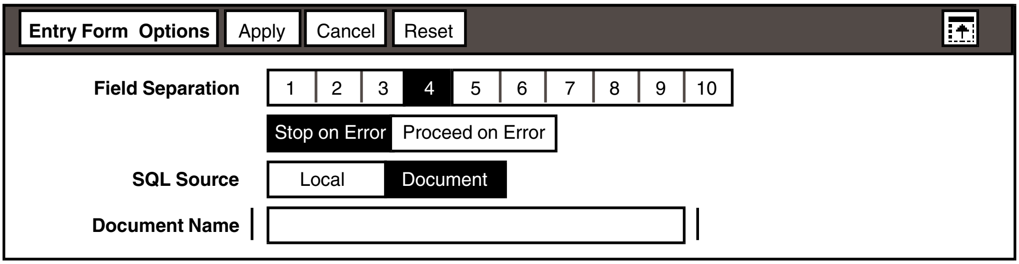
See “Using @-Keywords” on page 187 for more information.

Accessing SQL Statements in a Source Document

You can use existing SQL statements in a source document rather than enter SQL statements into the `Post-Entry SQL` field.

To retrieve SQL statements from a source document:

1. Select `Document` in the `SQL Source` field. The `Document Name` field is displayed as shown in the following illustration.



Field Separation: 4

Stop on Error

SQL Source: Document

Document Name:

2. Enter the name of the document that contains the SQL statements.

The following methods are used to locate a source document:

- If the form is in a container, such as in a capsule application, the container is searched. If the source document is not found within the container, then the desktop is searched. If still not found, then System|File Drawers is searched.
- If the form resides on the desktop, the desktop is searched. If the source document is not found, then System|File Drawers is searched.

The Data Entry tool can locate a source document that is placed in a folder or a file drawer if a path name is included in the Entry Form Options window. This method is useful if your source documents are filed in a central location for easy maintenance.

Make sure that any file drawer you specify in a path name is on the system disk of your file server. For example, if you specify SQL Script #1 as your source document, and your source documents are filed in the SQL Scripts folder within the Data Access Libraries file drawer, enter `Data Access Libraries|SQL Scripts|SQL Script #1` as a path name in the `Document Name` field. These containers are then searched for the source document. Use the broken vertical bar key on your keyboard to produce the vertical bar in the path name.

If the source document cannot be found, the following error message is returned:

```
Cannot find the Post-Entry SQL source document.
```

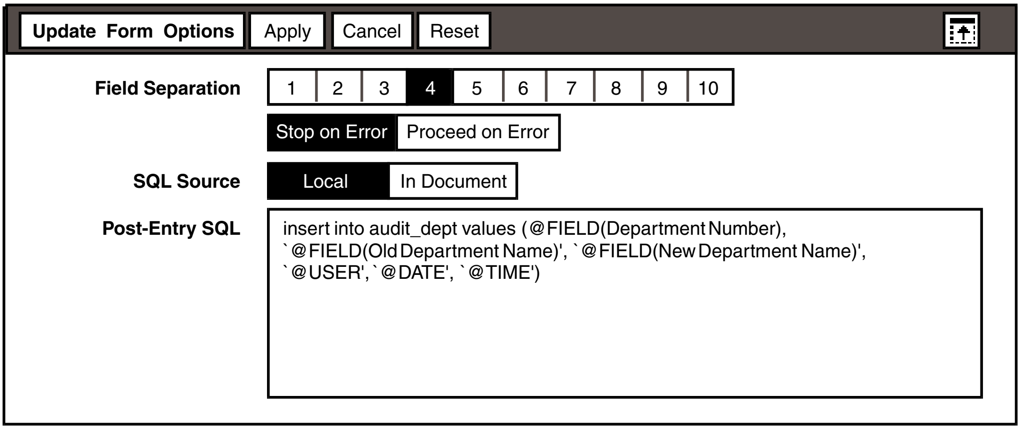
See the *Data Access Tools User's Guide* for more information on using a source document.

Producing an Audit Trail

You can produce an audit trail through the `Post-Entry SQL` field if you have defined the proper database tables. The audit trail lets you keep track of database activities.

To produce an audit trail:

1. Select the heading of a form in the main Forms Design window. For example, select the `Update Form` depiction.
2. Open the Update Form Options window.



Field Separation 1 2 3 [✓ 4] 5 6 7 8 9 10

[✓ Stop on Error] Proceed on Error

SQL Source [✓ Local] In Document

Post-Entry SQL

```
insert into audit_dept values (@FIELD(Department Number),
`@FIELD(Old Department Name)', `@FIELD(New Department Name)',
`@USER',`@DATE', `@TIME')
```

Figure 59. Update Form Options window

Figure 59 shows the post-entry SQL associated with the Update Form depiction. The Update Form is being used to change the department name in a Departments table. The post-entry SQL statements insert a row into an audit table that keeps track of the changes made to the Departments table.

See the *Data Access Tools User's Guide* for information on SQL statements.

Constructing an SQL Constraint

You can use the &CONSTRAINT function to construct an SQL constraint for a field. By using &CONSTRAINT, you can create an SQL statement that:

- Allows for multiple data items in a field – for example, 0, 1, or more.
- Specifies the type of column to be constrained.
- Combines &CONSTRAINT( ) with the rest of the SQL statement by using terms such as WHERE, AND, IN, and =.

The SQL statement must be in the following format when using the &CONSTRAINT function:

`select` *columnname1* `from` *tablename* `&CONSTRAINT(`*columnname2, datatype, operatortype, datavalues*`)`

where:

columnname1 is the name of the database column that contains the values used for validating the field.

tablename is the name of the database table that contains the values used for validating the field.

columnname2 is the name of the database column that contains the result.

datatype specifies the type of data to be constrained. The data type must be in uppercase. Valid types include:

CHAR¾a value in a character format.

NUMBER¾a value in a numeric format.

DATE¾a value converted to the appropriate server date.

operatortype is either EQUAL or NOTEQUAL.

datavalues are the single or multiple values to be used as the constraint values. Quotes are not used for text.

Table 24 lists three examples of SQL statements that use the &CONSTRAINT function. The examples are based on the following SQL statement:

```
select Sales from Salestable &CONSTRAINT(City, CHAR, EQUAL,
@FIELD(City))
```

Table 24. Examples using &CONSTRAINT in SQL statements

If the value is:	**The SQL statement is:**
City=NY	Select Sales from Salestable where city = 'NY'
City=NY, CA, GA	Select Sales from Salestable where city in ('NY', 'CA', 'GA')
No values in field	Select Sales from Salestable

Be sure to include any additional clauses before the &CONSTRAINT to prevent parsing errors.

See “Appendix A. Identifying Keywords and Statements,” on page 187 for information about other available &functions.

Capturing SQL

The forms designer can view and store the SQL activity generated when the Data Entry icon is used. Capturing the SQL activity is especially useful for troubleshooting problems in the Data Entry icon.

The Data Entry SQL Log window, discussed in this chapter, and the Data Entry Log window that open when you select `Generate Log` in the `Special` menu are different. The Data Entry SQL Log window records the actual SQL statements used to change the database tables currently being used in any form. The Data Entry Log window only records actual data entered in an entry form or update form. See “Recording Your Entries” on page 24 for more information on the Data Entry Log window.

To capture the SQL activity:

1. Select `Capture SQL` in the `Special` menu of the window header.

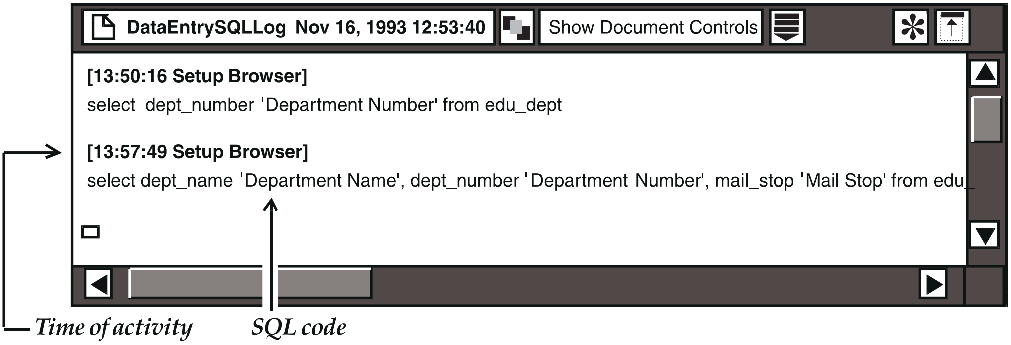
   The Data Entry SQL Log icon is displayed on your desktop and opens.

   When the Data Entry SQL Log icon opens, the date and time are displayed in the window header. The date and time in each icon name allow you to differentiate between logs.

   You must leave this window open to record the SQL activity. If you close the window, you have to repeat step 2 to begin capturing SQL activity again.

   You now have two windows open on your desktop, the Data Entry window and the Data Entry SQL Log window. The Data Entry SQL Log window captures all SQL activity generated in the entry form.

2. Enter information into the Data Entry form. The Data Entry SQL Log window, shown in Figure 60, reflects your entries.



```
[13:50:16 Setup Browser]
select  dept_number 'Department Number' from edu_dept

[13:57:49 Setup Browser]
select dept_name 'Department Name', dept_number 'Department Number', mail_stop 'Mail Stop' from edu_
```

Figure 60. SQL record

The SQL code in the Data Entry SQL Log window is preceded by the time of the activity, and each SQL activity is recorded in a separate block of text. The SQL code generated from the browse list and captured in the Data Entry SQL Log window is the same code that you entered when you created each field in the Forms Design icon.

3. Close the Data Entry SQL Log window when you have finished using the entry form. This icon now contains a record of all SQL activity for that specific time period.

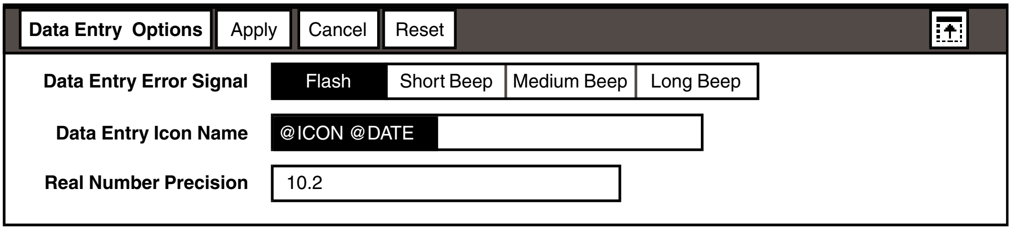
Specifying Forms Design @-Keywords

You can enter @-keywords in a field to generate system data, such as the current date, or in reference fields within Data Entry forms.

For example, when you create a Data Entry icon from a Forms Design icon, the name of that Data Entry icon is the same as the Forms Design icon, and also includes the current date. The @-keywords @ICON and @DATE were specified in the Forms Design icon so that the Data Entry icon is named in this manner.

To assign @-keywords to name a Data Entry icon:

1. Open the Forms Design icon.
2. Select `Show Data Entry Options` in the `Special` menu. The Data Entry Options window opens as shown in Figure 61 on page 165.



Data Entry Error Signal: [✓ Flash] Short Beep Medium Beep Long Beep

Data Entry Icon Name: @ICON @DATE

Real Number Precision: 10.2

Figure 61. Data Entry Options window with @-keywords

3. Enter @-keywords in the `Data Entry Icon Name` field.

   In this example, @ICON and @DATE are displayed in the `Data Entry Icon Name` field. When a Data Entry icon is first created, it automatically assumes the Forms Design icon name and current date.

4. Close the Data Entry Options window when complete.

The procedure is the same for entering any @-keyword into a field. The only @-keyword that has more than one function is @FIELD when it is used with the GLOBAL feature. See “Assigning a Global Name” for information about global names.

Assigning a Global Name

The GLOBAL feature assigns a global name to a field so that it can be referenced from any other field in a form. This feature is particularly helpful if there are many dependent fields and duplicate field labels in a form. Refer to the global name with the @FIELD @-keyword in the `Post-Entry SQL` field of the Update Form Options windows. The field with the global name does not have to be visible in the Data Entry window when post-entry SQL is executed.

For example, if you name a field `Product Name GLOBAL:Prod12`, the global name is `Prod12`. You can then refer to that name in the `Post-Entry SQL` field with `@FIELD(GLOBAL:Prod12)`. Although you see the entire field name in the Forms Design icon, only Product Name is displayed in the Data Entry icon.

Setting Up Conditional Expressions with @PARMs

When working with browse lists, you can access different parts of the browse list from different fields. The @-keywords @PARM1, @PARM2, through @PARM10 allow you to set up a constraint without specifying the conditions in the `General Qualifier` field. The specific conditions for each field are then given in the `Browser Parameters` field of the field options window. The position of the condition in the `Browser Parameters` field relates to the number of the @PARM. All fields that use this browse list must follow the constraint pattern given in the `General Qualifier` field.

For example, the expression `where @PARM1 = @PARM2` when given in the `General Qualifier` field can be filled by “City, Atlanta” in the `Browser Parameters` field. This expression, when compared to `where @PARM1 = @PARM2`, translates to City=Atlanta. The `@PARM` relationship is further explored in Table 25.

Table 25. @PARM relationships

General Qualifier field of the Browser Options window	Browser Parameters field of the field options windows	Resulting SQL
where @PARM 1 = @PARM 2	City, 'Atlanta'	City = 'Atlanta'

Table 25. @PARM relationships

General Qualifier field of the Browser Options window	Browser Parameters field of the field options windows	Resulting SQL
where @PARM 3 < @PARM 4	City, 'Atlanta', Employee #, 2365	Employee # < 2365
where @PARM 5 > @PARM 6	City, `Atlanta', Employee #, 2365, JOB CODE, 9011	JOB CODE > 9011

You can use the @PARMs in either the `General Qualifier` field, the `SQL Text` field in the Browser Options window, or the `Formula` field in the Browser Options window. The expression entered in any of these fields finds the values in the `Browser Parameters` field in the field options windows.

Modifying Forms Design Tools with Scripting

Sometimes it is necessary to modify information within a Forms Design tool. One way to change the information is to find the appropriate field and make a different selection. But what if you need to change something that is displayed throughout the Forms Design tool – for example, the name of a field?

With scripting, you use a Text tool to find and replace the information that needs to be modified. Scripting works by connecting the Forms Design icon to a Text icon within a capsule window. Figure 62 on page 167 shows the connection necessary to create a script from a Forms Design icon.

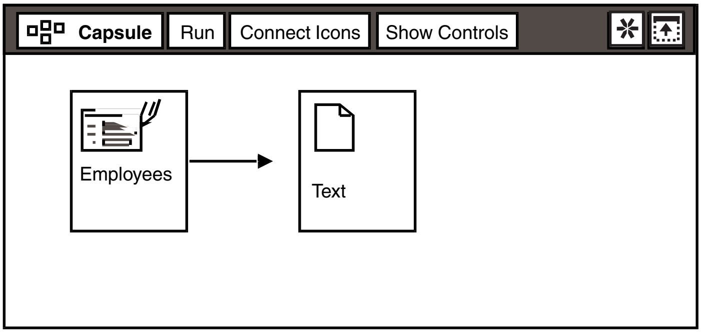
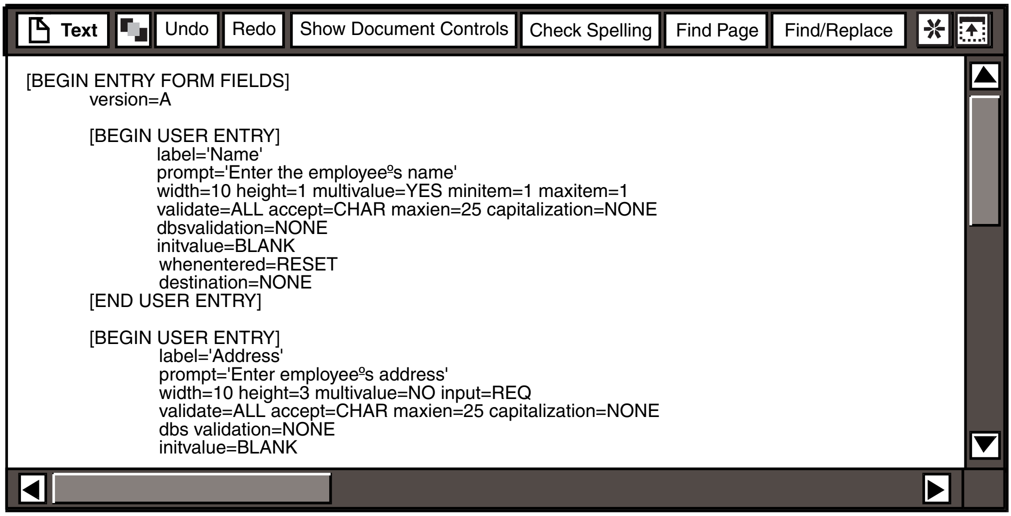


Figure 62. Connecting icons in a capsule application to create a script

As shown in Figure 62 on page 167, the Forms Design tool acts as the source of information, and the Text tool as the destination. When the capsule application runs, the Text tool is displayed as shown in Figure 63 on page 168.



```
[BEGIN ENTRY FORM FIELDS]
        version=A

        [BEGIN USER ENTRY]
                label='Name'
                prompt='Enter the employeeºs name'
                width=10 height=1 multivalue=YES minitem=1 maxitem=1
                validate=ALL accept=CHAR maxien=25 capitalization=NONE
                dbsvalidation=NONE
                initvalue=BLANK
                whenentered=RESET
                destination=NONE
        [END USER ENTRY]

        [BEGIN USER ENTRY]
                label='Address'
                prompt='Enter employeeºs address'
                width=10 height=3 multivalue=NO input=REQ
                validate=ALL accept=CHAR maxien=25 capitalization=NONE
                dbs validation=NONE
                initvalue=BLANK
```

Figure 63. A Text tool containing a script

Make any necessary changes to the script within the Text tool.

Before you make any changes to the Text tool, see “Knowing the Acceptable Values for Scripting Keywords” on page 189.

If a form is large and you do not want to create one document, you can create documents for different portions of the Forms Design icon by defining a region name in the arrow that connects the Forms Design and Text icons. Use the names of the forms for regions – for example, set the source region to entry form or entry. By using this method, you can break down the Forms Design script into three Text icons: one each for the entry, retrieval, and update forms.

When you have made all necessary changes, you can create a Forms Design tool from your modified Text tool. Place the modified Text tool into a Capsule window and connect it to a blank Forms Design icon as shown below.

Attention:

> Do not connect the Text tool to your original Forms Design tool. This process clears all existing fields and definitions from the Forms Design tool, and then constructs a new Forms Design tool. If an error is made within the Text tool, you will have lost the original, working Forms Design tool.

For information about the Text tool, see the *Text User's Guide*.

Using Column and Table Name Indirection

You can direct a Forms Design tool to another table without changing all of the occurrences of the table name in the icon by using the WTDD. This feature saves time and reduces error by enabling you to make the needed change in the WTDD, rather than in the Forms Design tool.

For example, use indirection if you have created a Forms Design icon for one department, and another department wants to use it for their databases. Instead of updating all of the occurrences of the table name, use the DBA tool to change the internal names in the WTDD so they match the internal names in the original table (check first to be sure that other applications are not using the same table). You can then configure the Forms Design icon to the new category in the Icon Options window, and it will reference the correct physical table and columns.

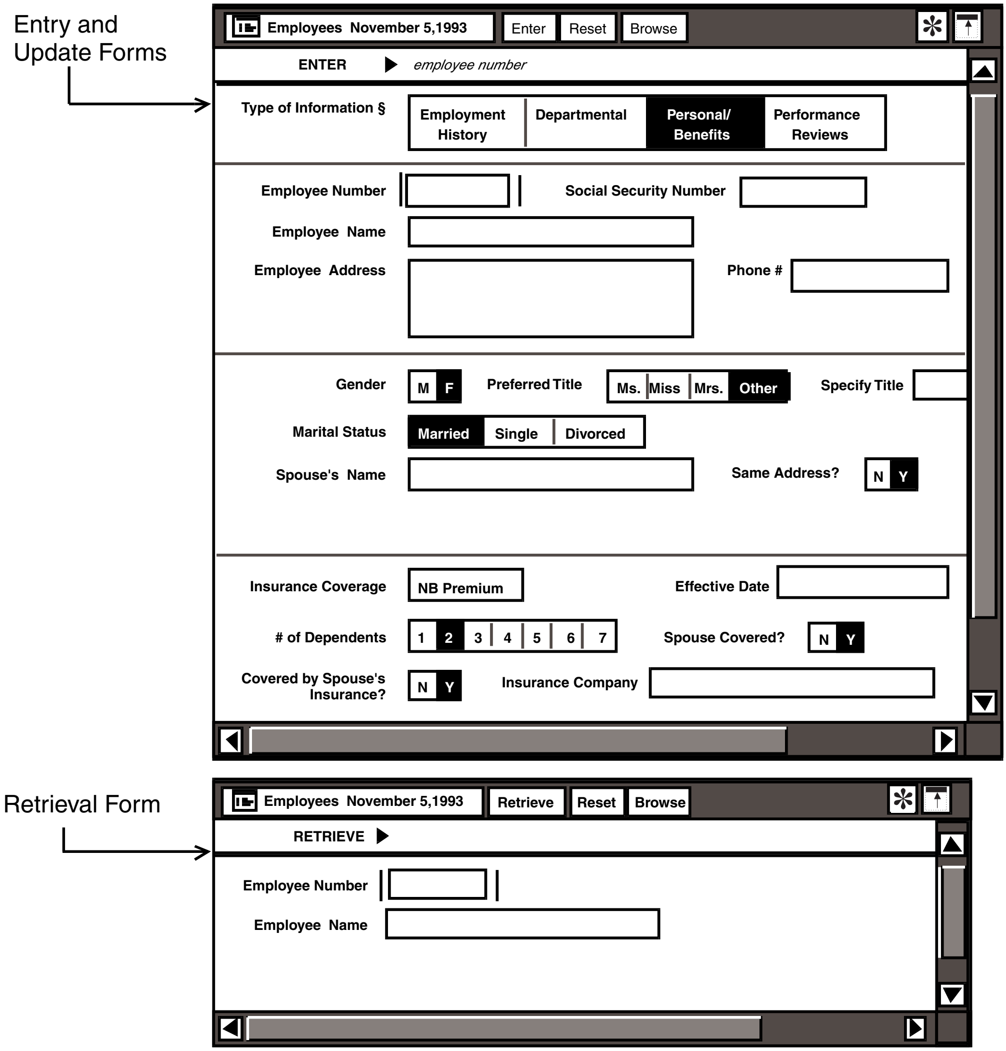
Also use indirection when you migrate tables from one database to another; for example, DB2 to Sybase**. Copy the WTDD to the new database, change the SQL names for tables and columns, and then make all icons reference the new database.

See *Administering Databases with Meta5* for more information about changing physical names in the WTDD.

Chapter 21. Designing Complex Forms

Now you have learned all of the individual benefits of Forms Design's features. But how are all of these features put together to create a form? This chapter provides a fairly complex form, and explains how certain aspects of the design were accomplished. References to more detailed information in the guide are also provided.

Suppose you are requested to create a Data Entry icon for Personnel to enter employee information into their databases. Because they want to update information later, you must create all three forms: entry, retrieval and update.


Entry and Update Forms

ENTER ▶ *employee number*

Type of Information § Employment History | Departmental | [✓ Personal/ Benefits] | Performance Reviews

Employee Number □ Social Security Number □

Employee Name □

Employee Address □ Phone # □

Gender M [✓ F] Preferred Title Ms. | Miss | Mrs. | [✓ Other] Specify Title □

Marital Status [✓ Married] | Single | Divorced

Spouse's Name □ Same Address? N [✓ Y]

Insurance Coverage NB Premium Effective Date □

# of Dependents 1 | [✓ 2] | 3 | 4 | 5 | 6 | 7 Spouse Covered? N [✓ Y]

Covered by Spouse's Insurance? N [✓ Y] Insurance Company □

Retrieval Form

RETRIEVE ▶

Employee Number □

Employee Name □

Figure 64. Entry, Update, and Retrieval forms

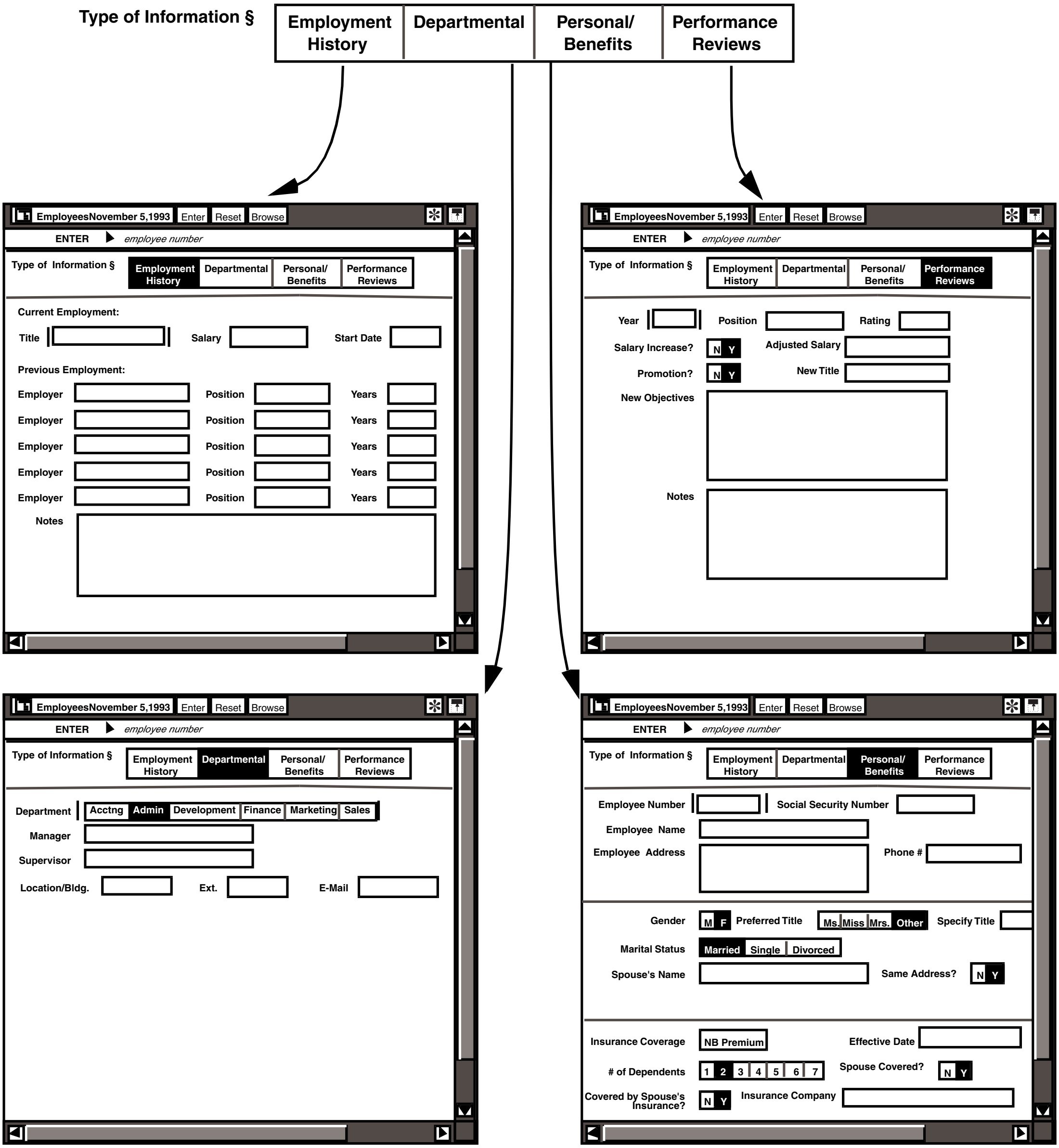
Figure 64 on page 172 only shows one section of a section field for the entry and update forms, but the logical structure of the forms is apparent.

While the entry and update forms contain all related fields, it is only necessary for the retrieval form to contain fields for the employee number and name. All of the associated information will be retrieved so that it can be updated or deleted.

The rest of this chapter focuses on the structure of the sections of the `Type of Information` field for the entry and update forms, and the contents of the retrieval form.

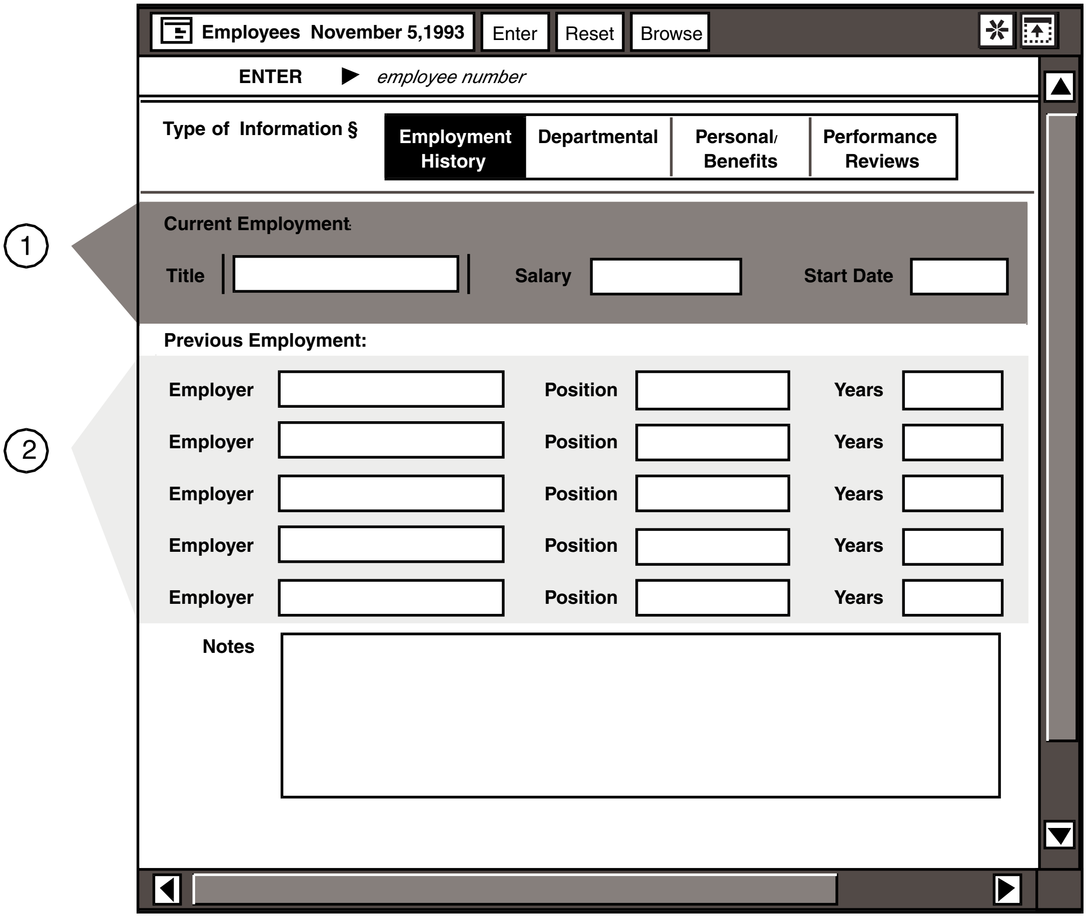
Structuring the Entry and Update Forms

The overall structure is created using one feature: the section field.



By using the section field, you can divide information into conceptual segments, and display it in different screens. The user enters information into all fields dependent upon a section field, and then validates the entries all at once by clicking on the `Enter` button. However, unless you define it that way, the user does not have to enter information into all the dependent fields.

Looking at the Employment History Section



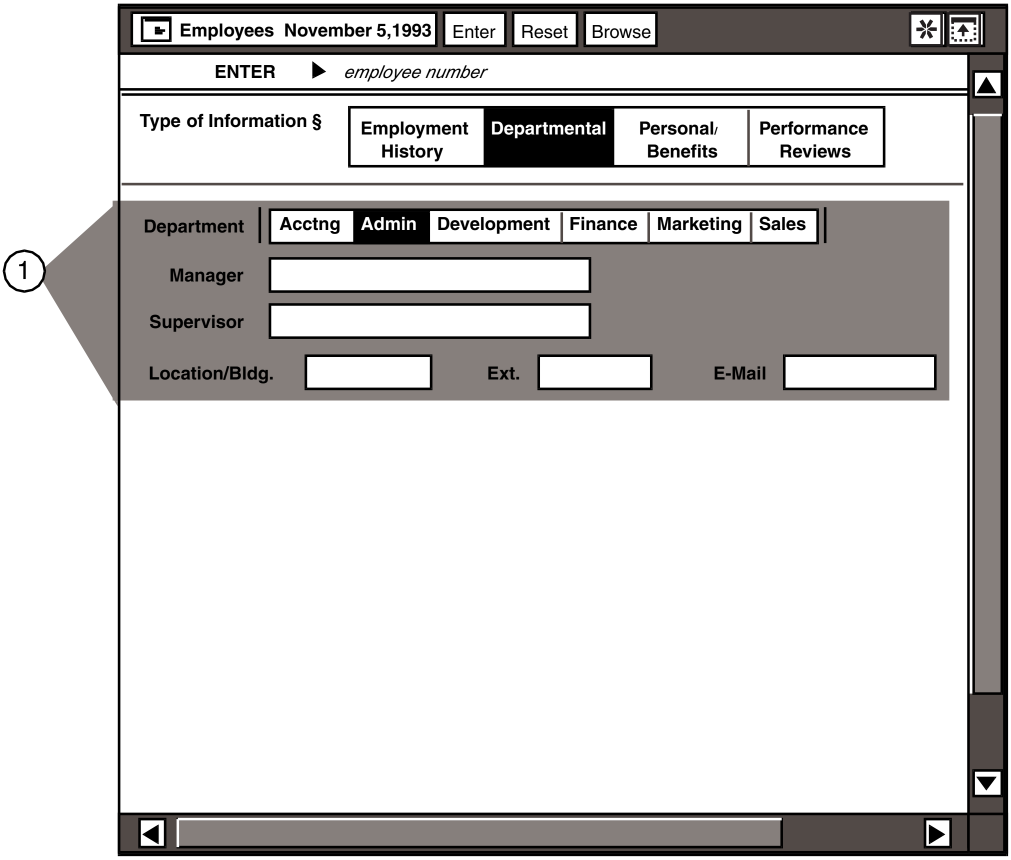
- Placing fields side-by-side helps to use space effectively. The `Title`, `Salary` and `Start Date` fields are part of a group in which the field layout is horizontal and the field size variable. These options are set in the Group Options window. See “Grouping Fields” on page 145.
- Making fields align horizontally and vertically gives your form an organized appearance. To align fields as shown in the Previous Employment section of the form, you must use the grouping feature.

  To align the fields as shown in the Previous Employment section of the form, you must create four groups: one group for each of the columns, and then one group to group all the columns together.

When grouping a column's fields, specify vertical alignment so that the fields align in a column. When grouping the columns together, specify horizontal alignment so that the column groups align by rows.

You can also use this method to align fields of differing lengths.

Browsing Through the Departmental Section

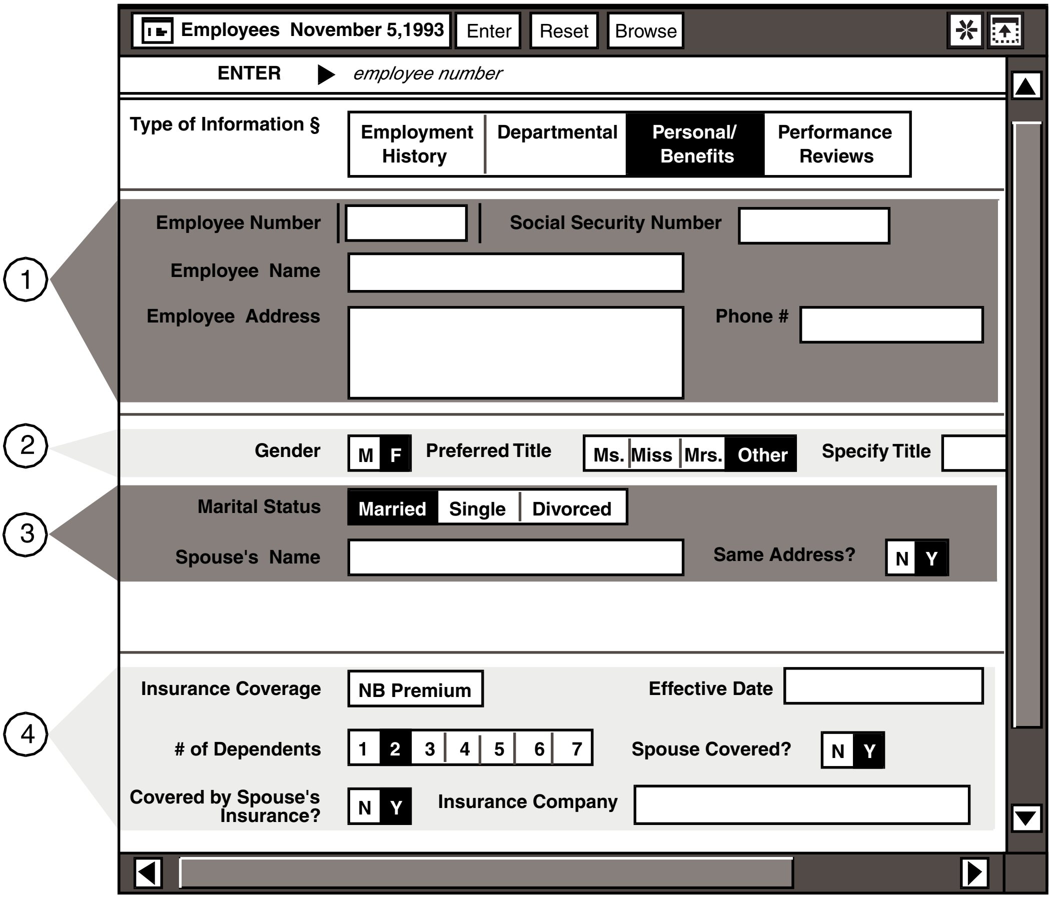


You can create a browse list to locate the name of the employee's manager and supervisor, and to find the employee's location in the company. To do this, create one browse list with these columns: Manager, Supervisor, Location/Bldg., Ext., and E-Mail. Then set the `Return Column` in the Entry Field Options window to the correct column. See “Chapter 18. Working with the Browse Feature,” on page 129 for information about constructing browse lists and setting the `Return Column` field for each column.

You can make the browse list even more efficient by using the `Department` field to constrain the information retrieved into your browse list while leaving the columns the same. For example, if `Marketing` is chosen from the `Department` field, then the browse list only returns information for the Marketing department.

Enter a constraint into the Browser Options window to constrain by department. To enter this constraint, select `Custom` from the `Browser SQL` field, and type your constraint in the `General Qualifier` field. For example, to constrain the information in the browse list by the department chosen in the `Departments` field, type `'DEPT='@FIELD(Department)'` in the `General Qualifier` field. See “Determining the Browser SQL” on page 133 for more information.

Exploring the Personal/Benefits Section

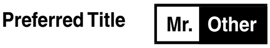


1. As shown in the first example, placing fields side-by-side helps to use space effectively. The fields in this section are part of groups in which the field layout is horizontal and the field size variable. These options are set within the Group Options window. See “Grouping Fields” on page 145 for more information.

   Be sure to allow enough space in fields for the required information to be entered. For example, the `Phone #` field is 1.5 inches wide so that a full phone number with area code can be entered. Also, use patterns when necessary as in the Social Security Number field.

To ensure that an employee number is not duplicated, validate the Employee Number field against the database. See “Preventing Duplicated Data” on page 112 for more information.

2. Again, fields are placed side-by-side, but the structure here is created by setting up dependencies between the fields. The selections within the Preferred Title field are dependent upon the selection of F in the Gender field. For a male, the Preferred Title field might be displayed as in the following illustration.

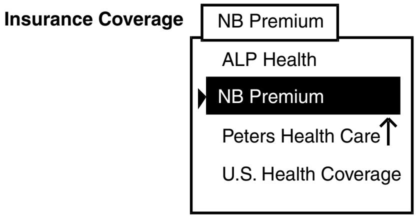
   Furthermore, the Preferred Title field enables the user to enter a title not given as a choice by providing the Other choice. The Specify Title field accommodates those choices of which the designer was not aware.

   The structure and placement of these fields is not accomplished by grouping, but is achieved using choice fields and dependent fields. The Preferred Title field is actually two different fields: one with choices for a female, another with choices for a male. The Specify Title field is also two different fields. The dependent fields of both the Gender and Preferred Title fields are placed to the right of the fields from within the field options window. See “Layering with Choice Fields” on page 84, “Creating Dependent Fields” on page 85, and “Positioning Dependent Fields” on page 93 for more information.

3. Eliminate work for the user by providing only those fields that are necessary. For example, the Spouse's Name and Same Address? fields are displayed only because Married is selected from the Marital Status field. Furthermore, an address field for the spouse is not displayed on the form unless N is selected from the Same Address? field because it is not necessary. Create this structure by using choice fields and defining dependent fields.

   Along the same lines, the entire section called out by point 4 can be made dependent upon the selection in the Marital Status field. For example, a married employee needs to be asked if her spouse is covered by insurance, but a single person does not. This format requires more fine-tuning on the part of the forms designer, but is rewarding for the user. See “Layering with Choice Fields” on page 84 and “Creating Dependent Fields” on page 85 for more information.

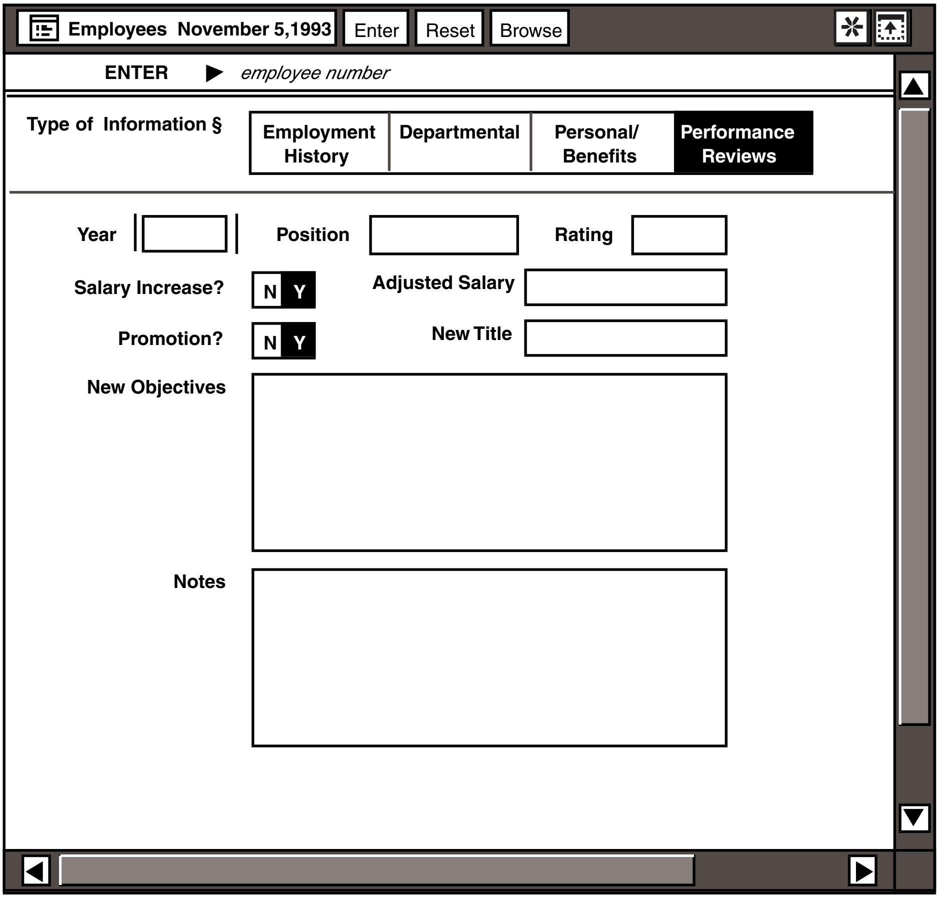
4. Place choices in a pull-down menu when there are many choices that take up too much space to display. An example is the `Insurance Coverage` field. Take a look at what that pull-down menu holds:



Keeping those choices in a pull-down menu saves a lot of form space. See “Presenting Choices” on page 88 for more information.

Reviewing with the Performance Review Section

This example uses several of the features described earlier in this chapter.



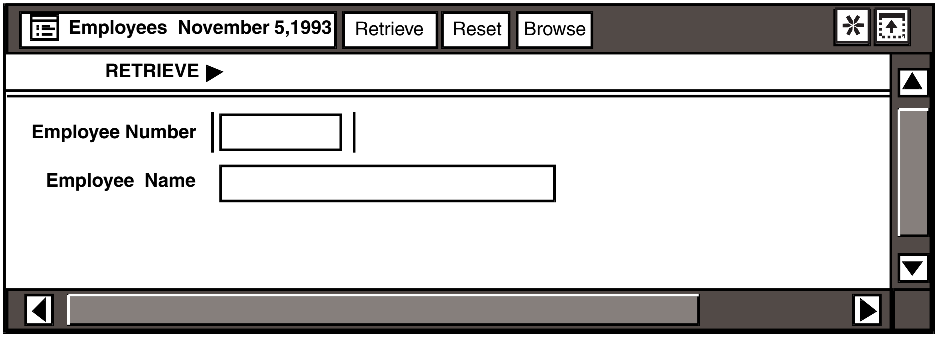
The features used in this example include:

- Placing fields side-by-side through grouping
- Creating hierarchies of fields through choices and dependent fields
- Structuring information so that fields are displayed only when necessary; for example, as a result of a choice
- Aligning fields horizontally and vertically through grouping
- Using one browse list for several fields

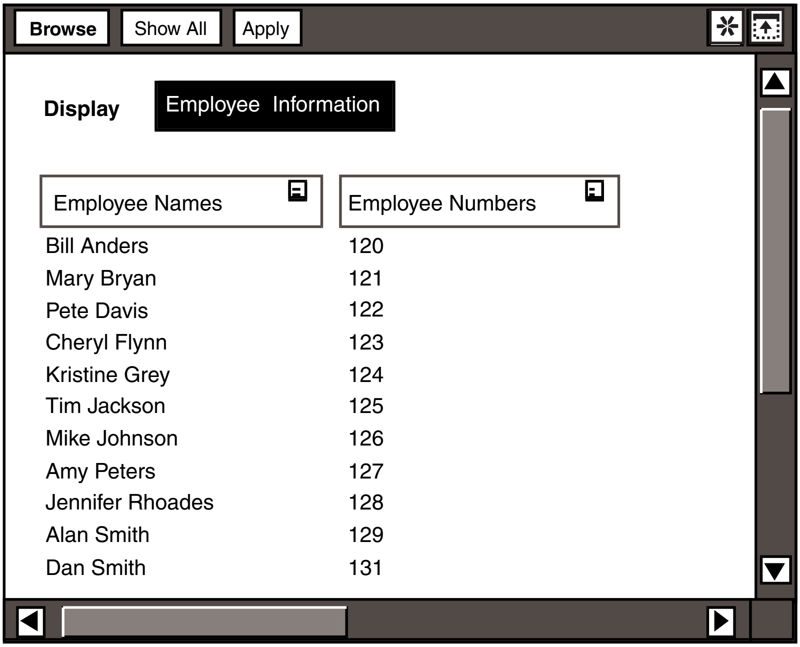
A pull-down menu could also have been used for the `Rating` field if there are only a few levels of scoring.

Creating the Retrieval Form

The retrieval form only needs to contain fields to retrieve the employee number and name. When designing these fields, it is important to ensure the accuracy of entries.



For both of these fields, the entry should be chosen from a browse list. You can create one browse list to be used by both fields. The browse list might be displayed as shown in Figure 65.



Display Employee Information

Employee Names	Employee Numbers
Bill Anders	120
Mary Bryan	121
Pete Davis	122
Cheryl Flynn	123
Kristine Grey	124
Tim Jackson	125
Mike Johnson	126
Amy Peters	127
Jennifer Rhoades	128
Alan Smith	129
Dan Smith	131

Figure 65. Browse list

Set the return column within the field options window to the appropriate column. See “Determining the Return Column” on page 78 for more information. See “Chapter 18. Working with the Browse Feature,” on page 129 for information about setting up browse lists.

Be sure to validate entries against the browse list as well. See “Validating Data with a Browse List” on page 112 for more information.

Chapter 22. Improving Security and Performance

Due to the complex nature of the Forms Design tool, there are many security and performance issues you must take into consideration when designing forms. These issues relate to tasks such as creating physical tables for the users, defining those tables to the logical database, specifying the users who can access those tables, and monitoring the performance. You might need the help of your database administrator to perform these tasks. This chapter contains information related to these DBA activities as they pertain to the Forms Design and Data Entry tools, including:

- Implementing database security
- Determining SQL performance
- Executing SQL statements through the Forms Design window
- Using nonproduction tables

For more information on assigning access keys, predefining access to the database tables, and using the `Secured` field, see the *System Administration Guide and Reference*, *Administering Databases with Meta5*, or *Getting Started with the Meta5 Developer's Desktop*.

Implementing Database Security

There are four methods with which you can limit user access in the data entry process and, consequently, ensure the security of the database:

- Assign user IDs and passwords
- Predefine user access
- Assign access keys
- Predefine database table access

Assigning User IDs and Passwords

Configure the Forms Design icon with the user's database name and password. When you create Data Entry icons from that Forms Design icon, the Data Entry icons automatically have the correct configuration. If you configure the Forms Design icon with your user ID and password, you might forget to change that information when you create the Data Entry icon.

Predefining User Access

When you design forms, you control the data that a user can enter, retrieve, delete, or update. For example, suppose you are designing a form to allow a user to retrieve information from an Employees table in the database. To restrict retrieval to only specific rows from a database table, define a field in the Retrieval Form depiction as predefined, and give it a constant initial value. For example, when defining a `Department Number` field, specify a particular department number so that the user can retrieve only the rows for employees in that specific department.

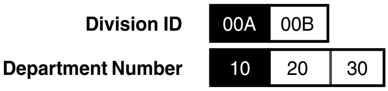
You also can allow the user to retrieve rows for more than one department. Either define an entry field whose values are validated against specific database values, or a choice field whose choices are the specific department numbers.

To constrain two or more fields where there is a dependency between the constraints, you must set up a dependency between those fields. For example, assume that you want to enable the user to retrieve all rows from an Employees table for people who work for division 00A or 00B. Assume that division 00A has departments 10, 20, and 30, and division 00B has departments 10, 15, and 20. With the proper SQL statement, you can set up a browse list to limit the division and department numbers, and indicate that values must be in the browse list. Or, you can restrict a user to certain data using dependent fields.

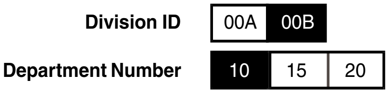
To restrict a user with dependent fields:

1. Define a field and set up the choices. For example, using a field named `Division ID`, define the choices as `00A` and `00B`.
2. Define dependent fields for each of the choices. For example, define a `Department Number` field as a dependent of 00A, and another `Department Number` field as a dependent of 00B.
3. Define the choices for the dependent field. For example, the `Department Number` field dependent on choice 00A has choices 10, 20, and 30. The dependent field for choice 00B has choices 10, 15, and 20.

When the user selects `00A` in the `Division ID` field, the `Department Number` field is displayed as shown in the following illustration.



When the user selects `00B` in the `Division ID` field, the `Department Number` field is displayed as shown in the following illustration.



Regardless of how you design the forms, the DBA is still responsible for granting access to the database tables.

See “Chapter 13. Designing Choice Fields,” on page 81 for instructions on how to create dependent fields.

Assigning Access Keys

By assigning an access key, you identify a user or a group of users to a Data Entry icon, and restrict the user's ability to view or change the icon's contents. An access key is a number from 1 to 64 assigned by the system administrator.

Because a user can mail icons to other desktops, there is still potential for a security violation. Even if an icon has an access key, the receiver of the icon becomes the owner and is therefore able to use it. You might want to discuss predefining database table access with your DBA. See “Predefining Database Table Access with the Secure Feature” for more information.

Predefining Database Table Access with the Secure Feature

The DBA is responsible for controlling access to the database tables by creating users and granting them privileges. You, as the forms designer, can ensure that only those desktop names that are granted database access by the DBA can use the Data Entry icon to access the tables in the database.

See “Securing a Forms Design Icon” on page 46 for information about securing an icon.

Determining Data Entry SQL Performance

The way SQL statements are used has an effect on the performance of the Data Entry tool. You can create fields that are:

- Database-defined
- Database-validated
- Database-initialized

Generating Database-Defined Choice Fields

If you generate choices for a field from the database, the SQL statement that generates the choices is executed when the Data Entry icon is opened.

If the field is large, it could take some time to execute that SQL statement. If delay in opening the Data Entry icon becomes a problem, it is better to make the field an entry field whose data is validated against the database.

Generating Database-Validated Entry Fields

When you define an entry field, there are several ways you can validate the values in that field against the database. Each of the following options have different implications for database performance. You can specify field values:

- In a browse list
- In a database table
- Not in a database table

Specifying Field Values in a Browse List

When you define a field whose values are specified in a browse list:

- The browse list is generated as specified in the Browser Options window.
- The user can always update the browse list. This option should be used with browse lists that are fairly static.

If the browse list is fairly dynamic, you might want to update the values automatically. There are three methods with which you can automatically update browse lists:

- Each time the browse list is used for validation

  With this method, the browse list is not stored with the Data Entry icon. Instead, the list is generated each time a field is validated. This choice should be used with fields that are constantly changing.
- Each time the Data Entry icon is opened

  With this method, the icon then takes longer to open.
- Each time the database is queried

  With this method, browse lists are updated only when necessary because they are only updated when a row is returned that differs from one that is specified in the Browser Options window.

Specify the frequency with which a browse list is updated in the Browser Options window. See “Chapter 18. Working with the Browse Feature,” on page 129 for information on browse lists.

Specifying Field Values in, or Not in, a Database Table

You can specify that an entry should either be in, or not in, a database table. The Data Entry icon runs an SQL statement each time the user enters data to validate the entry. If the database table has a large number of rows, this method could have a significant impact on performance.

Generating Database-initialized Fields

When you define a field that is initialized from the database:

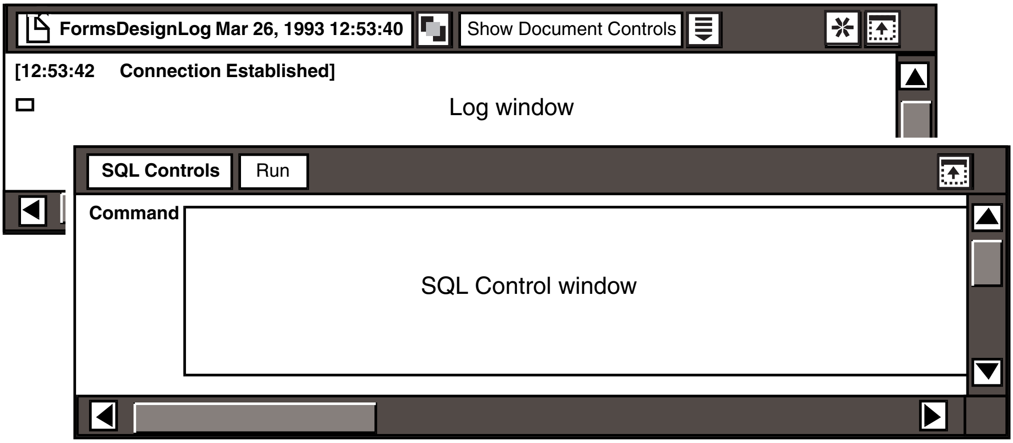
- An SQL statement generates the initial value when the Data Entry form is displayed.
- If the field is defined to reset after an entry, the SQL statement is executed each time the user clicks on the `Enter` button in the Data Entry window header.

Running Commands in the SQL Control Window

You can run SQL statements by entering them through the SQL Controls window. This method is beneficial because you do not need to open an SQL Entry tool each time you need access to the database server.

To execute SQL statements using the SQL Controls window:

1. Select Forms Design Log window, shown in Figure 66. The log window contains a date stamp that indicates the date and time the capture window was opened. The function of the log window is to capture the SQL activities.

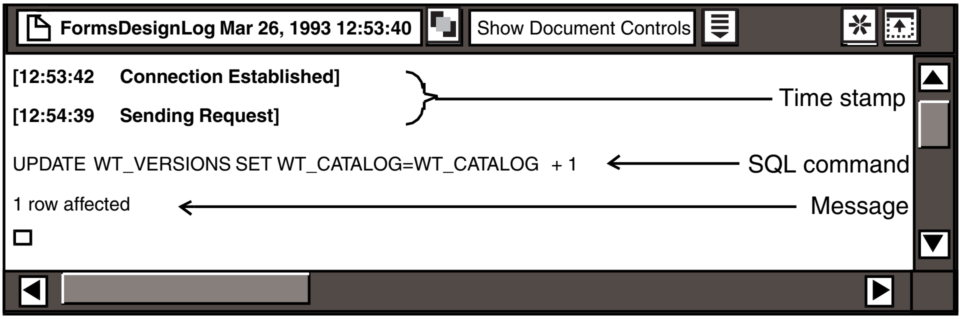


```
[12:53:42     Connection Established]
```

Figure 66. Forms Design Log and SQL Controls windows

2. Type or copy an SQL statement into the command area of the SQL Controls window.
3. Click on the SQL Controls window header.

   The Forms Design tool sends the statement to the database server, and two lines are returned to the log window as shown in Figure 67 on page 186.



[12:53:42 Connection Established]

[12:54:39 Sending Request]

```
UPDATE WT_VERSIONS SET WT_CATALOG=WT_CATALOG + 1
1 row affected
```

Figure 67. Forms Design Log window

Following the time stamp is the SQL statement that was sent to the database server, and a message indicating the result of the activity.

When you close the SQL Controls window, the log window closes and remains on your desktop as a record of the SQL activity. A new log window opens each time you open the SQL Controls window.

Working with Production Tables

Production tables, or database tables that hold vital data an organization uses for essential work, must not be impacted by queries. For example, a query issued by a brand manager should not cause the production applications to slow to a crawl and interfere with the schedules of many other people.

When a user enters information, the SQL statement issued by the Data Entry tool is sent to the database server immediately. This immediate update can cause performance or database integrity problems if users are performing queries against the same tables into which other users are updating or inserting data.

One way to avoid potential performance and integrity problems is to adhere to the following guidelines:

- Have the users entering data use a nonproduction copy of the tables.
- Put the nonproduction tables in a physical database different from the production database.

The users can then apply any changes to the nonproduction copy of the tables. At a later time, you can run the appropriate SQL program to apply those changes to the production tables.

Appendix A. Identifying Keywords and Statements

There are many keywords and statements that assist you when using Forms Design's more technical features. For example, you can specify many @-keywords and &statements that are useful when constructing post-entry SQL statements. You can also modify the contents of a script if you know the acceptable values for the keywords. This appendix provides the following information:

- Definitions of @-keywords
- Definitions and syntax for &statements
- Definitions and acceptable values for scripting keywords

Using @-Keywords

You can enter @-keywords in a field to generate system data, such as the current date, or reference fields in the Data Entry forms. @-keywords must be entered in uppercase. Table 26 on page 187 describes the @-keywords that you can use with the Forms Design tool.

Table 26. Forms Design @-keywords

@-keyword	Description
@COLUMN	A column in the database. This @-keyword is used only in the SQL statement that initializes a field in an Update Form. Use @COLUMN instead of @FIELD when you want to initialize a field to a value in the database, rather than the current value in the field.
@DATE	The current date.
@FIELD	The contents of the field specified in parentheses at the time the @FIELD is evaluated. The evaluation depends on the context in which it is used. If you use @FIELD in the `Post-Entry SQL` field, the field specified must be visible in the Data Entry window when the post-entry SQL is executed. If there is more than one visible field with the same field name, the one closest to the top of the form is used. If you want to reference a field that is used in sub-forms, then assign a global name to the field. If you use @FIELD in an Entry Field Options window and the field containing the @FIELD reference is in a path in a hierarchy, the Data Entry icon searches that path, from bottom to top, and uses the first field or choice encountered whose name is specified in the @FIELD. If there is more than one field with that name at the same level in the hierarchy, the Data Entry icon uses the topmost field.

Table 26. Forms Design @-keywords

@-keyword	Description
@FIELD (GLOBAL:)	The global name in a field that you can use to refer to any other field within the form. When you use @FIELD with a global name, you must first assign a global name to a field, and then use the same global name in the @FIELD command. In the label area of the field options window, you assign a global name by typing `GLOBAL:` followed by a name. If the Data Entry tool detects more than one field in a form with the same global name, a message indicating which field contains the conflicting global name is displayed. See “Assigning a Global Name” on page 166 for more information.
@ICON	The name of the icon.
@PARM(1-10)	A keyword used to set up a constraint on a browse list. The conditions of the constraint, however, are not specified in the expression. The @PARMs allow specification of the conditions from the `Browser Parameters` field in the field options windows. See “Setting Up Conditional Expressions with @PARMs” on page 166 for more information.
@TIME	The time in hours, minutes and seconds (HH:MM:SS).
@USER	The name of the desktop that contains the icon. The desktop name is capitalized exactly as entered by the user when logging on to the desktop.
@USERID	The unique 8-digit hexadecimal number that identifies the desktop.

Using &Statements

As with the SQL Entry tool, you can incorporate flow control and conditional processing by including &statements, functions, and function statements in post-entry SQL. Table 27 explains the &statements and functions that are supported by the Forms Design tool.

The &DISPLAY statement is not supported in the `Post-Entry SQL` field.

For detailed information on these statements and functions, see the *Data Access Tools User's Guide*.

Table 27. Acceptable functions and &statements in post-entry SQL

Statement/ Function	Syntax	Description
&CONSTRAINT	&CONSTRAINT (column name, data type, operator type, data value)	A function used to constrain a field that accepts multiple data items. For more information on the &CONSTRAINT function, see “Chapter 15. Validating Data,” on page 103.
&DATE	&DATE (date)	A function used in an INSERT command and a WHERE clause which allows you to convert date strings to those represented internally by the database server.

Table 27. Acceptable functions and &statements in post-entry SQL

Statement/ Function	Syntax	Description
&IMPMSG	&IMPMSG ('string')	A function used to post an important message in a Data Entry form. You can specify text and @-variables in the message parameter. When the message is displayed by the Data Entry tool, it is preceded by the phrase `Application Message:` to distinguish it from system generated important messages.
&ERROR	&ERROR action;	A statement used to control the method of handling errors (with the exception of parsing errors) by specifying an action such as STOP, IGNORE, or MESSAGE.
&GOTO	&GOTO label;	A statement that directs the flow of a program to the SQL statement after the specified *&label* variable.
&IF	&IF condition THEN GOTO label;	A statement that uses a condition to direct the flow of a program to the SQL statement after the specified *&label* variable.
&label	& label;	A statement that uses an ampersand (&) to precede a word of your choice in SQL, and allows you to direct the flow of the program when used with the &GOTO statement.
&SET	&SET @variable = expression;	A statement used to set @-variables while executing a program.
&SQLNAME	&SQLNAME (tablename [.columnname])	A statement used to take advantage of table name or column name indirection. The table name parameter is the internal table name specified in the WTDD.
COLVAL(n)	COLVAL(n)	A function that accesses the results of the last-executed statement using the value from the *n*th column of the first row of data.
ERRNUM	ERRNUM	A function that returns the error number of the last SQL statement executed from the database.
NROWS	NROWS	A function that returns an integer indicating the number of rows affected by the last statement executed.

In addition, the `Post-Entry SQL` field also supports @-keywords, as described in “Using @-Keywords” on page 187.

Knowing the Acceptable Values for Scripting Keywords

Certain keywords can only be followed by one of several predefined values. These predefined values correspond to the values displayed through the graphical user interface (GUI). Table 28 explains the predefined values for the scripting language.

Table 28. Acceptable values for certain scripting language keywords

Keyword	Function	Allowed values
error	Determines the data entry error signal.	“FLASH”, “SHORT”, “MEDIUM”, “LONG”
updatebrowserchoice	Defines how to update the browse list.	“REQUEST”, “OPEN”, “VALIDATION”, “SQL”
browserSQLchoice	Specifies the source of browser SQL.	“LOCAL”, “DOCUMENT”, “CUSTOM”
sqlsource	Specifies the source of post-entry SQL.	“LOCAL”, “DOCUMENT”
onerror	Determines the type of action taken when an error occurs.	“STOP”, “PROCEED”
width	Sets the width of an field.	Any number that correlates to ISUs (Incredibly Small Units). The width is represented by ISUs in the code regardless of whether the width is set in inches or centimeters. There are 57,600 ISUs per inch, and 145,728 per centimeter.
multivalue	Specifies if multiple values are allowed.	“NO”, “YES”
input	Determines if input is required for the field.	“REQUIRED”, “OPTIONAL”
initvalue (except update form)	Sets the initial value for a field.	“BLANK”, “CONSTANT”, “DBS”, “TODAY”, “NOW”, “USER”
initvalue (update form)	Sets the initial value for a field.	“COLUMN”, “CONSTANT”, “DBS”, “TODAY”, “NOW”, “USER”
choicewidth	Specifies if the choice width is constant or variable.	“CONSTANT”, “VARIABLE”
choicedef	Specifies the source of the choice definition.	“PREDEFINED”, “DBS”
destination	Determines the storage of information.	“DBS”, “VARIABLE”, “NONE”
accept	Specifies the acceptable data types.	“INTEGER”, “REAL”, “CHARACTER”, “DATE”, “TIME”, “TIMESTAMP”
date resolution	Sets the date resolution for a field or a browser column.	“DAY”, “WEEK”, “QUADWEEK”, “MONTH”, “ODDBIMONTH”, “EVENBIMONTH”, “QUARTER”, “YEAR”
capitalization	Determines the capitalization of entered character strings.	“NONE”, “ALL”, “INITIAL”

Table 28. Acceptable values for certain scripting language keywords

Keyword	Function	Allowed values
validate	Specifies the validation method of input.	“ALWAYS”, “IFCHANGED”
compute	Computes the aggregate value in SQL to determine the initial value of the field.	“MIN”, “MAX”, “COUNT”, “AVG”, “MAX+1”, “OTHER”
whenentered	Determines what to do with a field value when a user clicks on the `Enter` button.	“RESET”, “RETAIN”
dbsvalidation	Sets the type of DBS validation to be performed on a field.	“NONE”, “BROWSER”, “NOTIN”, “IN”
stovalue	Specifies what value is stored for section or fields.	“NAME”, “VALUE”, “DBS”
match	Matches the value for the destination description in section or fields.	“NAME”, “VALUE”
operatormay	Determines permission to delete rows associated with retrieval form options.	”NOTDELETEROWS”, “DELETEROWS”
initvalue	Sets the initial value for choice or section fields in an update form.	“COLUMN”, “DBS”, “OTHER”
length (browser column)	Specifies the length of a displayed browser column.	“SHORT”, “LONG”
order (browser column)	Sets the sorting order for browser columns.	“ASCENDING”, “DESCENDING”, “OTHER_COLUMN”
display (browser column)	Controls the display of a browser column.	“SHOW”, “HIDE”
displayas	Sets the data type of the displayed values in a browser column.	“INTEGER”, “REAL”, “CHARACTER”, “DATE”, “TIME”, “TIMESTAMP”
precision (browser column)	Determines the precision for a column that contains real numbers in a browse list.	“0000.1”, “000.02”, “00.003”, “0.0004”, “.00005”
fldlayout	Sets the field layout type for a group.	“VERTICAL”, “HORIZONTAL”
fldsize	Specifies that the field size for fields within a group are either constant or variable.	“CONSTANT”, “VARIABLE”
sepunits	Sets the separation units used in vertical field separation.	“PIXELS”, “LINES”
horizontalalign	Determines the horizontal alignment for a group relative to its space.	“DEFAULT”, “LEFT”, “CENTER”, “RIGHT”
verticalalign	Determines the vertical alignment for a group relative to its space.	“DEFAULT”, “TOP”, “MIDDLE”, “BOTTOM”

Table 28. Acceptable values for certain scripting language keywords

Keyword	Function	Allowed values
browserdispmode	Specifies when a browse list is displayed.	“MANUAL”, “AUTO”
choicestyle	Sets the style of choice or section fields to be either displayed horizontally or in a pull-down menu.	“FLAT”, “MENU”
placement	Determines placement of dependent fields for a choice or section field.	“BELOW”, “TO_THE_RIGHT”
initvalue	Sets the initial value for a choice or section field.	“LEFTMOST”, “DBS”, “OTHER”

Glossary

This glossary defines terms that are used in this book and throughout the Meta5 library. If you do not find the term you are looking for, see the index of this book.

access key. A code number that restricts access to a specific icon. Only users whose desktops include the code number can use the icon.

access privileges. A set of permissions assigned by an icon's owner to control who may view or change the icon.

Add Browser. The window header button that adds a browse list to the Browser Library header.

Add Choice. The window header button that adds a choice to a list of predefined choices.

Add Comment. The window header button that adds a field in which a forms designer can type instructions or comments.

Add Field. The window header button that adds a field to an entry form, a retrieval form, or an update form depiction.

Add Group. The window header button that adds a group to an entry form, a retrieval form, or an update form depiction.

Add Line. The window header button that inserts a line in a form.

Add Section. The window header button that adds a choice to a section field.

Apply. A window header button that puts into effect any changes that the user has specified in an options or control window.

@-keyword. A system-defined variable that returns current values within a capsule application or within the headers and footers of a printable icon.

@time. A system-defined @-keyword that returns the day of the week, date, and time.

@user. A system-defined @-keyword that returns the user's desktop name.

@-value. The value assigned to an @-variable in the Capsule tool's User Input Controls window.

@-variable. A control used to represent a value in any tool that can run inside a capsule application. The Data Entry tool can be used to set @-variables inside a capsule application, if the forms designer designed it in that way.

Browse button. A button that is displayed in the window header of a form when a particular field is selected. When the `Browse` button is selected, a window opens displaying a list of valid field entries.

browse list. A list that is displayed when you select the `Browse` button. The browse list contains valid entries for the particular field associated with the `Browse` button.

browser. A browse list as defined in the Forms Design icon by the forms designer.

Browser Column Options. The window in which the forms designer sets options that apply to the browse list column.

Browser Library. The heading in the main window which defines the browsers associated with the Data Entry icon.

Browser Options. The window in which the forms designer sets options for the browse list.

button. A small area within a window that provides other options or causes an action to occur. Many workstation activities are directed by using the mouse to click on software buttons.

Cancel. A window header button that closes an options or control window without saving any changes other than those that were saved the last time the user clicked on the `Apply` button.

Capsule tool. A tool in which you combine the functions of several Meta5 tools to build applications.

Capture SQL item. A restricted item within the Data Entry window's `Special` menu. This item captures all SQL activity generated by the form.

capture window. A Text window that records the SQL statements sent to the database from the Command area of the SQL Controls window.

catalog. Tables, maintained by the database manager, that contain descriptions of objects, such as tables, views, and indexes.

category window. The window that displays a specific category of database tables for which an icon is configured.

choice number. The relative number of the choice within an option. The first choice has a choice number of one.

Choice Options. The window in which the forms designer assigns a name and value to a choice in a predefined field.

choice value. A notation used to represent a Choice Name. The Choice Value can be stored in the database instead of the Choice Name.

Confirmation window. The window in which you confirm or cancel a request to delete data from the database.

constraint. A factor that limits the range of data retrieved from the database to reduce the amount of unneeded data that is returned. For example, the forms designer might constrain data based on territory and period, so that only data relevant to New York during March, 1992 is retrieved.

container. Any folder, file drawer, or other icon that can contain other icons.

Create Column. A button in the main window header that adds a customized column to the browser.

Create Data Entry Icon. The choice in the `Special` menu that creates a Data Entry icon based on the forms in the main window.

Create Forms Design Icon. The choice in the `Special` menu of the Data Entry tool that recreates a Forms Design icon from a Data Entry icon.

database administrator (DBA). The person who issues administrative instructions to a database service. The database administrator is responsible for defining the data tables that all data access tools use, as well as for monitoring database performance and assigning user access.

database validation. The process by which the value entered into a field is compared with values in the database.

Data Entry Log window. A window that records all the information you enter into the database while the main window of the form is open.

Data Entry SQL Log window. A window that records all SQL activity generated by the Data Entry icon.

Data Entry Options. The window where the forms designer or the data entry operator set options that apply to the Data Entry icon.

Data Entry Tool. The tool that is used to enter, retrieve, delete, and update data on a database server, or to set @-variables in the User Input Control window of a capsule application.

data type. The internal representation of data in a physical database column. Data types generally available in the Meta5 logical tables are integer, real, character, and date. Other data types available in some Meta5 logical tables include null, long varchar, varchar, time, and timestamp.

Data window. The window that shows all the data you requested from the database, and allows you to delete information from the database.

DBA. Database administrator.

decimal precision. The number of displayed decimal places in a number.

default value. A value that is assumed when no value has been specified.

dependent field. A field that displays only when a particular choice in a field is selected.

depiction. The illustration of an object. For example, the categories in the Forms Design category window, or the forms in the main Forms Design window, or an icon within a Capsule window.

displayed precision. A number that has been rounded or truncated. When a number has been rounded off, the displayed number rounds down if the last digit is less than 5, and round up if the last digit is 5 or greater (vice versa for negative numbers). The number is truncated at the specified number of decimal places.

Enter Data item. An item in the `Special` menu in the Data Entry window that displays the entry form.

entry form. The form used to enter information into a database table or capsule application.

error signal. A beep or screen flash warning you that you have made an incorrect field entry.

field. A field in a form from which you can select one of several choices, rather than entering text.

field type. The method of data entry in a form. For example, a choice field type allows the data entry operator to make a selection from a group of several choices. Other field types include entry, predefined, and section.

file drawer. A container, created or renamed by the system administrator, that is shared among desktops. A file drawer typically contains folders.

foreign key. A column in a database table that references a primary key in another database table. A foreign key can be viewed in the Forms Design tool's category window.

form. A component of the Data Entry tool used to enter, retrieve, delete, or update information in a database; or supply values to the User Input Controls window of a capsule application.

forms designer. The person who uses the Forms Design tool to create Data Entry icons.

Forms Design Log. The window that displays a record of all actions performed by the forms designer while the main Forms Design window is open.

function keys. Programmed keys on the workstation keyboard used to manipulate icons and windows.

group. A collection of fields that may be placed horizontally or vertically.

Group Options. The window in which the forms designer specifies the options for groups.

Icon Options window. A window in which you can reset values or display information about an icon.

information box. A box that is displayed in the Forms Design window when the forms designer double-clicks on a form depiction field or heading. If the forms designer selects a field, this box displays the field type, the acceptable data type, and the name of the database column to which the field corresponds. If the forms designer selects the depiction heading, the information box displays information about all fields in the depiction.

initialization. The process of giving value to a data object at the beginning of its lifetime.

initialization area. An area in the Field Options window that contains information about how the field is initialized.

insertion point. A symbol that indicates the point at which text can be inserted in a field.

Line Options. The window in which the forms designer sets options for lines in a Data Entry form.

list separator. A character, such as a comma, used to break a text string into separate items.

operator. A symbol the user selects or enters along with a value to define the amount, range, or characteristics of data to be retrieved from a database.

options window. Any window that the user can open by using the Options key or the options function on the pop-up function pad. Options windows enable the user to view and modify the properties of the selected object.

parser. Software that analyzes the syntax and determines the meaning of SQL statements.

path name. A designation for the location of a file or icon. For example, a path name for a file on a computer disk specifies the names of the disk and all the directories leading to the file; a path name for a script called by a Cooperative Application Services transformer specifies the name of any Capsule applications in which the script is located; the path name for an Meta5 icon specifies the names of any Meta5 folders that must be opened to get to the icon.

pattern. A series of characters arranged to illustrate the correct format for entering character data into a field. For example, 999-99-9999 is a character pattern for a Social Security number. The forms designer can define a pattern that the Data Entry tool uses to validate character data that a data entry operator enters.

physical database. The physical set of stored data tables that underlies a logical database.

physical table. A table in a physical database.

pixel. An acronym for picture (pix) element (el), the smallest element on a video display screen to which a color or light intensity can be assigned.

post-entry SQL. One or more SQL statements that run each time a forms designer enters data into an entry form or update form.

predefined field. A field in a form that contains information you cannot change.

primary key. One or more columns in a database table that uniquely identifies the rows in that table. The primary key can be viewed in the Forms Design tool's category window.

production tables. Tables against which users are currently and actively issuing queries. For example, tables used as a high access online transaction system.

relational data. In a relational database, information that is related to other information and whose relationships can be defined in the form of tables consisting of rows and columns of data.

restricted menu item. A choice in the `Special` menu that can be used only by the system administrator or forms designer.

retrieval form. The form used to retrieve information from a database table.

Retrieve Data item. An item in the `Special` menu in the Data Entry window that displays the retrieval form.

Retrieval Form Options. The window in which the forms designer can define the appearance and operation of the retrieval form.

section field. The field in a form that displays several choices from which the data entry operator can make a selection. Each choice displays a section that contains dependent fields. All fields within each section are active and are validated and entered when the data entry operator clicks on the `Enter` button.

Section Options. The window that displays the section field options.

secure. To restrict access to an icon.

Show All button. A browse window button that redisplays a browse list after you have specified constraints.

Show Controls button. A button in the window header of a Capsule icon that opens the User Input Control window, where @-variables are stored.

Show Data button. A button in the Data window header that displays the updated data.

Show Data Entry Options item. An item in the `Special` menu in the Data Entry window that opens the Data Entry Options window.

Special button. A button in some window headers and on the far-right side of the message area which opens the `Special` menu.

Special menu. A menu that is displayed when the user points to the `Special` button and presses the left mouse button. In tool windows, the menu choices are usually reserved for special functions or controls associated with the icon.

SQL. Structured Query Language.

SQL Control. The window in which the forms designer enters SQL statements and they are sent to the database.

Start Transcript item. An item in the `Special` menu in the Data Entry window that opens a Data Entry Log window and records the user's entries.

Structured Query Language (SQL). A command language for use with relational databases. The language consists of statements to insert, update, delete, query, and protect data.

storage area. An area in the Field Options window where the forms designer specifies where data is stored.

system administrator. The person who is responsible for operating and configuring Meta5, including its workstations, servers, and network.

table. A single database file, consisting of data organized in columns and rows.

table name. The title or label for a division of a database category.

tool. A workstation program, such as Text, Reporter, or Query.

Update Browse Lists item. An item in the `Special` menu in the Data Entry window that updates the browse lists.

update form. The form used to modify information in a database table.

Update Form Options. The window in which the forms designer sets options to define the appearance and operation of the update form.

User Input Control window. The window in a Capsule tool that is used to store @-variables.

user name. A name used to access a desktop, system, gateway, or database service.

validation. The checking of entered data for completeness, accuracy, or reasonableness.

validation area. An area in the Field Options window used to specify how data is validated and how character strings are edited.

view. A copy of a query, which can be named, stored, edited and redisplayed for later use.

Workstation Tools Data Dictionary (WTDD). A logical data dictionary that helps standardize the workstation view of data. The WTDD stores the external names of data items and defines how the items are to be displayed. The WTDD is implemented as a set of standard relational tables, so it can be managed through icon-based workstation tools.

WTDD. Workstation Tools Data Dictionary.

Meta5 Publications

This section lists Meta5 publications. To order copies of the books listed here, or to get more information about a book, see your Meta5, Inc. representative.

- **Meta5 Volume 1**

  Getting Started with the Meta5 Developer's Desktop

  PC Integration Tools

- **Meta5 Volume 2**

  Spreadsheet User’s Guide

  Plot User’s Guide

- **Meta5 Volume 3**

  Text User’s Guide

  Layout User’s Guide

- **Meta5 Volume 4**

  Capsule User's Guide

  BASIC Tool User’s Guide

- **Meta5 Volume 5**

  Data Access Tools User’s Guide

- **Meta5 Volume 6**

  Forms User’s Guide

- **Meta5 Volume 7**

  Transformers Guide

- **Meta5 Volume 8**

  Error Messages and Codes

- **Meta5 Volume 9**

  Installing Meta5 LAN Components

  System Administration Guide and Reference

- **Meta5 Volume 10**

  Database Gateway Services Guide

  Administering Databases with Meta5

- **Meta5 Volume 11**

  Installing and Configuring Meta5 Open Clients

  Developing Applications with Open Data Access Service

- **Meta5 Volume 12**

Administering Host Services for MVS:DB2 and Cooperative Application Services

Planning and Installing Host-to-LAN Communications for MVS

Index